Industrial Control System Cyber Attacks

Thomas H. Morris[1], Wei Gao
Mississippi State University
Mississippi State, MS, USA
[1]*morris@ece.msstate.edu*

This paper presents a set of attacks against SCADA control systems. The attacks are grouped into 4 classes; reconnaissance, response and measurement injection, command injection and denial of service. The 4 classes are defined and each attack is described in detail. The response and measurement injection and command injection classes are subdivided into sub-classes based on attack complexity. Each attack described in this paper has been exercised against industrial control systems in a laboratory setting.

Industrial Control System. Threat Model. Taxonomy.

1. INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) systems are computer-based industrial control systems which interconnect and monitor remote physical processes. SCADA systems collect data from remote facilities about the state of the physical process and send commands to control the physical process creating a feedback control loop. SCADA systems are widely used in chemical processing, petroleum refining, electrical power generation and distribution, water purification and distribution, intelligent buildings and nuclear plants.

There have been several real-world documented incidents and cyber attacks affecting SCADA systems, which clearly illustrate critical infrastructure vulnerabilities. These reported incidents demonstrate that cyber attacks on SCADA systems might produce a variety of financial damage and harmful events to humans and their environment.

This paper presents a set of 17 attacks against SCADA control systems. The attacks are grouped into 4 classes; reconnaissance, response and measurement injection, command injection and denial of service. The response and measurement class is divided into Naive Malicious Response Injection (NMRI) and Complex Malicious Response Injection (CMRI) sub classes. The command injection class is divided into Malicious State Command Injection (MSCI), Malicious Parameter Command Injection (MPCI) and, Malicious Function Code Injection (MFCI) sub classes.

Each attack presented in this paper was implemented and tested in a laboratory setting (Morris 2011). Attacks were targeted against fully functional SCADA control systems which model a gas pipeline and a water storage tank using commercial control system hardware and software.

2. RELATED WORK

Many works have been published which introduce cyber attacks or sets of cyber attacks against industrial control systems.

Vulnerabilities of a SCADA system which monitors and controls the Gignac irrigation canal system located in South France are discussed in (Amin 2012). Jie Yan et al (Yan 2011) identify vulnerabilities and develop cyber attack scenarios in wind farm SCADA system. Dillon Beresford (Beresford 2011) exploiting processes such as reconnaissance and fingerprint attacks, replay attacks, authentication bypass and remote exploitation on the Siemens Simatic S7 PLC. A taxonomy of attacks against energy control systems is proposed in (Fleury 2009). A set of SCADA cyber attacks against a MODBUS TCP test bed are presented in (Mallouhi 2011). Data integrity attacks and a Denial of Service attack on SCADA control system are presented in (Sridhar 2010). The data integrity attacks include the Min attack and the Max attack (Yu-Hu Huang 2009). False data injection attacks against state estimation algorithms in electric power systems are discussed in ((Liu 2009) and (Le Xie 2010). An event buffer overflow attack on the DNP3 protocol is introduced in (Dong Jin 2011). These works together justify the need for work to identify classes of vulnerabilities and examples of such vulnerabilities for industrial control systems.

This work offers a set of 17 attacks against industrial control systems which use the MODBUS communication protocol. The attacks in this paper are grouped by type to provide insight into the various types of threats to industrial control systems.

3. ATTACKS AGAINST INDUSTRIAL CONTROL SYSTEMS

In this section a set of cyber attacks against industrial control systems (ICS) are grouped into four attack classes; reconnaissance, response and measurement injection, command injection, and denial of service. Table 1 lists 17 individual attacks described in this paper and lists each attack's sub-class. This section defines the four classes and their sub-classes and provides detailed descriptions of each attack listed in Table 1.

3.1 Reconnaissance Attacks

Reconnaissance attacks gather control system network information, map the network architecture, and identify the device characteristics such as manufacturer, model number, supported network protocols, system address, and system memory map. This section describes 4 reconnaissance attacks against MODBUS servers; the address scan, the function code scan, the device identification attack, and the points scan. The address scan discovers ICS servers connected to a network. The function code scan identifies supported network operations which can be performed for an identified server. The device identification attack allows an attacker to learn a discovered device's vendor name, product code, major and minor revision, et cetera. The points scan allows the attacker to build a device memory map.

Attacks described in this section were implemented against MODBUS servers because MODBUS is an open standard and is a popular network protocol used for ICS devices. While these attacks are known to function against MODBUS servers the vulnerabilities the attacks exploit are general enough to also likely be found in other network protocols used by ICS.

Industrial control system users often develop standard hardware, software, control scheme parameter configurations which are duplicated throughout a control system. For example, an electric transmission system may use a standard panel which includes the same protective relays used at many substations throughout a single company's transmission system. A second example is pump stations for a gas pipeline. Pump stations are distributed along the gas pipeline to ensure product flow. Programmable logic controllers (PLC) and the programmation on those controllers will be similar throughout the system.

Table 1: *List of Attacks against MODBUS Industrial Control Systems*

Attack Index	Name	Classification
1	*Address Scan*	*Reconnaissance*
2	*Function Code Scan*	*Reconnaissance*
3	*Device Identification*	*Reconnaissance*
4	*Naïve Read Payload Injection*	*NMRI*
5	*Invalid Read Payload Size*	*NMRI*
6	*Naïve False Error Response*	*NMRI*
7	*Sporadic Sensor Measurement Injection Attack*	*NMRI*
8	*Slope Sensor Measurement Injection*	*CMRI*
9	*High Slope Measurement Injection*	*CMRI*
10	*High Frequency Measurement Injection*	*CMRI*
11	*Altered System Control Scheme*	*MSCI*
12	*Altered Actuator State*	*MSCI*
13	*Altered Control Set Point*	*MPCI*
14	*Force Listen Only Mode*	*MFCI*
15	*Restart Communication*	*MFCI*
16	*Invalid Cyclic Redundancy Code (CRC)*	*DOS*
17	*MODBUS Slave Traffic Jamming*	*DOS*

The combined results of the address scan, function code scan, the device identification attack, and the points scan can be used to generate a signature for MODBUS servers common to a particular company, use case, or vendor. This signature can then be used to build maps of discovered systems by company, by use case, or by vendor. Such signatures can also be used to build a database of vulnerabilities and exploits for each aforementioned category.

Attack 1 is the Address Scan. MODBUS servers use either an IP address for MODBUS/TCP systems or a single byte address for MODBUS RTU and ASCII systems. Attackers can perform an address scan to identify MODBUS server addresses which are in use. Each MODBUS server is assigned a unique address. MODBUS systems typically have a static configuration in which the number of servers does not change and the address assignment of the individual clients does not change. MODBUS/TCP servers can have any legal IP address. MODBUS/TCP servers listen on

TCP port 502. Equation 1 defines the legal address range for MODBUS RTU and ASCII systems.

$$ADDR \in \{0,..,247\} \quad (1)$$

The MODBUS protocol requires addressed servers to return a response code after being addressed by a query. The response may be acknowledgement of a successful transaction or indicate an error message. No response will be received for MODBUS queries addressed to nonexistent servers. To identify MODBUS servers an attacker can send MODBUS queries to each legal MODBUS address and wait for any response. Note, for MODBUS RTU and ASCII systems the 0 address is for broadcast commands. No response is sent for broadcast commands and therefore this address would typically not be used for address scan attacks.

Attack 2 is the Function Code Scan. After MODBUS server addresses are identified an attacker may wish to scan servers to identify supported function codes. The MODBUS function code field is a single byte. MODBUS specifications define 4 types of function codes; public function codes (PFC), user defined function codes (UFC), reserved function codes (RFC), and error function codes. Equation 2 lists the set of public function codes.

$$PFC \in \{1,2,3,4,5,6,7,8,11,12,15,16,17,20,21,22,23,24,43\} \quad (2)$$

User defined function codes must be in the range defined by equation 3.

$$UFC \in \{65,\ldots,72,100,\ldots,110\} \quad (3)$$

Reserved function codes are codes in the public code space which have been used by legacy devices and which are not supported as public codes. This set is most often empty. Reserved function codes are in the set defined by equation 4.

$$RFC \in \{9,10,13,14,41,42,90,91,125,126,127\} \quad (4)$$

When a MODBUS query generates an error at the MODBUS server an error function code is returned in the response. The error function code is the query function code + 0x80. An error function code exists for all legal function codes regardless of whether the underlying public, user defined, or reserved function code is supported by the MODBUS server. Therefore MODBUS function code scans should not scan the error function codes.

An attacker can perform a MODBUS function code scan by sending a query to all function codes in the sets defined by equations 2-4. MODBUS query payloads vary by function code. However, an attacker need not form a proper payload for each function code to determine if a function code is supported by a MODBUS server. Function code scans can be grouped into two categories by the function code scan attack. If the function code is not supported an exception code 1 (invalid function code) response will be returned. All other responses, whether indicating an error or transaction success, indicate the function code is supported by the targeted server.

Attack 3 is the Device Identification attack. Attackers can also fingerprint remote devices to learn specific information such as the vendor name, product code, and revision number. This information can be used to search for known vulnerabilities in exploit databases such as Exploit Database (EDB) (Offensive Security 2013).

MODBUS servers may implement a function code to allow a client to read device identification information. For MODBUS RTU and ASCII servers function code 0x11 allows an attacker to retrieve the current run status and additional information which is device specific. Device specific contents may include sensitive information.

MODBUS servers implement second read device identification function code, 0x2B. There are 3 Read Device ID object types basic, regular, and extended. Basic information is mandatory for all MODBUS servers and includes the vendor name, the product code, and the major and minor revision. The regular information is optional and includes the vendor uniform resource locator (URL), the product name, the model name, and the user application name. The extended information is optional and includes user defined objects.

3.2 Response and Measurement Injection Attacks

Industrial control systems commonly use polling techniques to continuously monitor the state of a remote process. Polling takes the form of a query transmitted from the client to the server followed by a response packet transmitted from the server to the client. The state information is used to provide a human machine interface to monitor the process, to store process measurements in historians, and as part of feedback control loops which measure process parameters and take requisite control actions based upon process state.

Many industrial control system network protocols lack authentication features to validate the origin of packets. This enables attackers to capture, modify, and forward response packets which contain sensor reading values. Industrial control system protocols also often take the first response packet to a query and reject subsequent responses as erroneous. This enables to craft response packets and use timing attacks to inject the responses into a network when they are expected by a client.

Response injection attacks take 3 forms. First, response injection attacks can originate from control of a programmable logic controller or remote terminal unit, network endpoints which are the servers which respond to queries from network clients. Second, response injection attacks can capture network packets and alter contents during transmission from server to client. Finally, response injections may be crafted and transmitted by a third party device in the network. In this case, the response there may be multiple responses to a client query and the invalid response may assume prominence due to exploiting a race condition or due to secondary attack such as a denial of service attack which stops the true server from responding.

In this section multiple response injection attacks are discussed. The response injection attacks are grouped into two categories; 1) Naive Malicious Response Injection (NMRI) attacks and 2) *Complex Malicious Response Injection (CMRI) attacks.*

Naive Malicious Response Injection (NMRI) attacks lack sophistication. NMRI attacks leverage the ability to inject response packets into the network but lack information about the process being monitored and controlled. NMRI attacks may send invalid payloads. For example, an attacker may know have performed a set of reconnaissance attacks to learn system addresses, function codes, and memory map, but lack specific details on what the monitored process is or lack details on valid data contents for each point found on a server. In this case, the attacker may craft a response injection attack with a payload of all zeroes, all negative numbers, all very large numbers, or other likely invalid contents. Alternatively, NMRI attacks may be based on limited process information. For example, an attacker may know process details such as process limits or valid contents for each point found on a server but not have the capability to craft more sophisticated attacks. For example, an attacker may be able to cause an alarm.

Attack 4, Naïve Read Payload Size, is the first NMRI attack. The Naïve Read Payload Size attack is based only on network protocol knowledge. MODBUS read coil, discrete input, holding register, and input register queries include a quantity field to specify the number of objects to be returned by the server. An NMRI attack can craft malicious responses which include the correct quantity of returned objects which are all zeroes or all ones. Alternatively, the NMRI can return the correct number of requested objects with random contents. Random contents is particularly interesting for the read coils and discrete inputs cases since these returned values are specified to be limited to only 0x00 and 0xFF for each coil or discrete input.

Attack 5, Invalid Read Payload Size, is an NMRI attack in which the requested number of objects from the read coil, discrete input, holding register, or input register query is ignored. The response payload is either larger or smaller than the requested amount. The response payload may be formed by trimming or extending a valid payload, or by creating a payload with zeroes, ones, or random bytes.

Attack 6, Naïve False Error Response, is an NMRI attack in which falsified error responses are returned to the client after a read command. For MODBUS an error packet is formed by adding 0x80 to the function code followed by an exception code. Read command exception codes are limited to 0x01, 0x02, 0x03, and 0x04. This NMRI attack can send random exception codes which fall in the legal range or send random exception codes which are outside the legal range.

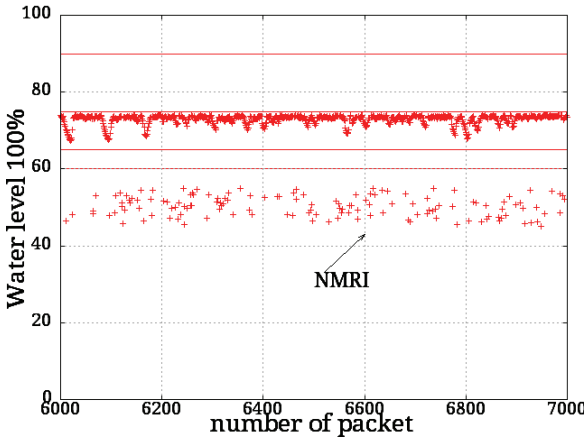

***Figure 1**: NMRI Sporadic Sensor Measurement Injection*

Attack 7, Sporadic Sensor Measurement Injection, is an NMRI attack which sends sporadic false process measurements outside the bounds of the high (H) and low (L) control set points while not outside the alarm set point range formed by the high high (HH) and low low (LL). During normal operation, when a measurement reaches the H set point the pump is turned off. When a measurement reaches the L set point a pump is turned on. For the attack used for this work falsified measurements are sporadically sent to the MODBUS client. Both the water tank and gas pipeline systems regularly have measurements outside the H and L limits due to a time delay between measuring the gas pressure or water tank level and sending the command to turn off the pump adding water or gas to the physical process. This makes developing an automated intrusion detection rule based strictly on the H and L limits difficult. However, this NRMI attack differs from the ordinary out of bounds situation in that the response injection packets are sporadic in nature. Figure 1 shows a sporadic sensor measurement injection attack. A line of normal measurements show a water level trend varying between 60 and 60 percent full. Sporadic measurements are shown between the 40 and 60 percent full range.

The sporadic measurements appear at seemingly random times and are obviously not part of a trend.

Complex Malicious Response Injection (CMRI) attacks add a level of sophistication above that of the NMRI attacks. CMRI require understanding of the cyber physical system being attacked. CMRI attacks attempt to mask the real state of the physical process being controlled to negatively affect the feedback control loop managing the cyber physical system.

Attack 8, Calculated Sensor Measurement Injection, is a CMRI attack in which calculated process measurements are injected. This attack simulates a process measurement trend such as a water level or gas pressure increasing or decreasing. For example, an attack can inject falsified response packets which simulate a water level trend increasing from a normal level such as 20% to 100%. Such an attack would cause the operator to turn off the water pump while the actual water level is 20% full. This attack requires system knowledge and an accurate model of the system being attacked.

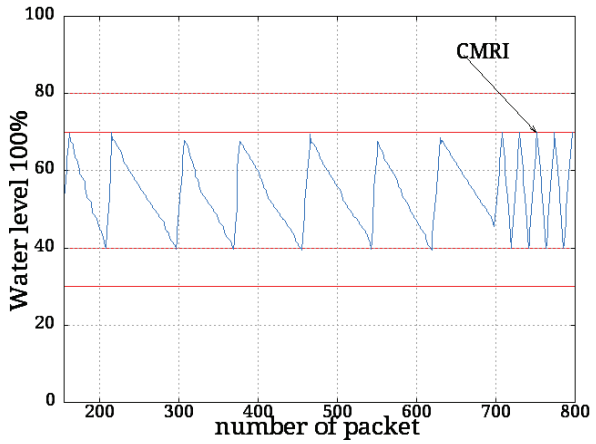

***Figure 2**: CMRI High Frequency Measurement Injection*

Attack 9, Replayed Measurement Injection, is a CMRI attack in which means the attacker replays captured process measurements to the client to give the operator the impression the system running normally.

Attack 10, High Frequency Measurement Injection, is a CMRI attack in which the frequency of process measurement changes is increased beyond a normal rate. This attack is a special case of attack 8. For example, the falsified responses may indicate a fast rising water level or fast decreasing gas pressure. This attack scenario may appear to match the system behavior common at a different time of a day and may cause an operator to misconfigure the system to handle the falsified demand. However, in the case of water or gas distribution, increased pump speed may lead to overflow since demand is actually lower. Figure 2 shows a graph of changing water storage tank level measurements before and during a High Frequency Measurement Injection Attack. In the figure, the frequency of liquid level changes is normal at first (the left side of the graph) and then during the attack (right side of the graph) the liquid level rises and falls more rapidly. Such a change may simulate a period of high demand.

Attacks 8-10, are designed to appear like normal process functionality. These attacks can be used to mask other process changes such changes to process state through malicious command injection attacks. Because these attacks project a state of normalcy they are very difficult to detect.

3.3 Command Injection Attacks

Command injection attacks inject false control and configuration commands into a control system. Human operators oversee control systems and occasionally intercede with supervisory control actions. Hackers may attempt to inject false supervisory control actions into a control system network. 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.

The potential impacts of malicious command injections include interruption process control, interruption of device communications, unauthorized modification of device configurations, and unauthorized modification of process set points.

As mentioned in the response injection discussion above much industrial control system network protocols lack authentication features to validate the origin of packets. This enables attackers to capture and alter command packets. Additionally, attackers can craft original command packets and directly inject them into the control system network.

In this section multiple command injection attacks are discussed. The command injection attacks are grouped into three categories; Malicious State Command Injection (MSCI) attacks, Malicious Parameter Command Injection (MPCI) and, Malicious Function Code Injection (MFCI).

MSCI attacks change the state of the process control system abnormally to drive the system from a safe state to a critical state by sending malicious commands to remote field devices. MSCI attacks may require a single injected command or multiple injected commands.

Typically actuators, such as switches or valves, connected to physical processes are connected to a digital or analog output connected to a remote terminal unit (RTU) or intelligent electronic device (IED). Each output connects to the cyber system by modelling it as a digital point in a register. Changing the state of a bit or bits in such a register has an immediate impact on the physical actuator. For example, a pump may have an ON/OFF mechanism which is changed by writing a value to a bit a in a register on a remoter terminal unit (RTU). Such registers can be manipulated by network protocol write commands. For example, the MODBUS protocol includes write coil and write register commands. An attacker who understands a device's implementation specifics including a memory map can craft a command to alter actuator states.

Many control systems allow operators to change between automatic a manual control modes. Attack 11, Altered System Control Scheme, is a MSCI attack which changes this control mode. For example for a gas pipeline control system from automatic mode to manual mode and then turns on a pump which increases the pressure within the pipe. In this laboratory scale control system programmable logic controller (PLC) with MODBUS-RTU server is connected to a pump, a solenoid, and a pressure meter. In automatic mode the user sets a target pressure and the PLC switches the pump between ON/OFF mode, and switches the solenoid between OPEN/CLOSED modes to fire the solenoid to open and close the relief valve which in turn controls the pressure in the pipe. In manual mode the pump state and solenoid state are no longer controller by the PLC program and become directly controlled by register values store in the PLC. MODBUS commands can be used to change the values stored for system control mode, pump state, and solenoid state. To implement the attack first a write register (MODBUS function code 03) command to address 0xABCD is used to switch the control mode to manual. Next, a write register command to address 0xABCD is used to turn on the pump. For the gas pipeline control system a pipe pressure above 60 PSI is considered a critical state. Pressure above this value will potentially damage system components. Placing the system in manual mode and turning on the pump causes the pressure to climb toward this critical value. An operator monitoring the system with a human machine interface should notice the climbing pressure and can take control to correct the issue. Additionally, the rising gas pressure may trigger a process alarm to gain an operators attention.

Attack 12, Altered Actuator State, is an MSCI attack scenario which changes system actuator states one time. For the gas pipeline system Altered Actuator State attacks include command injections which turn the pump on or off and command injections which open or close the relief valve. For the water storage tank system an Altered Actuator State attack was implemented to turn the pump on or off.

Attack 13, Altered Control Set Point, is an MPCI attack which changes device set points. Set points are typically used to provide variable control over a system. For example the water storage tank system uses an ON/OFF control scheme to keep the amount of liquid in a tank between a low set point and a high set point. A level sensor continuously monitors liquid level as a percentage of tank full and turns a pump on an off to add liquid to the tank. A MODBUS write register command was used to change both the high and low set points. This attack also alter alarm values stored in PLC registers to disable alarms by changing set points liquid level alarms to values in line with the altered high and low set points.

Application layer protocols sometimes include commands which have unintended consequences when used by attackers. MODBUS function code 8, named "Diagnostics", includes 3 sub-function codes, commands, which can be used to disrupt the client server communication link. The original intent of the diagnostics function code was to provide a means to diagnose and address communication issues. The diagnostics command is only required for serial port MODBUS systems.

Attack 14, Force Listen Only Mode, causes a MODBUS server to no longer transmit on the network. The diagnostic function code, MODBUS function code 8, includes a sub function code to force a MODBUS server into listen only mode. Many industrial control systems use polling technique in which the master node, such as human machine interface (HMI) software, polls the MODBUS servers periodically for data. The HMI displays data to human operators who may then take supervisory control actions based upon the current state of the system. There also exist wide area control schemes which poll MODBUS servers for data to support automated control actions. A MODBUS server which is placed in listen only mode by an attacker will not respond to queries and in the situations described will result in a loss of system visibility and control.

Attack 15, Restart Communication, sends a command which causes the MODBUS server to restart which leads to a temporary loss of communication. The diagnostic function code includes a sub function code to restart the remote device and cause it to execute its power up diagnostic tests. This loss of communication leads to a temporary inability to observe and control the process. During the restart period local control from the program running on the field device is also lost. Multiple successive restart communication attacks

can lead to a near complete loss of communication with and control over the process.

3.4 Denial of Service Attacks

Denial of Service (DOS) attacks against industrial control system attempt to stop the proper functioning of some portion of the cyber physical system to effectively disable the entire system. As such DOS attacks may target the cyber system or the physical system. DOS attacks against the cyber system target communication links or attempt to disable programs running on system endpoints which control the system, log data, and govern communications. DOS attacks against the physical system vary from the manual opening or closing of valves and switches to destruction of portions of the physical process which prevent operation. This work concentrates on DOS attacks against the communication system.

MODBUS systems may be MODBUS TCP/IP, MODBUS RTU, or MODBUS ASCII. MODBUS TCP/IP is a routable protocol which allows other devices to initiate DOS attacks targeted to a victims IP address. MODBUS RTU and ASCII use RS-232 or RS-485 physical layers. These serial port protocols are considered non-routable. However, MODBUS RTU and ASCII devices are vulnerable to certain DOS attacks. RS-232 is a point to point protocol used for MODBUS connections over short distances, typically less than 20 meters. However, control systems often connect a remoter terminal unit or master terminal unit to a wireless radio using RS-232 then use the radio to transmit across longer distances. Such radio links can be penetrated by attackers and can therefore be the source of a DOS attack. RS-485 serial links allow multipoint network topologies. In these cases a device on the network can become infected with malware and then the infected device can initiate a DOS attack against other devices on the network.

Traffic jamming is a class of DOS attacks in which high volumes of traffic are sent to a network endpoint. Attackers attempt to overwhelm the endpoint by either sending transmissions faster than they can be processed or by sending packets crafted to cause software errors which generate exceptions which crash the network stack, the running program, or the operating system of the targeted device.

Attack 16, Invalid Cyclic Redundancy Code (CRC), injects large volumes of MODBUS packets with incorrect CRC. Packets with invalid CRC are rejected by both MODBUS servers and clients. The victim device must check the CRC of each packet. A flood of packets with invalid CRC can overwhelm a device and cause it to crash or the flood may stop communication with other legitimate devices via loss of ability to transmit and/or receive packets.

Attack 17, MODBUS Master Traffic Jamming, is a traffic jamming attack against the medium access control (MAC) layer in which a non-addressed slave transmits out of turn. MODBUS RTU and ASCII systems often are configured with a single master connected to multiple slaves. When there are multiple slaves only the addressed slave should respond to master queries. For RS-485 systems, in both the 2-wire and 4-wire cases, the slave transmit wire is shared by all slaves attached to the bus. In this case, a slave transmitting out of turn will cause a legitimate slave's transmission to be garbled and lost and result in a timeout and retransmission by the master. A MODBUS Slave Traffic Jamming Attack against a RS-232 system with wireless radio between the master and slave node is described in (Reaves 2009). In this attack a wireless penetrator transmits continuously. The proprietary wireless radio includes a carrier sense back off arbitration scheme which causes legitimate slaves to continuously wait for a clear line to transmit. In laboratory experiments, attackers were able to force a legitimate slave to stay idle ad infinitum.

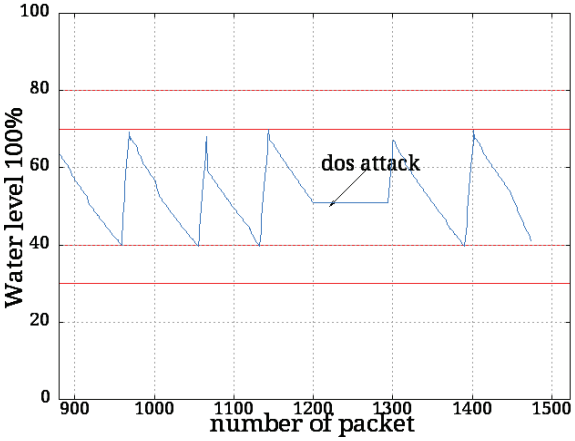

***Figure 3**: MODBUS Slave Traffic Jamming*

Figure 3 plots water level measurements observed by a HMI connected to the MODBUS master. The MI continuously queries the slave to read the water level. For this experiment the water storage tank was set to keep the water level in the tank between 40% and 70% full by cycling a water pump which fills the tank. The tank was configured to continuously drain water during the experiment. During normal operation the HMI sees the water level rises to the high set point when the pump is on and drops to the slow set point when the pump is off. Figure 3 shows the impact of the MODBUS Slave Traffic Jamming Attack from the perspective of the HMI. When the attack starts the HMI no longer receives responses to its water level queries and therefore the water level in the plot no longer changes. This loss of process visibility can cause an operator or automated algorithm to misoperate the system. During this attack the master is also no able to transmit commands. As such the operator may notice that something is wrong but is not able

to send commands to the remote system during the attack.

4. FUTURE WORK AND CONCLUSIONS

This work provides detailed descriptions of 17 attacks against SCADA control systems which use the MODBUS communication protocol. The attacks presented are grouped into 4 classes; reconnaissance, response and measurement injection, command injection and denial of service. Each attack described in this paper has been exercised against industrial control systems in a laboratory setting. The laboratory SCADA control systems are implemented using commercial hardware and software. While the attacks presented are limited to MODBUS based systems the classes of attacks presented are applicable to industrial control systems of all types regardless of communication protocol. A data set has been collected which includes time stamps, captured communication protocol parameters, system state information. Each tuple in the data set has been marked with the type of attack associated with the data point. In future work this data set will be used to validate signature based, specification based, and anomaly based intrusion detection systems designed to detect attacks against industrial control systems.

5. REFERENCES

Morris, T.; Srivastava, A.; Reaves, B.; Gao, W.; Pavurapu, K.; Reddi, R., A Control System Testbed to Validate Critical Infrastructure Protection Concepts, International Journal of Critical Infrastructure Protection. Elsevier. 2011.

Amin, S.; Litrico, X.; Sastry, S.; Bayen, A. M., Cyber Security of Water SCADA Systems-Part I: Analysis and Experimentation of Stealthy Deception Attacks, *IEEE Transactions on Control Systems Technology,* 2012.

Jie Yan; Chen-Ching Liu; Govindarasu, M.; Cyber intrusion of wind farm SCADA system and its impact analysis, IEEE PES Power Systems Conference and Exposition (PSCE), pp.1-6, 2011

Beresford, D; Exploiting Siemens Simatic S7 PLCs, Black Hat USA, July 8, 2011

Fleury, T.; Khurana, H.; Welch, V., Towards A Taxonomy Of Attacks Against Energy Control Systems, Critical Infrastructure Protection II, The International Federation for Information Processing, Volume 290. Springer US, 2009.

Mallouhi, M.; Al-Nashif, Y.; Cox, D.; Chadaga, T.; Hariri, S.; , A Testbed for Analyzing Security of SCADA Control Systems (TASSCS), IEEE PES Innovative Smart Grid Technologies (ISGT), pp.1-7, 2011.

Sridhar, S.; Manimaran, G.; , Data integrity attacks and their impacts on SCADA control system, IEEE Power and Energy Society General Meeting, pp.1-6, 2010.

Yu-Hu Huang, Alvaro A. Cardenas, et al, Understanding the Physical and Economic Consequences of Attacks on Control Systems, Elsevier, International Journal of Critical Infrastructure Protection 2009.

Le Xie; Yilin Mo; Sinopoli, B., False Data Injection Attacks in Electricity Markets, First IEEE International Conference on Smart Grid Communications (SmartGridComm), pp.226-231, 2010.

Liu, Y; M. Reiter, K; and Ning. P; False data injection attacks against state estimation in electric power grids, The 16th ACM Conference on Computer and Communications Security, 2009.

Dong J.; Nicol, D.M.; Guanhua Y., An event buffer flooding attack in DNP3 controlled SCADA systems, Winter Simulation Conference (WSC), pp.2614-2626, 2011.

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.

Offensive Security. The Exploit Database. http://www.exploit-db.com/ (Accessed July 22, 2013)