A Retrofit Network Intrusion Detection System for MODBUS RTU and ASCII Industrial Control Systems

Thomas Morris
Mississippi State University
morris@ece.msstate.edu

Rayford Vaughn
Mississippi State University
vaughn@research.msstate.edu

Yoginder Dandass
Mississippi State University
yogi@cse.msstate.edu

Abstract

MODBUS RTU/ASCII Snort is software to retrofit serial based industrial control systems to add Snort intrusion detection and intrusion prevention capabilities. This article discusses the need for such a system by describing 4 classes of intrusion vulnerabilities (denial of service, command injection, response injection, and system reconnaissance) which can be exploited on MODBUS RTU/ASCII industrial control systems. The article provides details on how Snort rules can detect and prevent such intrusions. Finally, the article describes the MODBUS RTU/ASCII Snort implementation, provides details on placement of a MODBUS RTU/ASCII Snort host within a control system to maximize intrusion detection and prevention capabilities, and discusses the system's validation.

1. Introduction

National Electric Reliability Council (NERC) Critical Infrastructure Protection (CIP) Standard 005-4a [1] requires utilities and other responsible entities to place critical cyber assets within an electronic security perimeter. Electronic security perimeters must be subjected to vulnerability analyses, use access control technologies, and include systems to monitor and log the electronic security perimeter access. Industrial control system operators from other critical industries have followed the electric transmission and generation industry lead and have begun to adopt the electronic security perimeters to protect cyber assets in both control rooms and in the field. Electronic perimeter security minimizes the threat of illicit network penetrations, however, the concept of defense in depth encourages cybersecurity defenses within the electronic security perimeter including but not limited to virus scanning and deployment of intrusion detection systems (IDS) and intrusion prevention systems (IPS). This work documents an extension of a MODBUS RTU and MODBUS ASCII data logger to enable the use of the Snort [2] intrusion detection and intrusion prevention system features to protect retrofitted industrial control system assets within an electronic security perimeter.

The recent discovery of the Stuxnet [3] worm highlights the need to protect legacy serial based cyber assets such as remote terminal units (RTU) and intelligent electronic devices (IED). As of September 2010, the Stuxnet worm has infected over 100,000 computers in over 155 countries [4]. The Stuxnet worm searches for hosts with the Siemens WinCC human machine interface software package installed. If a WinCC host system is found and the WinCC host is connected to a Simatic S7-417 PLC and certain signatures match the targeted physical process controls, Stuxnet alters the firmware in the PLC. The Simatic S7-400 PLC series supports both Ethernet and serial port communications. This suggests it is possible to alter PLC the firmware of a serially connected RTU, IED, or PLC after a HMI host node is compromised.

A compromised computer serially connected to a control system device may also inject control system commands and false measurements, alter configuration settings on devices, and perform denial of service attacks against devices.

Serially linked control system devices are often connected using industrial radios. Such radio links can be compromised to allow attackers to remotely inject control system commands and false measurements, to perform system reconnaissance attacks, and to perform denial of service attacks [5].

Many security professionals consider serial links secure because they are non-routable protocols. However, the Stuxnet worm and the presence of vulnerable industrial radio links show this to be inaccurate and motivate the need for intrusion detection and intrusion prevention systems to protect RTU, IED, and PLC type devices connected to serial links.

The body of this paper includes a section discussing related works. Next, a section provides discussion on intrusion detection in industrial control systems for various types of threats. Next, a section describes the MODBUS RTU/ASCII Snort implementation including details on MODBUS RTU/ASCII to MODBUS TCP/IP conversion, details on the MODBUS RTU/ASCII Snort software architecture, guide lines for Snort host placement within a MODBUS RTU/ASCII network, and information on how the MODBUS RTU/ASCII Snort was validated. The paper ends with discussion of future works and conclusions.

978-0-7695-4525-7/12 $26.00 © 2012 IEEE
DOI 10.1109/HICSS.2012.78

2. Related works

Snort is a rule based open source network intrusion detection and intrusion prevention tool [2]. Snort collects and logs network traffic, analyzes network traffic searching for rule violations, and alerts the administrator of suspicious activity. Snort is commonly used to monitor Ethernet and TCP/IP communications traffic.

Quickdraw [6] is a Snort preprocessor and a set of Snort rules developed for industrial control systems using the MODBUS/TCP, DNP3, and Ether/IP communication standards. The quick draw rules include alerts for invalid device configuration attacks, coil and register read and write attacks, high traffic volume attacks, malformed MODBUS application data unit (ADU) content attacks, unresponsive device scenarios, and port and function code scanning attacks. Currently, the Quickdraw preprocessor and Snort rules are limited to protecting TCP/IP systems. This article describes a mechanism to enable use of the Quickdraw Snort rules and other Snort rules to protect a serial based industrial control system. The Quickdraw MODBUS/TCP rules were used to validate the work described in this article. Currently there are no known Snort rules or preprocessors to support monitoring industrial control serial port protocols.

Many researchers have developed statistics based intrusion detection systems targeted for industrial control systems [7][8][9][10][11]. Statistical intrusion detection systems use statistical methods to classify network traffic as normal or abnormal (or into smaller sub-classes). Various model types or classifiers can be used to build the statistical model, including neural networks, linear methods, regression models, and Bayesian networks. There are two types of inaccuracies in intrusion detection systems: false positives and false negatives. False positives generate a false alarm when there is no intrusion, while false negatives miss an actual intrusion. False positives may lead to dropping a valid communication packet which can have catastrophic results on an industrial control system. Snort uses pattern matching to deterministically detect rule violations. Correctly formed rules will not generate false positives. Because Snort is deterministic, validated rules may be used with Snort in intrusion prevention mode; a mode which drops illegal network packets.

3. Industrial control system overview

Industrial control systems are distributed cyber-physical systems. Figure 1 shows an example of a typical industrial control system. Remote terminal units are connected to sensors and actuators to interface directly with the controlled physical system. RTU store control parameters and execute algorithmic code (such as ladder logic or C programs) to directly control the physical process. Industrial control systems also support supervisory control and data acquisition. Industrial control systems include a master terminal unit (MTU) connected to the RTU via a communication link. This communication may use Ethernet or serial port technologies including RS-232 and RS-485. A common application layer protocol for MTU to RTU communication is MODBUS which includes MODBUS/TCP for Ethernet networks and MODBUS RTU or MODBUS ASCII for serial port networks. The MTU polls the RTU periodically to read physical measurements from the controlled process. This information is displayed on a Human Machine Interface (HMI) to allow situational awareness and control. HMI allow control system operators to interact with the physical process. For example an operator may open a breaker to island an electric circuit, or open a valve to release pressure in a pipe or direct material flow. The MTU, RTU, communication link, HMI, and operator form a supervisory feedback control loop.

Figure 1 shows a typical industrial control system network. Often industrial control systems are connected via Ethernet to a corporate network used for day to day business operations. In the figure, the corporate network and the control system network are isolated by a security gateway. Isolation of the corporate and control system networks may be accomplished via use of virtual LANs, gateway devices which enforce various authorization schemes, and/or firewalls.

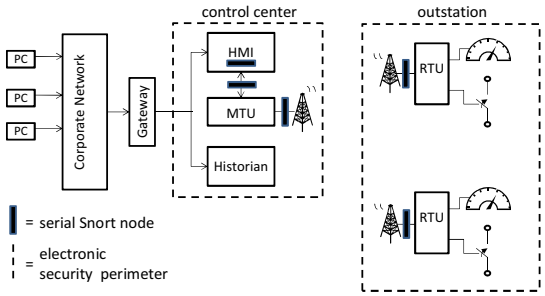

Figure 1. Example industrial control system communications architecture

The diagram in Figure 1 includes 4 networked devices HMI, MTU, Historian, and RTU. Often multiple kinds of communication links are used within an industrial control system. In the example network, the security gateway connects to the HMI host and the historian via an Ethernet link. The HMI host is connected to the MTU via a RS-232 serial link. The MTU communicates with the RTU via an RS-232 or RS-485 serial link. The MTU to RTU link includes a proprietary radio link. Radio links are often used to connect device over distance without the need to install a wired link. The HMI to MTU and the MTU to RTU links use the MODBUS RTU or MODBUS ASCII protocol. The system diagramed in Figure 1 matches control systems in the

MSU SCADA Security Laboratory which includes multiple industrial control systems using commercial hardware and software and laboratory scale functional physical process from various critical industries.

There are many variations on the system shown in Figure 1. Contemporary systems often replace the serial links with Ethernet links. Many types of wireless links and many application layer protocols are available.

This article describes a Snort based intrusion detection and prevention system for serial based MODBUS networks (MODBUS RTU and MODBUS ASCII). MODBUS was chosen for initial development because of its prevalence in industrial control systems. The methods described in this article can be extended to support other communication protocols which also use serial communication.

4. Detecting attacks against industrial control systems

Four types of attacks against industrial control system communication networks are denial of service, response injection, command injection, and reconnaissance attacks.

Denial of Service (DOS) attacks attempt to break the communication link between the remote terminal and master terminal or human machine interface. Breaking the communication link between master terminal or human machine interface and the remote terminal breaks the feedback control loop and makes process control impossible. DOS attacks take many forms. Many DOS attacks attempt to overwhelm hardware or software on one end of a communication link so that it is no longer responsive. Other DOS attacks send ill timed or malformed network packets which cause errors in a remote device's network stack and cause the remote device to become unresponsive.

MODBUS devices include configuration settings to allow operators to remotely force a RTU into listen only mode, remotely restart communications, and remotely clear RTU counters and diagnostic registers. Each of these features can be used by an attacker to execute a denial of service attack. The Quickdraw tool [6] includes Snort rules to detect these configuration exploits. With MODBUS RTU/ASCII Snort MODBUS serial based control systems can also be protected against these threats.

In [5] Reaves et al. discuss how to discover and connect to a proprietary SCADA radio used to form the MTU to RTU communication link (such as shown in Figure 1). This penetration enables an attacker to execute a denial of service attack against industrial control systems. The industrial radio used to demonstrate this attack uses a carrier sense and back off arbitration scheme to regulate slave device wireless transmissions. If a single device continually transmits without breaks other slave radios in the network will back off and wait to transmit until the offender stops transmitting. In practice such an attack stops communication between the MTU and RTU. This causes the HMI to no longer update with fresh physical process measurements and causes the operator to see stale process measurements. Without an intrusion detection system the operator may not know that an attack is in progress. The only symptom is that HMI inputs seem to have no affect on the controlled process. This attack may be executed via continuous transmission of invalid data such as streams of random numbers, or as continual transmission of otherwise valid MODBUS traffic. In either case Snort rules can be used to detect the attack. Such a rule can be based upon packet length, such as exceeding MODBUS/TCP maximum length, or on the volume of traffic in a given time frame. The Quickdraw tool includes a Snort rule to detect packets of illegal length. MODBUS RTU/ASCII Snort times out on continuous transmission of greater than 300 bytes and transmits the captured traffic as a TCP/IP packet to allow Snort to detect the large continuous streams of data.

The network stack implemented in MODBUS MTU and RTU devices may not be sophisticated enough to handle receipt of malformed communication traffic. Some devices will reset themselves or hang when a malformed packet is received. This problem has been recognized by many equipment vendors and contemporary devices often include more robust network stacks. However, many serial systems in existing control systems are legacy devices with no support for malformed packet checks. The Quickdraw tool [6] includes a Snort rule to detect non MODBUS/TCP traffic on the network. That particular rule checks a field which is not used in MODBUS RTU/ASCII protocols and therefore this rule is not applicable for these protocols. However, the length check rules described above serves as a minimal malformed packet check. Also, some malformed packets are possible with MODBUS RTU/ASCII protocols. For example, only addresses in the range 0-247 are allowed in a field which can hold numbers greater than 247. A Snort rule could check for a packet with an illegal address. Many other rules related to detecting malformed packets are possible.

RTU and PLC control algorithms often perform mathematical computations on device measurements. When such computations include the division operation care must be taken to ensure the denominator is never zero to avoid divide by zero exceptions. Many RTU halt the running program when a divide by zero exception occurs. Halting the program in turn stops process control until the program is restarted. Snort rules can be used to prevent RTU coils and registers from being remotely loaded with values which will lead to a divide by zero exception. Such rules are

specific to the controlled process and cannot be generalized.

A device which has been attacked may exhibit symptoms of the attack by continually reporting that it is busy when queried. The Quickdraw tool includes Snort rules to detect excessive device busy responses and acknowledge exception code responses. The fields used for these rules are also used for MODBUS RTU and MODBUS ASCII systems and therefore can be used to detect similar symptoms with serial MODBUS control systems.

Response injection attacks inject false responses into a control system. Since control systems rely on feedback control loops which monitor physical process data before making control decisions, protecting the integrity of the sensor measurements is critical. False response injection can be used by hackers to cause control algorithms and operators to make misinformed decisions.

In a MODBUS RTU or MODBUS ASCII network the MTU regularly queries the RTU to read registers associated with the physical process state. In normal operation the RTU responds to the MTU queries with data from the requested registers. If two responses to a single query are received the MTU accepts the first received response and rejects subsequent responses. The proprietary wireless vulnerability described in [5] (and mentioned above) can be exploited to eavesdrop on MTU to RTU communications and to inject false RTU to MTU communication packets. This exploit can be used to inject falsified MODBUS response packets after a query is issued. The attacker monitors the radio network for MODBUS queries. If the attacker's falsified response arrives at the MTU before the valid response, the false response is accepted. Responding faster than the RTU can be achieved in two ways. First, the attacker can take advantage of the periodic nature of MTU queries and queue a false response before the MTU query transmission completes. In this case a Snort rule can be developed to alert when multiple MODBUS responses are received with the same address and same function code within a time period to small to normally receive 2 responses. Two responses in less than the normal query period would indicate an attack.

In a second version of this attack there is only one response because the attacker stops all transmissions from the radio connected to the legitimate RTU via the continuous transmission DOS attack mentioned above. The attacker inserts the falsified MODBUS response within the continuous stream of DOS traffic. The proprietary radio network exploited for these attacks uses slotted communications with dedicated slots for MTU to RTU communication and RTU to MTU communication. The MTU continues to query the RTU during the DOS attack. In this case, the Snort rule mentioned above which alerts for packets larger than legal MODBUS packets would detect this attack.

Control systems often read measurement values such as pipe pressure, voltage, or current. Often these values are transmitted in fields of a type which can hold values which are physically impossible measurements. For instance, floats can hold negative numbers; however, a pipeline pressure can never be negative. Other measurements may have legal ranges which correspond to the physical properties of a specific system. In both cases, Snort rules can be developed to alert when measurements fall outside predefined ranges.

Command injection attacks inject false control and configuration commands into a control system. Control injection can be classified into 2 categories. First, human operators oversee control systems and occasionally intercede with supervisory control actions, such as opening a breaker. Hackers may attempt to inject false supervisory control actions into a control system network. Second, remote terminals and intelligent electronic devices are generally programmed to automatically monitor and control the physical process directly at a remote site. This programming takes the form of ladder logic, C code, and registers which hold key control parameters such as high and low limits gating process control actions. Hackers can use command injection attacks to overwrite ladder logic, C code, and remote terminal register settings.

System specific rules can be developed to block commands not supported for a given control system. MODBUS supports RTU addresses in the range 0-247; however, most control systems will use a small portion of this allowed set. MODBUS supports function codes in the range 0 to 255. Many function codes are reserved for common functions, while others are available for custom functions. Finally, MODBUS read and write calls include a start address, within RTU memory, and the number of (bytes, coils, inputs) to be read or written. Snort rules can be developed to detect unsupported RTU addresses, unsupported function codes, and unsupported read/write address and size combinations.

RTU often have registers which hold system setpoints; such as target measurement values for a PID scheme, limits to generate a system alarm, or limits to generate on/off control actions. Snort rules can be developed to detect invalid register values which may cause incorrect or unsafe process functionality.

Reconnaissance attacks allow cyber attackers to reconnoiter a system before attacking. For example, NMAP is used to identify connected systems and then finger print the system. Finger printing allows an attacker to learn which ports are open, the identity and version of the remote operating system, and the identity and version of remote network stack daemons. With this information the attacker can plan a more effective attack. MODBUS systems can be

scanned to learn connected RTU addresses, supported function codes of connected RTU, and supported address ranges of MODBUS inputs and coils. Quickdraw includes rules for detecting function code scans and scans to learn supported address ranges of MODBUS inputs and coils. These rules also apply to MODBUS RTU and MODBUS ASCII systems. MODBUS RTU/ASCII Snort places the MODBUS RTU/ASCII RTU address value in the MODBUS/TCP *Unit Identifier* field. This allows rules to be developed to detect RTU address scans.

The vulnerabilities, exploits, and associated rules described in this section are relevant for MODBUS and other serial port protocols. Snort can be used for intrusion detection and prevention on systems using other serial port protocols as well.

5. MODBUS RTU/ASCII Snort implementation

MODBUS RTU/ASCII Snort is software developed to allow Snort to monitor and analyze MODBUS RTU and MODBUS ASCII traffic. The software can run on an existing PC connected to the serial link, such as on the PC hosting the human machine interface software. The software can also be run on single board industrial computers and placed in an inline or tap configuration to monitor MODBUS traffic. MODBUS RTU/ASCII Snort is a retrofit device intended to add Snort intrusion detection and prevention capabilities to previously installed MODBUS RTU/ASCII control systems.

Snort can be run in passive or inline mode. In passive mode Snort analyzes traffic for matches against a predefined set of rules. Packets which match rules are logged for review off line by a network or system administrator. In inline mode Snort acts an intrusion prevention system. Special rules, drop rules, are used to detect network traffic which should be dropped.

MODBUS RTU/ASCII Snort captures MODBUS RTU and MODBUS ASCII network traffic using an existing retrofit data logger [12]. Captured traffic is converted to MODBUS TCP/IP and transmitted over a closed virtual Ethernet network to allow Snort to capture the traffic. Snort parses the captured traffic to detect rule matches. Rule matches lead to logging and or dropping of packets.

5.1 MODBUS RTU/ASCII to MODBUS TCP/IP Conversion

Figure 2 shows the contents of MODBUS RTU and MODBUS ASCII protocol data units (PDU). The PDU contains a slave address (*ADDR*), a function code (*FC*), a payload, and a cyclic redundancy check (*CRC*) or linear redundancy check (*LRC*) check sum value. The *CRC* is used for MODBUS RTU transactions and the *LRC* for MODBUS ASCII transactions.

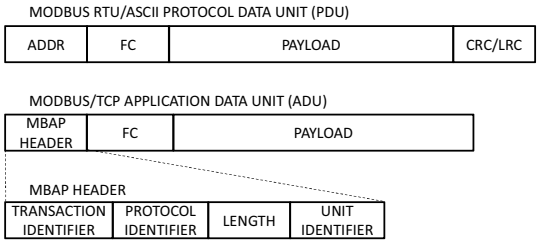

Figure 2. MODBUS RTU/ASCII to MODBUS TCP/IP conversion

The MODBUS/TCP application data unit (*ADU*), shown in Figure 2 includes a MBAP header, a function code (*FC*), and payload. The MBAP header includes a transaction identifier, a protocol identifier, a length field, and a unit identifier. The transaction identifier is a unique number for each MODBSUS/TCP query and response pair. The protocol identifier is a 2 byte field to identify the protocol encapsulated. Currently only one value is supported, 0, which indicates MODBUS. The length field indicates the length of the MODBUS ADU after the length field, i.e. the length of the unit identifier, the function code, and the payload. The unit identifier field is a 1 byte field to provide a unique ID for each connected slave device. The function code and payload in the MODBUS/TCP ADU are the same as the function code and payload in the MODBUS RTU/ASCII PDU.

Converting a MODBUS RTU/ASCII PDU to MODBUS/TCP ADU requires a few steps. MODBUS RTU/ASCII transactions are captured separately from the MTU and RTU. Transactions which originate from the MTU are queries and transactions which originate from the RTU are responses. The capture tool at the MTU keeps a linear count of queries. This value is placed in the ADU MBAP header transaction identifier field of the MODBUS/TCP query. The capture tool at the RTU keeps the last transaction identifier received from the MTU and places it in the response ADU. The protocol identifier field is always 0. The ADU length is calculated by adding the payload length and the unit identifier length (1 byte) and the function code length (1 byte). The MODBUS RTU/ASCII address (*ADDR*) is placed in the MODBUS/TCP unit identifier field. The function code and payload from the MODBUS RTU/ASCII PDU are copied to the same fields in the MODBUS/TCP ADU. The MODBUS RTU/ASCII CRC/LRC fields are not used in the MODBUS/TCP ADU. The TCP CRC is used for error checking in MODBUS/TCP. The ADU is encapsulated in a TCP packet for transmission over an Ethernet TCP/IP network.

5.2 MODBUS RTU/ASCII Snort software components

After conversion from MODBUS RTU/ASCII to MODBUS/TCP, packets are transmitted over an Ethernet network for Snort to capture and analyze. MODBUS RTU/ASCII Snort uses two virtual machines. One virtual machine (*VM1* in Figure 3) captures MODBUS RTU/ASCII traffic from the MTU or upstream direction. The second virtual machine (*VM2* in Figure 3) captures MODBUS RTU/ASCII traffic from the RTU or downstream direction. Captured traffic is converted to MODBUS/TCP and then transmitted from one virtual machine to the other. In passive mode, VM1 runs Snort to capture and analyze the traffic. In inline mode Snort is run on both virtual machines to enable drop rules to supervise traffic from either direction.

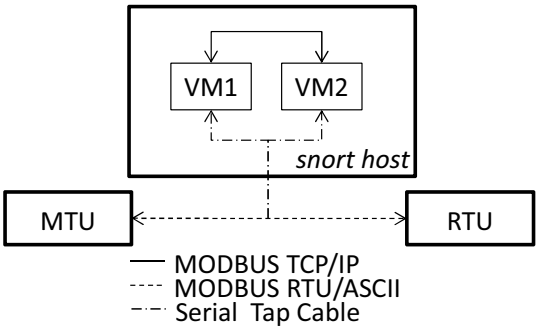

Figure 3. MODBUS RTU/ASCII Snort passive (tap) configuration

In the passive configuration a serial tap cable is used to capture traffic. Serial tap cables typically sample the RX pin of an RS-232 cable but do not include a TX pin and therefore cannot transmit. Figure 3 shows a passive configuration. The MTU's RX pin is provided to VM1 and the RTU's RX pin is provided to VM2. The passive configuration has the advantage of not adding delay to the packet transmission time from MTU to RTU. However, the passive configuration can only monitor transmissions, it cannot block transmissions.

Figure 4 shows a MODBUS RTU/ASCII Snort host in an inline configuration. In the inline configuration MODBUS RTU/ASCII traffic is captured by the first virtual machine, transmitted over Ethernet to the second virtual machine, and then forwarded to the endpoint by the Snort host. Placing the Snort host inline adds the ability to drop traffic at the cost of additional transmission delay. MODBUS RTU/ASCII systems typically operate between 1200 bits per second and 19200 bits per second (bps). The increased delay associated with the inline configuration is dominated by the delay associated with storing and forwarding a packet at serial port speeds. The Ethernet transmission delay has only a second order impact. This is true because 10 megabit per second Ethernet links are approximately 1000 times faster than common serial link speeds. The difference is more dramatic for gigabit Ethernet.

A laboratory scale gas pipeline control system was used to measure the delay added by the Snort host in inline mode. Latency added by the Snort host averaged 3.5 milliseconds per byte when operated at 9600 bits per second. For the maximum length MODBUS RTU packet (256 bytes) this equates to 903 milliseconds added latency. The acceptability of this value depends upon the polling rate of the HMI in a control system and is a system specific decision.

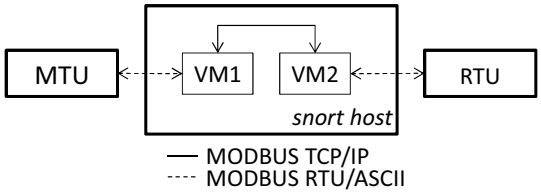

Figure 4. MODBUS RTU/ASCII Snort inline configuration

A significant speedup is available for the inline configuration by leveraging the YASIR method [13]. YASIR forwards bytes of a MODBUS packet through an intermediate node while simultaneously performing computations to make a pass or drop decision. Passed packets receive the benefit of the parallel decision and transmit logic. Dropped packets are not dropped but sabotaged by intentionally altering the CRC or LRC to cause the MODBUS appliance receiving the packet to ignore it. This method offers a significant speed-up over a traditional store and forward solution.

5.3 MODBUS RTU/ASCII Snort placement within a MODBUS RTU/ASCII Network

MODBUS RTU/ASCII Snort can be placed in multiple locations within the control system network.

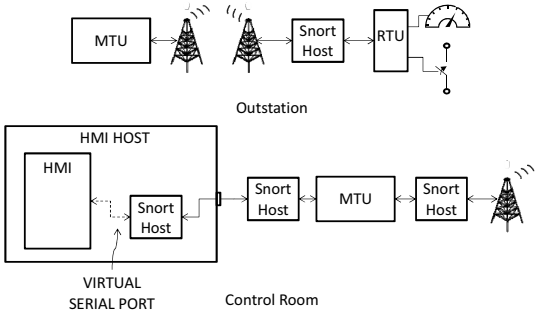

Figure 5. MODBUS RTU/ASCII Snort placement options

Figure 5 shows Snort host placement in the outstation (at the RTU) and in the control room. In all cases Figure 5 shows the Snort hosts in an inline configuration. A passive tap configuration is also supported. A passive tap configuration is shown in Figure 3.

At the outstation a Snort host can be used to capture and analyze MODBUS traffic before it is received by the RTU. The Snort host at the outstation can monitor traffic for illegal configuration commands sent to the RTU, illegal values written to setpoint registers, and attempts to overwrite device firmware. Intrusion traffic may originate at a compromised HMI host, MTU device, or from a device which has penetrated the radio network between the MTU and RTU.

Figure 5 shows three potential locations for MODBUS RTU/ASCII Snort in the control room. First, a Snort host may be added as a virtual device on the existing computer which hosts the human machine interface software. Second, a Snort host may be added on a standalone platform between the HMI host and the MTU. Finally, a Snort host may be added on a standalone platform between the MTU and the radio link. Each location offers different intrusion detection/prevention capabilities. All three locations can capture traffic from the RTU(s) and any intruders on the radio network between the MTU and RTU(s). Both the Snort hosts between the HMI and the MTU can capture traffic from a compromised HMI before the traffic reaches the MTU. The Snort hosts between the HMI and MTU are redundant and only one of the two is required. The choice of which to use, a virtual or standalone host, will primarily be made on cost and HMI host performance capabilities. The Snort host between the MTU and the radio link can capture traffic from a compromised HMI or MTU before it is transmitted to the RTU. The Snort host between the MTU and the radio link cannot protect the HMI from traffic from a compromised MTU. In summary, one Snort host between the HMI and MTU and one Snort host between the MTU and radio link provide the most intrusion detection/prevention coverage.

5.4 MODBUS RTU/ASCII Snort validation

MODBUS RTU/ASCII Snort was validated using rules available from Digital Bond, Inc. in their Quickdraw software package [6].

The Quickdraw tool includes 14 Snort rules for MODBUS/TCP implementations. Some rules found in the Quickdraw tool do not apply to MODBUS RTU and MODBUS ASCII systems. Quickdraw includes two rules to detect reads and writes from unsupported master IP addresses. Since MODBUS RTU and MODBUS ASCII do not have a source IP address these rules do not apply. Quickdraw also includes a rule to confirm the value of the MODBUS/TCP *Protocol Identifier* field is 0. This field also does not exist in MODBUS RTU and MODBUS ASCII and therefore the rule does not apply.

The other 11 rules were tested by developing a program to send illegal MODBUS RTU and ASCII queries and responses through MODBUS RTU/ASCII Snort. All 11 rules functioned correctly.

6. Future Works

The Quickdraw rules mentioned above represent a good start for rules for industrial control systems. Additional rules sets should be developed which perform deep packet inspection to confirm control system setpoints are within legal ranges. Denial of service and false response injection vulnerabilities associated with the radio attack described in [5] were discussed in section 4. A snort rule should be written to detect the presence of multiple responses associated with the same query to detect an exploit related to the false response injection vulnerability.

In addition to rules development, a YASIR like speed-up for MODBUS RTU/ASCII Snort inline configuration would significantly improve the feasibility of using the system as an intrusion prevention system.

Finally, MODBUS RTU/ASCII Snort was developed for MODBUS only. However the technology is applicable to DNP3 and other serial based industrial control system technologies. The tool should be extended for other protocols to demonstrate its flexibility.

7. Conclusions

MODBUS RTU/ASCII Snort is software to retrofit serial based industrial control systems to add Snort intrusion detection and intrusion prevention capabilities. Such a system is motivated by the existence of worms such as Stuxnet, which has been shown to have altered remote terminal unit firmware over the communication link between a human machine interface (HMI) host and the remote terminal unit, after compromising the HMI host. Additionally, industrial radios used to wirelessly extend serial links in many critical industries have been shown to be vulnerable to intrusions [5]. MODBUS RTU/ASCII Snort converts MODBUS traffic on a serial link to Ethernet TCP/IP traffic and transmits it on a closed private network to enable Snort based intrusion and intrusion prevention features. MODBUS RTU/ASCII Snort was validated in simulation and in the laboratory with industrial control systems which use commercial hardware and software to control laboratory scale fully functional physical processes.

8. References

[1] North American Electric Reliability Corporation. Standard CIP–005–4a - Cyber Security - Electronic Security Perimeter(s). January 2011. http://www.nerc.com/files/CIP-005-4a.pdf

[2] Caswell, B. Bealeand, J., Foster, J. and Faircloth, J. "Snort2.0 Intrusion Detection," Syngress, Feb. 2003.

[3] O'Murchu, L. Last-minute paper: An indepth look into Stuxnet. The 20th Virus Bulletin International Conference. September 29 – October 1, 2010, Vancouver, BC, Canada.

[4] Falliere, N., Murchu, L., Chien, E., W32.Stuxnet Dossier, Version 1.3. Symantec Security Repsonse. November 2010. http://tinyurl.com/36y7jzb

[5] Reaves, B., Morris, T. Discovery, Infiltration, and Denial of Service in a Process Control System Wireless Network. IEEE eCrime Researchers Summit. October 20-21, 2009. Tacoma, WA

[6] Peterson, D. "Quickdraw: Generating Security Log Events for Legacy SCADA and Control System Devices," Conference for Homeland Security, 2009. CATCH '09. Cybersecurity Applications & Technology , vol., no., pp.227-229, 3-4 March 2009

[7] Roosta, T. Nilsson, D., Lindqvist, U. Valdes, A. An intrusion detection system for wireless process control systems. 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, pp866-872, Atlanta, GA, 2008.

[8] Cheung, S., Valdes, A. Communication Pattern Anomaly Detection in Process Control Systems. IEEE International Conference on Technologies for Homeland Security. Waltham, MA. May 11-12, 2009.

[9] Valdes, A., Cheung, S. Intrusion Monitoring in Process Control Systems. Proceedings of the 42nd Hawaii International Conference on System Sciences. 2009.

[10] Rrushi, J., Kang, K., Detecting Anomalies in Process Control Networks. IFIP International Federation for Information Processing 2009.

[11] Gao, W., Morris, T., Reaves, B., Richey, D. On SCADA Control System Command and Response Injection and Intrusion Detection, in the Proceedings of 2010 IEEE eCrime Researchers Summit. Dallas, TX. Oct 18-20, 2010.

[12] Morris, T., Pavurapu, K. A Retrofit Network Transaction Data Logger and Intrusion Detection System for Transmission and Distribution Substations, in the Proceedings of 2010 IEEE International Power and Energy Conference. Kuala Lumpur, Malaysia. Nov 29, 2010.

[13] Patrick P. Tsang, Sean W. Smith: YASIR: A Low-Latency, High-Integrity Security Retrofit for Legacy SCADA Systems. SEC 2008: 445-459.