Discovering Cyber Vulnerabilities in SCADA Control System via Examination of Water Treatment Plant in Laboratory Environment

Puneet S. Muddu

University of New South Wales at the Australian Defence Force Academy

Increased connectivity of critical infrastructure to public networks has resulted in an increase in the number of threats faced by SCADA (Supervisory Control and Data Acquisition) systems. A SCADA system is used to monitor and remotely control critical infrastructure such as traffic control system, electricity distribution, water storage and treatment plants and transportation. The entities of the control systems communicate via legacy communication protocols like MODBUS and often have a high level of automation. The connectivity and automation while essential to the function of a SCADA system, these introduce a myriad of cyber vulnerabilities into the system. This paper gives a brief overview of the cyber vulnerabilities that have been identified in SCADA systems by reviewing related attacks; experiments have been be conducted on a Water Treatment Plant in a laboratory environment using cyberattack techniques identified to have a high success rate in SCADA environments. The techniques involved in completing attacks on the SCADA system will be included in a Laboratory Kit such that they can be used by future students, in order to educate and enhance awareness of cyber vulnerabilities in SCADA systems.

Contents

I. Introduction 2
II. Literature Review 2
III. Related Work 4
IV. Cyber Attack Techniques 4
V. Discovering Cyber Vulnerabilities in Water Treatment Plant 5
   A. Aims and Motivation 5
   B. Experimental Setup 5
   C. Tools 5
   D. Experiments 5
      1. Lab 1 – Reconnaissance 5
      2. Lab 2 – Command Injection attack 7
      3. Lab 3 – Response Injection attack 8
      4. Lab 4 – Denial of Service (DoS) 9
   E. Experimental results and Conclusion 9
VI. Future work 9
VII. Conclusion 9

References 10

Nomenclature

ICS = Industrial Control System
HMI = Human Machine Interface
PLC = Programmable Logic Controller
RTU = Remote Terminal Units
SCADA = Supervisory Control and Data Acquisition

1 Engineering Project, School of Engineering & Information Technology. ZEIT4500/4501.
I. Introduction

SCADA refers to a collection of servers that work in conjunction with sensors, PLCs and field devices in order to monitor and remotely control critical infrastructure processes (CERT 2012). The critical systems can include traffic control system, electricity distribution, water storage and treatment plant and transportation. Robust functionality of these systems directly correlates to social cohesion, economic prosperity and public safety (TISN 2016). For quicker access and for remote operability, these systems are increasingly being connected to corporate networks via the Internet.

The information flow in a SCADA system takes place via communication protocols such as MODBUS over communication channels. The communication protocol was developed with reliability as a core requirement as opposed to security and the channels were developed to span large distances based on system requirements. The critical nature of SCADA systems, the automation and the added connectivity to the internet makes them vulnerable to cyber-attacks.

In a 2016 report from FireEye showed that there was an increase from 149 ICS vulnerabilities disclosed between Jan 2000 to Dec 2010 to 1,552 in April 2016. This upward trend together with the slow patch time and lack of vendor fixes (less than 33%) identifies the growing need in the industry for vulnerability research in ICS. (FireEye 2016)

This paper gives a brief overview of the cyber vulnerabilities that have been identified in SCADA systems by classifying threat actors and related attacks. Using related work in SCADA environments and by exploring cyber-attack techniques, laboratory tests/experiments that have a high success rate have been built to run on the SCADA laboratory environment at ADFA.

II. Literature Review

The focus of the literature review is on Cyber vulnerabilities that exist in SCADA control system. By reviewing the SCADA architecture and exploring various attacks on these systems; the impact of this project can be identified.

SCADA comprises of three levels as seen in Figure 1; level 0 consists of sensors and actuators. The sensors are used to collect data about the system (pressure monitors, water level gauges, and laser sensors) and actuators are used to control the system’s state: pumps, motors, etc. Level 1 comprises of PLCs/RTUs; these control and collect information from the first level which is utilised to determine the system’s state. Level 2 of the SCADA system is supervisory control; this level communicates with the RTUs to complete tasks such as turning pumps on/off or opening gates as required; additionally these can also be used to send read queries to RTU registers that contain system parameters such as pressure levels, water levels etc. Profibus, Fieldbus and Modbus are some communication protocols that can be used to accomplish this. The human-machine interface (HMI) also resides in this level. The HMI is used by an operator to display the sensor data collected by the supervisory control. The HMI usually contains a visual representation of the system and the operation of the subsystems; it is used to change parameters and states within the SCADA system.

![Figure 1 SCADA Architecture (adapted from FireEye 2016)](image-url)
PLCs and RTUs are mostly affected by cyber-attacks due to their dependency on equipment like operating systems, databases and other IT systems that have vulnerabilities of their own. The technology used at this level can be easily obtained in comparison to the other levels. The central role that PLCs and RTUs play means that an attacker has direct access to HMIs and other connected systems; more crucially the limited authentication required to use Modbus protocol at this level allows attackers to send malicious commands to different systems. It is of no surprise then that between Feb 2013 and April 2016, 465 of the 801 vulnerabilities disclosed affected products at this level. (FireEye 2016)

**History of attacks**

There have been multiple recorded instances of attacks against SCADA systems. The attacks on SCADA systems can be distinctly categorised as Pre-Stuxnet, Stuxnet and Post-Stuxnet. This distinction is due to the steep increase in attacks on SCADA systems after Stuxnet in 2010, from 55 to 219 in 2011. This can be attributed to the public recognition that Stuxnet received, which attracted a whole host of attackers interested in making a statement.

a. **Pre-Stuxnet (1982 - 2010)**

There were multiple cyberattacks towards ICS during this period that received limited coverage due to various reasons. The earliest known attack was the Siberian Pipeline Explosion in 1982, where a trojan caused an explosion that resulted in part of the trans-Siberian pipeline vaporising. Similarly, a trojan was used by disgruntled employees to gain control of the central switchboard of a Gas company in Russia (Gazprom) in 1999, which allowed the attackers to control the gas flow in the pipelines. In the same year, a SCADA system failure was attributed to a cyberattack on the Gas pipeline at Bellingham that resulted in the leakage and ignition of 237,000 gallons of gasoline which caused 3 deaths and 8 injuries.

Between 1992-97, there were 3 attacks where attackers gained root privilege into the ICS and were able to alter credentials, disable alarms and modify the system to serve malicious needs at Chevron Emergency Alert System, Salt River Project Phoenix and MA Airport Worcester. (Ismail et al 2011)

On the shores of Australia in 2000, a disgruntled consultant executed a response injection attack on the sewage plant system at Maroochy Shire Council. The attacker was able to assume control of 150 sewage pumps and released 800,000 gallons of raw sewage into local rivers and parks; resulting in death of marine life and area residents being severely affected by the discharges. Being an insider, the attacker was able to use in depth knowledge of the system to alter electronic data in the system resulting in malfunction between Level 1 and 0 of the SCADA architecture. (Abrams & Weiss 2008)

b. **Stuxnet (2010)**

Stuxnet was the first publicly recognised attack on SCADA system. This attack was on a uranium-enrichment plant in Iran via the Siemens SCADA industrial software. The attackers infiltrated a third party provider for the plant and infected the Windows and USB based devices. In doing so, the attackers were able to attack the system despite the system not being connected to a public network.

The Siemens software was originally used to program the PLCs; which controlled the industrial systems. Stuxnet was able to collect surveillance data, place the system into a critical state and even falsely respond, to prevent alarms. This was done by overwriting the ladder logic and firmware on the PLC, allowing the attacker to force the PLC to report false responses. This attack resulted in Iran’s nuclear centrifuges sustaining severe damage that delayed their nuclear program.

This attack was only possible due to intimate knowledge of the plant itself; which was possible as an identical plan was located in Israel, additionally the Siemens Step 7 software was common place in most SCADA systems. (Broad et al 2011) The time and resources taken to develop an attack of this nature is often beyond the scope of non-state actors. (Turnipseed 2015)

c. **Post-Stuxnet (2012 - 2016)**

Shamoon was considered to be the most destructive attack since Stuxnet to a critical infrastructure. While it did not cause physical damage, interruption of service or stealing of data; the attack itself consisted of deletion of data within the system. The malware attack was initiated by a disgruntled insider who had complete access to the system. (Mackenzie 2012)

Malware is a popular tool used in attacks on SCADA systems. In 2014, a malware attack (Havex) was used to target ICS/SCADA systems. The attackers modified the legitimate downloads present on ICS manufacturer's website to upload and execute an additional file that housed the HAVEX malware. The malware once installed would then send information back to the master server as a form of intelligence gathering. (Constantin 2014) In 2015-16, a Russian based hacking group sent phishing emails that contained documents that once opened would infect vulnerable computers with malware infection; to attack utilities, Ukrainian firms, television channel and SCADA. (Polityuk 2015; Leylen 2016)

In 2016, hackers from a known hacktivism group were able to gain access to a water treatment plant at Kemuri and modify chemical values used to treat tap water. The attackers had used a vulnerability that existed in the payment system to gain access to the web server. As it happened the web server was hosted on a critical
system (AS400) that also connected the company’s IT network to the SCADA system. The command injection attack on this plant was possible due to the out-dated technology (10-year-old operating system), single point of failure (one system that connected many major systems), vulnerability in payment system and people issues (only one employee was responsible for managing AS400, INI file with administrative privileges). (Cimpanu 2016)

III. Related Work

Work in this area has been extensive due to the critical nature of the systems involved and their impact on social cohesion, economic prosperity and public safety in the event of an attack. In this section existing research into different laboratory SCADA environment and cyberattack techniques are investigated.

SCADA Physical Research Environments - SCADA Environment is a core necessity for research into cyberattacks. Due to the 24/7 nature of ICS research into live systems is not possible, therefore for research purposes physical SCADA research environments are built that can replicate the workings of a SCADA system. These systems house physical PLCs that are used in industry to make the simulation as similar to live systems as possible.

Idaho National Labs, Oak Ridge National Labs, Sandia National Labs and British Columbia Institute of Technology are established SCADA labs developed to combat physical and cyber threats to SCADA systems. (King 2010; Schwartz et al 2010; Boyer 2007) In an cross-instructional collaborative effort; University of South Australia (UniSA), Queensland University of Technology (QUT) and Mississippi State University have developed a SCADA environments to give local students access to equipment that help mitigate the costs of travelling overseas for similar training. (Sitnikova et al 2013)

SCADA Virtual Simulation Environments - Creating a SCADA environment has its limitations owing to its large size and complexity. Physical environments are not portability, which makes it infeasible to expose the system to researchers from different backgrounds. The cost is another limitation that restricts the number of organisations capable of building and maintaining them. A virtual environment that can simulate the function of a SCADA system would circumvent the portability and cost issues making the system accessible to more researchers than otherwise possible. Some such examples are; Emulab and Simulink (Genge et al 2012), SCADASIM (Mahoney & Gandhi 2011), SCAD/HMI (Adamo et al 2007), Matlab/Simulink ICS (Gao et al 2014) and Open Virtual Bed for ICS built in Python (Reaves & Morris 2012). These systems are still capable of using an industry standard physical PLC for a more realistic simulation.

Newly Developed Lab at UNSW@Canberra - Lab developed at UNSW@Canberra has been setup to simulate SCADA system. The system consists of 4 industrial level physical PLCs; inclusive of Allen-Bradley and Schneider. The PLCs are coded using ladder logic and use Modbus communication channels to control Train Station, Medical center facilities, Power supply, and Schneider. The PLCs simulate SCADA system. The system consists of 4

IV. Cyber Attack Techniques

Experiments on the system developed at UNSW@Canberra have been narrowed to focus on PLC related attacks. The techniques mentioned below will be limited to the water treatment plant control system; ranging from low level detection techniques for Reconnaissance to Communication protocol attacks (using MODBUS) and finally a low sophistication technique that has high impact Denial of Service (DoS) attack.

Reconnaissance – Reconnaissance is used to gain information about the control system and is often done prior to any destructive attack. (Thornton 2015)

Command injection attacks - These attacks inject false control commands into a SCADA system in order to either manipulate the controllers into completing incorrect control actions or to overwrite RTU programming to change control parameters. In either instance the intention of such an attack is to control the system via injection of malicious commands into the system. (Thornton 2015)

Response injection attacks - Response injection attacks change the expected response to alter the control decisions made by the system to suit an ulterior purpose. Many SCADA control system network standards do not have mechanisms to check for response integrity. (Gao et al 2010)

Denial of Service (DoS) - These attacks disrupt the communication link between the RTU or HMI, by overloading the network with traffic. The result of this attack is to break the feedback control loop and make process monitoring and control impossible. (Gao et al 2010)
V. Discovering Cyber Vulnerabilities in Water Treatment Plant

A. Aims and Motivation

PLCs are a core part of a SCADA control system, as they hold the programming and control parameters for the proper functioning of the system. As a result, they are the prime target for attackers as the literature review alluded to. The experiments in this report were chosen based on attacks on water treatment plants. Through these lab experiments the reader can impact each of the techniques have on the system.

B. Experimental Setup

Using four cyberattack techniques identified earlier in the report, the experiments have been broken into 4 labs and named accordingly. Scenarios were developed for attacks within each lab and their impact on the system was recorded.

The PLC for the water tower was developed by Schneider Electrics, the communication medium between the PLC and the RTU is done via Modbus ASCII (Serial) and Modbus RTU / TCP (Serial & Ethernet). As a result, the attacks in this report will be based on the vulnerabilities with Modbus communication.

C. Tools

The following tools were used on the attacker computer in order to complete the attacks on the system.

CAS Modbus Scanner - This tool formats inputs so that they can be transmitted via Modbus/TCP protocol to PLCs. It is capable of creating read and write commands to specific IP addresses, ports and memory addresses.

Wireshark - A network monitoring software that captures data. This tool is used to identify and read the data packets sent between the attacker computer and the PLC.

D. Experiments

1. Lab 1 - Reconnaissance

Aim: Reconnaissance is a low level detection technique used to gain system intelligence. Information such as the IP address of the PLC, ports used for Modbus and commands that can be executed; give attackers information that is used to build attack scenarios. With system specific information such as device manufacturer, model number, supported network protocols, system address, and a system memory map, attackers can build repository of attacks that work on specific systems which are then sold to other would-be attackers.

1.1 Attack 1: Address scan

Aim: An attacker can use an address scan to identify a MODBUS server IP address (PLC address) and it’s relevant port. MODBUS devices listen on port 502 by default but it can be changed based on system design requirements.

Procedure: As per Modbus protocol, servers are required to return a response code when addressed. Using this vulnerability, read commands were sent to IP addresses of the SCADA network and the responses are analysed. The attacker’s intention would be to try every iteration of the IP address range until a response is received. The response can either be an actual data stream from the PLC or an error message; indicating that a MODBUS device exists at that address.

Using the address scan on the water treatment plant the system produced three different outputs as detailed below.

1st Outcome: Read command to IP address with no MODBUS device

Read commands were sent to IP address and timeout errors were received, as seen in Figure 2. This meant that the IP addresses didn’t house PLCs.

2nd Outcome: Read command to IP address with MODBUS device

During the course of the address scan a “0x01” error message was returned by the system, as seen in Figure 3. This error message signals that the IP address houses a PLC or a device capable of responding to MODBUS/TCP.
3rd Outcome: Successful read from the water system

Using the IP address from the previous test, a read command to the system returned “01 77”, as seen in Figure 5. The returned message is in hex and translates to ‘375’ in decimal. ‘375’ as it turns out is the water level of the system at that point of time, as seen in Figure 6.

Address scan result: Based on the system responses for the read commands sent to the PLC, it was possible to identify the IP addresses used by the PLC; and in one instance the response revealed system relevant information (water level of the plant). The address scan also revealed that the system used the default port 502 for MODBUS communication. An attacker can use this information to proceed with the next stage of attack.

1.2 Attack 2: Function code scan

Aim: After identifying the IP address and port of PLC, an attacker would want to know what function codes are supported by the PLC at the known address. Function codes as the name suggests are functions that can be performed by the PLC (read/write). A PLC capable of read and write will have the ability to send and receive commands, making it more valuable to an attacker than a PLC that can only read.

Procedure: Different function codes are transmitted to the PLC IP address and their responses are analysed. The function codes have to be sent to specific memory addresses at the IP. Identifying the correct memory address an intensive time consuming task that is hard without insider information.

1st Outcome: Use a write a command on the “water level” address

A read command was sent to the PLC to identify the “water level” of the system at the memory address ‘30011’ as seen by the poll ‘01 04 00 0A 00 04’; where ‘01’ is the slave id; ’04’ refers to function code: Read Input registers (in the vicinity of 3xxxx); ’0A’ refers to the offset of ‘11’ giving the memory address ‘30011’ (hex translation of ‘0A’ is 10, but in Modbus due to the start address being ‘1’ instead of ‘0’ we get the address ‘30011’). The system returned ‘BF’ which corresponds to ‘191’, the water level at that point of time.

A write command was then send to the same memory address as seen by the poll ‘01 06 00 0A 03 E8’; where ‘01’ is the slave id; ’06’ refers to function code: Preset Single register (in the vicinity of 4xxxx); ’0A’ refers to the offset ‘11’ giving the memory address ‘40011’; finally, ’03 E8’ refers to the write value of ‘1000’. The coding in this PLC allowed the value to be written using a different function code into the read only address, as seen by the successful response in the screenshot above.

2nd Outcome: Write to an address housing a bool (0 or 1) variable

Two write commands were sent to a memory address, with mixed results. In the first instance ‘1’ is written to the memory address resulting in a successful write, as seen in figure 9.

Checking the variable in the system reflects the successful write command.

<table>
<thead>
<tr>
<th>Offset</th>
<th>Standard address</th>
<th>6 digit address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>227</td>
<td>00227</td>
<td>000227</td>
<td>0</td>
</tr>
<tr>
<td>228</td>
<td>00228</td>
<td>000228</td>
<td>0</td>
</tr>
<tr>
<td>229</td>
<td>00229</td>
<td>000229</td>
<td>1</td>
</tr>
</tbody>
</table>

In the second instance ‘36’ (translates to ‘24’ in hex) is written to the memory address, the response from the system identified the write command as successful. Upon checking the variable value, no change was discovered. Investigating this discrepancy showed that the memory address was a boolean address (accepts only ‘0’ and ‘1’). To an attacker identifying bool variables adds to the system intelligence being collected, as a bool
value is used in situations were certain conditions need to be met. In this scenario the bool variable was the ‘Hydro_BVOpen’ which is the Hydro butterfly value open request. Setting the variable ‘On’ and ‘Off’ would allow the attacker to step towards opening or closing the Hydro butterfly.

**Function scan result:** Using function scan it was possible to identify that the PLC accepts write commands which allowed the “water level” of the system to be written to; this information will aid in the later parts of the lab. Additionally, it was possible to identify variable types in the system which adds to the collection of system intelligence for future attacks.

1.3 Attack 3: Point scan

**Aim:** Programming points are used to store programming variables that link to sensors. Points are available for read and write access via network interfaces. A points scan is used to read the contents of implemented points to gain system intelligence.

**Procedure:** Sending a read command with a large data length results in the system returning more data than originally intended. Attackers can use this method to gain system intelligence in order to plan higher order of attacks.

**Outcome: Reading a large data length**

A read command to a known set point with a large data length results in the system responding with additional information for that memory address. In this scenario we sent a read command to the Hydro Load Reserve with a large data length. As a result the PLC returned data from the neighbouring address location to the attacker, as seen in the image below.

![Figure 8 Large data reveal (Modbus scanner)](image)

**Point scan result:** Using point scan it was possible to read data from registers following the memory address. This technique can be used on different memory addresses to reveal additional information about the system adding to system intelligence for future attacks. This attack formed the basis for the heartbleed bug that was prevalent with Open SSL, as documented by Codenomicon. The bug allowed for theft of protected information from secure web servers (email, instant messaging, VPNs) by asking for more information than required normally, resulting in a dump of system relevant information.

**Reconnaissance Lab Outcome:** Using the Address scan, Function code scan and Point Scan attacks on the system it was possible to identify the IP address and port of the PLC; the functions (read/write, variable linked to memory address) possible at different addresses and recover additional information from the system than intended.

2. Lab 2 – Command injection attack

**Aim:** Injection of malicious commands into a SCADA system interrupts process control and device communication. It modifies device configuration and can be used to change set points. The MODBUS protocol used in SCADA system lacks authentication to validate the originality of packets, this vulnerability is used to inject malicious data into the system to alter its state.

2.1 Attack: Altered Control Set Point

**Aim:** This attack is used to change the device set point. Set points are variables used to control the system. In the water treatment plan a reserve load set point allows the system to set aside a certain quantity of water as reserve. The reserve is used by the system to supplement the water levels if required.

**Procedure:** Identify a set point address and send a write command with a new value. Depending on the intention of the attacker this could be any value. Based on system design the change could trigger alarms. Identifying the correct memory address is the time consuming part of this attack as the attacker would have to tediously try each memory address and check the system for changes.
Outcome: Change Hydro Load Reserve Set point

The value for the Hydro Load Reserve Set point was identified as a writable memory location. This system uses this memory location as a set point to determine the amount of water that has to be kept in reserve in case of shortage.

A write command is sent to change the set point from 10 to 50 at the address location, the results in a change from ‘1.0%’ to ‘5.0%’ in the PLC. The Modbus scanner output as shown below shows the initial change to 50 denoted by “32” in hex and then back to 10 denoted by “0A” in hex.

The change to the system can be seen in the HMI output as seen in the Figure 11. The HMI is the primary tool used by the controller to monitor the system. The change in the reserve point did not raise any alarms and as such would go un-noticed by the controller.

Command injection attack outcome: Using the altered control set point it was possible to change a set point in the system without raising alarms. The IP address, port used by the system and the memory address of the set point were the only system information required to complete the attack. While getting some of the information is time consuming, no specific credential were needed as MODBUS does not require authentication.

3. Lab 3 – Response Injection attack

Aim: SCADA subsystems use polling to monitor the state of remote processes. Polling is the process sending a response to every request sent from a client to a server. As suggested by Gao, this technique lacks authentication and as such packet validation does not take place. Using this vulnerability, malicious response packets can be injected into a network to alter the state of the system.

3.1 Attack: Constant Sensor Measurement Injection

Aim: This attack is used to mask the real state of the system. Malicious packets are sent to the system resulting in the portrayal of an incorrect system state. A resultant outcome would be to make a controller take incorrect actions in order to control the system.

Procedure: Send malicious packets containing incorrect measurements over a period of time in order to mask the real state of the system. To a controller monitoring the system, it will appear as the system is going out of bounds and may result in the controller or automated control algorithm to take incorrect control actions.

Outcome: Incorrect water level

By sending a continuous stream of incorrect water level data to the HMI, it could convince controllers that the water pump is jammed on resulting in an increasing water level as opposed to normal operation of steady increase and decrease.

In reality the system is functioning as required, but the information on the HMI could prompt the controller to turn off the water pump to prevent the water tower from spilling.

In this scenario a script was written to continuous write a variable that increased every minute resulting in an output as seen the figure 12.
This attack modifies the communication between the PLC and the HMI. The system itself is unaffected, but a convincing attempt at this attack could make the controller turn off the pump, thereby meeting the attack objective.

Response Injection attack outcome: Using constant sensor measurement injection, it was possible to successfully hide the current state of the system. This attack can be used on any variable in the system in order to meet the needs of the attacker.

4. Lab 4 – Denial of Service (DoS)
Aim: To stop the proper function of the system. DoS attacks can be used to take the system or part of the system offline. SCADA systems are designed to lock out control actions (including system controllers) when faced with what the system perceived to be a DoS attack. These attacks have a low level of sophistication as a result they can be attempted with minimal system information, but the impact to the system can be very severe.

4.1 Attack: Distributed Denial of Service (DDoS)
Aim: Prevent the system from responding to controllers. Distributed DoS requires multiple systems sending requests to the system so as to overwhelm it, resulting in the system not responding to any request.
Procedure: Send a large amount of malicious data to overload the system which will eventually cause it to go offline.
Outcome: No Response

The result of the attack is a system that does not respond to the controller or the attackers. Checking system response from a controller system showed that the system stopped responding, as seen in figure 13. This attack has limited use to an attacker, as while the system is offline the attacker has no control over the system. The attack requires limited planning or research into the system and there are IT solutions that monitor and actively stop attacks such as these, but such solutions require implementation of IT policy on SCADA systems which isn’t always as straightforward as it seems.

Denial of Service outcome: Using DoS attacks it is possible to take a poorly planned system offline. The only information required by an attacker is an IP address. When used as part of an extended attack that targets other parts of the system, modifies set point variables and then locks out controllers its impact can be quite devastating.

E. Experimental Results and Conclusion

The techniques above showed the ways in which a system can be attacked using vulnerabilities known to exist in Modbus/TCP protocol. As this protocol is used in most PLCs today these attacks are possible in a majority of systems.

Using Reconnaissance it was possible to identify the IP address and ports of MODBUS devices; Command injection made it possible to alter system set point; Response injection allowed for masking the state of the system and Denial of Service showed the ease in which the system communication can be taken offline.

Gaining specific information about the system is a time consuming process, but cannot be relied to be a protection layer of the system. Stronger systems with better hardware and a more secure communication channel can implement similar IT policies used on computer networks which will allow for better security.

VI. Future Work

Using the basis of the attacks in this report, labs can be designed as part of courses to educate students on the cyber vulnerabilities in SCADA systems. More sophisticated cyberattacks designed to work on Level 0 and 1 of the SCADA architecture can be developed and tested. This continued process of attacking, recording results and educating will allow for better awareness in this field and can help inform engineers about IT relevant problems in the SCADA domain.

VII. Conclusions

Research in this project allows for a better understanding of threats facing SCADA systems today. Using the four cyberattack techniques (Reconnaissance, Command Injection, Response Injection and Denial of Service) it was possible to; gain system intelligence, identify and modify set points, alter communication between the
system sensors and HMI, and stop all communication with the system. The vulnerabilities present in the MODBUS communication channel present in most SCADA systems; procedures followed by engineers during system setup and general practice by controllers make way for a compromised system.

The increased connectivity of SCADA systems and legacy systems employed by critical infrastructure means that security of a SCADA system can no longer play second fiddle to reliability. Sophisticated attacks scenarios need to be developed using the existing methodology such that countermeasures can be developed and deployed to critical infrastructure. Lessons learnt from exploring vulnerabilities, creating attacks and developing countermeasures can be used to educate and enhance awareness of cyber-vulnerabilities and develop a cybersecurity strategy for SCADA systems.

References


Gao, W 2013, Cyberthreats, attacks and intrusion detection in supervisory control and data acquisition networks’, Mississippi State University.


