A Systems Approach to Cyber Security
A. Roychoudhury and Y. Liu (Eds.)

doi:10.3233/978-1-61499-744-3-70

Mind the Gap: Security Analysis of Metro Platform Screen Door System

Luying Zhou[1], Huaqun Guo, Dong Li, Jun Wen Wong and Jianying Zhou
*Institute for Infocomm Research, A*STAR, Singapore*

Abstract. A smooth operation of the Platform Screen Door (PSD) system is critical to the Metro system, and any disturbance to it may disrupt the train's normal operation. The paper presents the security analysis of a Metro Supervisory Control and Data Acquisition (SCADA) system, specifically the cyber security vulnerabilities in its PSD system. The PSD system includes control subsystem and signaling subsystem, and its operation can be controlled from the moving train and the station as well. The security features of communication protocols that are employed in the SCADA system and PSD system operation mechanisms are discussed in this paper. The weak security features render the PSD vulnerable to cyber attacks. Countermeasures, from both technical and human aspects, to protect the PSD system are studied. An experiment is conducted on a testbed of simulating Metro supervisory control system to test the system vulnerabilities. The results demonstrate that the PSD control system could be compromised by an attacker who gains physical access to the control network and launches forged message or replay message attacks. A firewall cyber security countermeasure is evaluated to show that it can prevent some of the attacks but has limitations due to its rule-based mechanism. Thus, it is necessary to mind the gap for the security of metro PSD system.

Keywords. SCADA, Platform Screen Door, Vulnerability, Cyber Security, Firewall

1. Introduction

Supervisory Control and Data Acquisition (SCADA) systems are Industrial Control Systems (ICS) which are widely used for monitoring and controlling geographically distributed operations, including power plants, manufacturing and processing industries, transportation (e.g., Metro railway) systems, and so on. SCADA systems interconnect remote physical processes, monitor and collect data from remote facilities and control the physical process. Availability and integrity are the most important aspects in the ICS. Real-time access to data assets is critical to control system operations. Unavailable or interrupted control operations result in the loss of facility functions. Manipulated data by unauthorized parties causes false actions. While unlike in the Information Technology (IT) system, confidentiality plays a less important role in the SCADA system [1].

Over the past two decades, SCADA systems have evolved from closed, proprietary systems to open networks comprising common platforms running common operating systems and standard TCP/IP stacks. Presently, the system is generally connected to the

[1] Corresponding author, Institute for Infocomm Research, A*STAR, 1 Fusionopolis Way, #21-01 Connexis, Singapore 138632; E-mail: Lzhou@i2r.a-star.edu.sg.

corporate intranet which may have connections to outside networks. The open architecture and increased connectivity provide more functionalities and reduce operation costs, but they significantly increase the exposure to cyber security threats. Those threats can exploit vulnerabilities uniquely in the design and implementation of SCADA and communication protocols to attack the system or launch attacks already known in the IT world but with a higher impact on the process managed by the SCADA system [2, 3]. Attacks on availability intend to deny access to system assets as well as operations. In SCADA system, Denial of Service (DoS) refers to deny of access to the components of a system, operator workstations, and communication channel as well. Attacks on integrity modify the content of a message or the content of system assets. In the SCADA system, it refers to modify acquired data or control commands transiting through the system. SCADA communication protocols, such as Modbus, and DNP3, lack authentication features to prove their origins and sequence of network traffic [4, 5], which lead to potential attacks of forged commands or false response messages injection into a SCADA system either through direct generation of such messages or replaying messages.

In this paper, we study a specific Metro SCADA system and examine its security vulnerabilities, in particular the vulnerabilities from cyber attacks against the Platform Screen Door (PSD) system operation. The PSD doors are located along the trackside at the station, and are designed for passenger safety and efficient train operation. Any cyber attack on the PSD system to disrupt or disable its operation will result in service unavailability and train delay, causing damage to operator reputation and financial loss as well. The countermeasures to such cyber attacks can be done from both technical and human aspects. Technical approaches such as developing secure communication protocols and secure firewalls could enhance the system capacity against cyber attacks, however human factors play an equal important role in providing system security.

The remainder of this paper is organized as follows. First, the Metro SCADA system is described in Section 2, which includes the station supervisory control system, communication system and signaling system. Modbus protocol employed in the SCADA network is also introduced. In Section 3, the operation of PSD system is specifically presented and its vulnerabilities are further discussed in details. In Section 4, the security countermeasures to protect the PSD system operation against cyber attacks are proposed and discussed. In Section 5, experiments of cyber attack exploiting the SCADA system vulnerabilities and the countermeasures of preventing the attack are demonstrated and discussed. Finally, Section 6 outlines our conclusions.

2. Background of Metro SCADA System

The Metro SCADA system is composed of an Integrated Supervisory Control System (ISCS) and a train signaling system. ISCS provides facilities for integrated, centralized and localized control and supervision of electrical and mechanical subsystems remotely located at the passenger stations, power substations, depots, and tunnels. Through Communication Backbone Network (CBN) the whole Metro system can be remotely monitored, communicated and controlled from the Operation Control Centre (OCC). The signaling system supports communications between trackside controllers and trainborne controllers, and controls trackside equipment, e.g., PSD doors, and necessary mechanisms for train position localization. The standard Modbus protocol is employed in the ISCS for the information transfer among the devices.

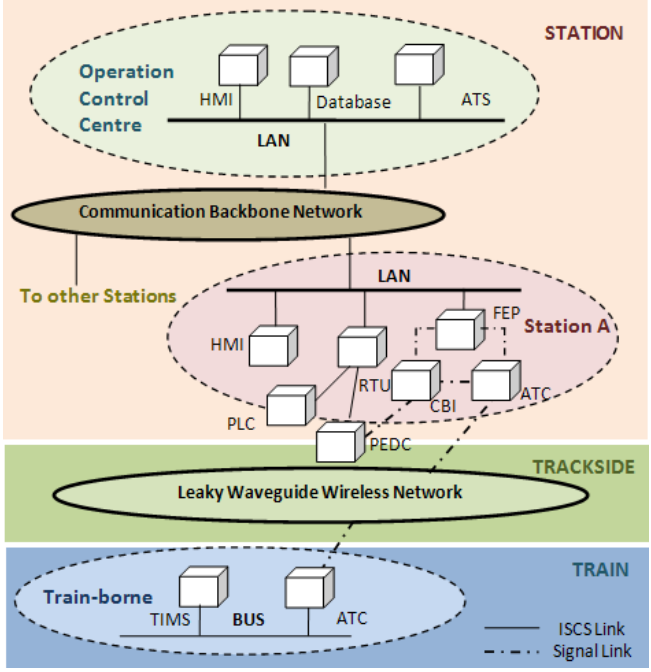

Figure 1. A typical Metro SCADA system

A typical Metro SCADA system is schematically drawn in Fig. 1. The abbreviations appear in this paper and corresponding definitions are summarized in Table 1 below.

2.1. Control and Signaling System

The main subsystems of the SCADA system, i.e., ISCS system and the signaling system, are briefly described as follows.

ISCS system

The ISCS system consists of Station Management Systems (SMS) and a Central Management System (CMS). SMS, located at each train station, handles all the monitoring and control activities within each station. It handles internal/external interface with field devices, and acquiring data from and sending commands to field devices. The field devices include controllers of PSD doors, elevators, escalators, station and tunnel lighting, and so on. Standard Modbus protocol is employed for data exchange between Remote Terminal Unit (RTU) and Programmable Logic Controller (PLC) where sensors and controllers are connected to. CMS is installed at the OCC, provides control and monitoring covering all the stations. Human Machine Interface (HMI) allows human operators to monitor the state of a process under control, and to manually override automatic control operations in the event of an emergency.

Table 1. Abbreviations and Definitions

Abbreviation	Definition	Abbreviation	Definition
ATC	Automatic Train Controller	ICS	Industrial Control System
ATS	Automatic Train Supervision	ISCS	Integrated Supervisory Control System
CAN	Controller Area Network	OCC	Operation Control Centre
CBI	Computer Based Interlocking	PEDC	Platform Edge Door Controller
CBN	Communication Backbone Network	PLC	Programmable Logic Controller
CMS	Central Management System	PSD	Platform Screen Door
DCU	Door Control Unit	RTU	Remote Terminal Unit
DoS	Denial of Service	SCADA	Supervisory Control and Data Acquisition
DPI	Deep Packet Inspection	SMS	Station Management System
FEP	Front End Processor	TDS	Training and Development System
HMI	Human Machine Interface	TIMS	Train Information Management System

Communication system

CMS and SMS subsystems have their own Local Area Networks (LAN) respectively, which may use dual 10/100 Mbps Fast Ethernet built on switches and employ TCP/IP standard protocols. OCC and all stations communicate with each other through CBN, which actually is a Wide Area Network (WAN). The communication between trainborne and trackside devices in signaling system can be done through leaky waveguide wireless channel, where microwave signals are transmitted in a trackside hollow rail.

Signaling system

The signaling system is responsible for the train communication and control. It includes a central Automatic Train Supervision (ATS) server located at OCC, trackside Automatic Train Controller (ATC) and Computer Based Interlocking (CBI) devices located at stations. The CBI device at each station also has a connection to the corresponding station LAN via a Front End Processor (FEP) and manages control signals on the trackside. Trackside ATC communicates with trainborne ATC through wireless channel, which operates in certain frequency band carrying control messages between the moving train and the station [6]. Trainborne Train Information Management system (TIMS) provides central management of control and monitoring of a range of devices in the moving train and issues instructions for emergency situation handling.

2.2. Modbus Protocol

In the Metro SCADA system, Modbus protocol is used for the communication between various field devices, e.g., RTUs and PLCs. Modbus protocol was developed for

industrial automation systems, and has become a common industrial standard to transfer digital or analog I/O information, diagnostic commands, system logs, and register values between industrial control and monitoring devices. Modbus protocol based devices establish the communication by using a master-slave technique in which only one device (the master) can initiate transactions. The other devices (slaves) will respond by taking the action requested in the query, or by supplying the requested data to the master.

Modbus protocol has two variants: Modbus/RTU and Modbus/TCP. Modbus protocol starndard defines the message structure and communication procedures [7, 8]. Modbus/RTU messages are transmitted between a master and slave devices over serial lines using the ASCII or RTU transmission modes. These messages usually have four main components: slave ID, function code, application data, and an error checking field. Comparing to Modbus/RTU, Modbus/TCP or Modbus over TCP protocol provides connectivity within an Ethernet LAN-based network as well as for IP-interconnected network, enabling a Modbus master to have multiple outstanding transactions, and a Modbus slave to engage in concurrent communications with multiple masters. Modbus/TCP utilizes TCP to carry the data of the Modbus message structure between compatible devices.

From the angle of cyber security, Modbus protocol lacks mechanisms for verifying the integrity of messages sent between a master and slaves. Modbus protocol does not authenticate the master and slaves. Moreover, the protocol does not incorporate any anti-repudiation or anti-replay mechanisms. The security limitations of Modbus can be exploited by attackers to compromise industrial control systems [5].

3. PSD Operation Mechanism and Security Vulnerability

The PSD serves as a safety barrier to prevent objects or passengers from falling onto, or gaining unauthorized access to the track, and in the case of underground track, protect passengers from strong tunnel draughts accompanying an approaching train. PSD usually has a strong frameless structure that aesthetically blends with the architectural design of the stations.

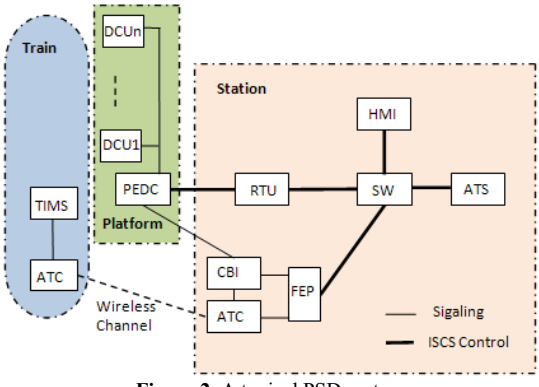

Figure 2. A typical PSD system

The PSD system is designed to provide an efficient and safe door operation, and vital circuits are hardwired and based on safety logic with fail-safe principle. This means that

any single fault will not put passengers in danger. The PSD system includes two subsystems: **signaling** and **control** systems. The PSD signaling system consists of an ATC located in the train, an ATC and CBI at the station, and the Platform Edge Door Controller (PEDC) at platform, as shown in Fig. 2. The communication between ATCs is taken over wireless channel and proprietary protocols, and the channel between CBI and PEDC is hardwired serial line. The PSD control system, being a part of the ISCS, consists of in HMI, ATS, RTU and also the PEDC, and uses standard communication protocols, such as Modbus/RTU and RS485.

3.1. Platform Screen Door Operation

The train controls the open and close of the platform doors via the PSD system when it arrives and leaves the platform, in addition to simultaneously controlling the opening and closing of train doors through an internal train control system.

The vital commands, e.g., open and close the PSD doors, are communicated through the CBI in the signaling system. Other commands, such as PSD doors state setting, are transmitted in the control system. In CBI, the interlocking is an arrangement of signals and signal appliances which are interconnected such that their movements must succeed each other in proper sequence, and an action would not be performed without predefined conditions satisfied. For example, when a train is arriving at a station, the command of opening PSD doors will be issued only after the train has fully stopped and also stopped at the pre-defined position. Trainborne ATC issues open/close PSD door commands through wireless channel to ground station ATC, and through wired line to CBI for interlock checking and then to the trackside PSD controller - PEDC. PEDC executes these commands to open or close the doors via hardwired lines to the Door Control Units (DCUs). The train's operation is also affected by doors' states fed back from DCUs to PEDC and to CBI. The doors' open/close states are detected or monitored by sensors, and these can be mechanic touch sensors, and the sensed door state information is processed in CBI. Only all doors are closed or opened will the CBI acknowledge the state of the accomplished command to the train [9, 10, 11].

PSD doors can also be controlled/monitored through PSD control system, i.e., through ISCS. The main functions performed by the PSD control system are to collect platform door state information, and enable/disable the door operation. ATS in the ISCS can issue PSD open/close commands, but the commands have to pass through the signaling system via FEP and to CBI. The data collecting and state setting commands are transmitted in ISCS among HMI, the RTU and the PEDC. E.g., when being informed that certain train doors are inoperative, the ISCS sends state setting commands to PEDC via RTU to disable the corresponding PSD doors. PEDC links with DCUs via Controller Area Network (CAN) bus to communicate control system commands received by the PEDC from the ISCS to the DCUs, and status reports from DCUs to the PEDC and en route to the ISCS.

3.2. PSD System Security Vulnerabilities

The availability and reliability of the PSD system are directly relevant to the train's smooth operation. Any disturbance to the operation of the PSD system will disrupt the train's schedules, e.g., failure to open the PSD doors will block passengers from exiting or entering the train. Although such disruption may not jeopardize the safety of

passengers due to the fail-safe mechanisms enforced in the system, it affects train service availability, results in revenue loss, and damages operator's reputation.

The normal PSD operation could be disrupted by cyber attacks, such as forged message injection, and message replay attacks. The PSD signaling system employs closed and proprietary communication protocols and interfaces, which are secretly kept and is thus harder for the cyber attacker to penetrate to the system. However, it is different for the ISCS system, where openly accessible standard protocols and interfaces are employed. With known protocols and message format, and compounded with weak security protections, the ISCS is more vulnerable to cyber attacks. In the following sub-sessions, the PSD control system's security vulnerabilities are analyzed.

Forged Message Attack

In the Modbus data units there is no command sequence number. The master device associates the first response received with its command, and responses which are not associated with a command will be discarded by the master. There is no digital signature or authentication mechanism in the Modbus data units to allow the master to confirm the response is from the addressed slave. An attacker could insert a response to a command sent to another slave before the addressed slave can respond. This vulnerability allows an attacker to eavesdrop on Modbus communications and forge a response to a command before the addressed slave responds. For example, a forged response to a state collecting command would report false PSD doors state information which causes the operator to make wrong assessment and take wrong action.

Message Replay Attack

Due to the lack of message authentication and integrity checking, the message receiver would fail to identify the replaying messages. The control commands or responses to/from the PSD system could be captured over the transmission process and later simply replayed by an attacker. The replayed control command could set wrong PSD door states, and the replayed response messages could lead the operator to make wrong control decision. Besides the PSD system, other electrical and mechanical subsystems in the Metro system are controlled and monitored by operators via the communication networks in ISCS, and they are vulnerable to the message reply attacks due to the weak security measures in the ISCS system.

Human Factors

State-of-the-art technology cannot secure the Metro system alone, minimizing human error is even more crucial. Errors from operators —violations of standard procedures, wrong configured settings, lack of vigilant and prompt investigation of suspicious communications —open the door to successful attacks. For example, using a personal USB thumb drive on the ISCS equipment may introduce malware, virus into the system that compromises the PSD system, and not strictly following the procedure when operating the system or handling any anomaly could let the attacker bypass the security wall as well. Human factors play an important role in securing the Metro system, since the system is still vulnerable to human error even with strong security measures enforced.

4. Countermeasures to PSD System Attacks

The PSD system vulnerabilities could be exploited by adversary to disrupt the system operation. We study the available security approaches to counter the attack, and to secure the PSD system.

4.1. Secure Modbus protocol

To counter the message modification and replay attacks due to the lack of authentication controls in the protocol, i.e., to identify and prevent forged messages from an attacker, one approach is to develop and apply a secure industrial protocol, e.g., secure Modbus protocols [12, 13]. The secure protocol implements an authentication mechanism that provides a verifiable and secure way of indentifying the source of all data transmission. The authentication feature can be enhanced by employing symmetric cryptography and hashed message authentication codes. In the protocol both sender and receiver share a common secrete key from which session key is generated. The approach is an ideal one as it solves the problems at their origin, but to implement such a secure industrial communication protocol faces challenges because of the introduced significant changes to the long lifecycle legacy SCADA system and configuration.

4.2. SCADA Firewall

An approach that effectively addresses the forged message injection attack issues while without significant changes to the legacy system is to deploy SCADA firewalls. A SCADA firewall can be an independent device that locates between the sender and receiver and examines the packets passing through it. It filters the packets based on pre-defined rules, e.g., on IP address, protocols, port numbers and Modbus function codes. It could filter out the packets with irregular formats or unspecified instruction [14, 15]. However a pure firewall could not block the replaying packets and forged packets which have normal formats and contents, and comply with the pre-defined rules, as it does not perform message authentication and message sequence checking.

4.3. Security Gateway

The PSD system does not demand very stringent reaction time to the door opening/closing action. As the default time to execute the opening/closing commands are in a few seconds [9], an extra time in the milliseconds for processing security schemes would be acceptable.

The approach of deploying a security gateway could enforce the security measures such as message authentication and integrity. The security gateway, located before the field devices, e.g., before RTU or PLC as they generally interface with multiple field devices, can be an independent device that has the computation power and storage capacity to perform key management, authentication and message integrity functions in the place of RTU or PLC. A secure channel could be setup between the HMI and the gateway. There is no change required to RTU or PLC, but one of drawbacks of this approach is the extra processing time introduced in implementing the enhanced security features, which may not be feasible for time critical control process. As pointed out above, the PSD system can tolerate milliseconds delay, thus the gateway approach can provide a secure communication channel from the server and HMI up to the front of RTU or PLC.

4.4. Human Factor

Despite all the countermeasures enforced in the technique domain, an error contributed to human factors could break up the defense line, resulting in the PSD operation disruption. The importance of technology, procedures and people must be emphasized in equal order. The countermeasures to prevent human errors should involve the awareness education of the human vulnerabilities, commitment to operational principles, compliance to operation standards, and clear communication of responsibility and accountability.

5. Experiment Study

Experiments are conducted to demonstrate the cyber attacks to the ISCS system and evaluate the feasible countermeasures to prevent such attacks. We present the experiments in which an attacker exploits security vulnerabilities of the protocol employed in the ISCS system and launches message replay and forged message attacks, and these attacks could compromise the PSD system and other systems related to station equipment operation.

The experiments are conducted on a Training and Development System (TDS) -- a testbed of the ISCS system of a Metro system, which is built to test new software and system, and train new staff. The experiments could not be conducted on the real Metro system, for fear of disturbing its operation. The hardware and software configuration and operation of TDS are similar to those of the ISCS system in a train station. The hardware and software components include HMI, RTU, Ethernet switch, HMI software and Modbus protocol. The TDS is a small scale implementation of the ISCS system, so the experiments of cyber attacks and countermeasures on the TDS are equally applicable to the real operating Metro system, without facing the risk of interfering with train revenue-producing operation. There are some cases of SCADA testbed, but not Metro SCADA testbed, can be used by us as reference [16]. Our experiments assume that attacker can gain physical access to the ISCS network, e.g., access a switch by an insider or a visitor in the control or station room. The testbed and experiment setup is shown in Fig. 3.

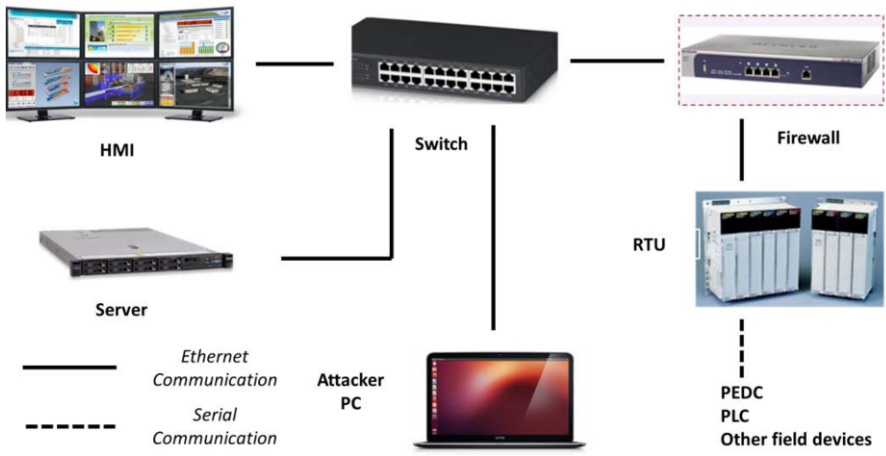

Figure 3. Testbed and experiment setup

The testbed consists of a HMI, a server, and a RTU connected by an Ethernet switch. An attacker's laptop PC is connected to the switch. A firewall device is later inserted in the testbed between the switch and RTU to counter the cyber attacks. The HMI is a software application and an important part of ISCS system that presents the real-time information to an operator about the states of multiple processes, and to implement various control instructions. For example, as shown in Figure 1, an operator sits in a train station can use the HMI to remotely monitor the real-time conditions of field devices. The operator can remotely retrieve RTU status information, read register values of certain PLC, control station lighting systems, open or close station power systems, disable one or multiple platform screen doors, and many more. Typically, the above information is displayed in a graphic format or Graphic User Interface (GUI). Meanwhile, the operator can also use pre-defined buttons and hyperlinks in the HMI to implement commands to control the above systems.

The RTU provides intelligence in the field, and allows the operator or central SCADA server to communicate with the field instruments. Its function is to control process equipment at the remote site, and acquire data from the equipment. A RTU may be interfaced to multiple servers and field devices with different communication media. In the typical Metro implementation, RTU connects PEDC and multiple PLCs with RS485 serial interface, and the commands to field devices will go through RTU and be relayed to the field devices. If forged messages could be accepted by RTU, then they would be accepted as well by the connected field devices, as over the RS485 line and in the field devices there is no cyber security mechanism to examine whether a seemingly normal message is an authentic one.

Various cyber-attack techniques can be applied to attack a SCADA system, such as Distributed Denial of Service (DDoS) attack, Man in-the-middle attack, forged message attack, virus and Trojans, and social engineering. In the testbed environment, to test the attack on a certain subsystem without affecting others, forged message attack or packet replay attack technique is selected. The experiments demonstrate that control commands to the field devices could be captured, modified or replayed by an attacker, and be injected into the ISCS, and be accepted and processed by the RTU. In the following, due to the project confidentiality requirements, some descriptions of raw data and specific protocols employed in the SCADA system are not disclosed.

5.1. Replay/Forged Message Attack

The objective of replay attack is to control the field devices in SCADA system by sending "legal" commands. However, this "legal" only indicates the packet frame, because the attacker can generate the command packets by using his/her knowledge of the industrial protocol. Both the original command and forged command can be accepted and executed by SCADA device, such as RTU and PLC. The difference is that the forged command is sent to the RTU at a wrong time. For example, the attacker sends a forged command to shut down the station lighting during peak hours.

To successfully send a forged single packet that can be accepted by RTU and the field device, the process can be divided into the following stages:

- Packets sniffing
- Packets analyzing
- Packets impersonating

Packets sniffing

We assume that an attacker can find a way to physically connect his/her attacking tools into the SCADA network. This assumption is necessary if the SCADA network is a closed network without connecting itself to public Internet. One of the sniffing methods has been described below.

The attacker connects a device, such as a Linux Ubuntu OS laptop, to the switch located in the middle of communication channel between the HMI and RTU, and sniffs the control commands destined to field devices. We use a Cisco 2960 switch, which is a popular module and employed in many real systems. For example, the attacker can connect his/her laptop into an empty port of station switch, and configure that port as a mirror port. Hence, all the network traffic between HMI and RTU will be copied into the attacker's laptop. This step is easy for insider attack. To capture and view the packets on the laptop, the attacker can install Wireshark and make sure that the network card is running on promiscuous mode. As the network traffic is not encrypted, all network traffic is in plain text and readable by the attacker's sniffer. The attacker eavesdrops and records the network traffic and then further analyses the control command for its corresponding functions and contents.

If the switch administrator sets a password to log in the switch configuration mode, the attacker may face certain challenges when he/she tries to configure the mirror port. The attacker can use other hacking techniques such as Address Resolution Protocol (ARP) spoofing. Another easy option is to unplug the RTU Ethernet cable from the switch, and plug attacker's laptop in between. In such case, the laptop works in network bridge and sniffer mode. In reality, the attacker also needs to pay attention on the security mechanisms employed in the SCADA system, such as Intrusion Dedection System, which may alert the administrator any irregular operations. However many legacy SCADA systems do not employ good protection mechanism as they were built in more than 20 years ago.

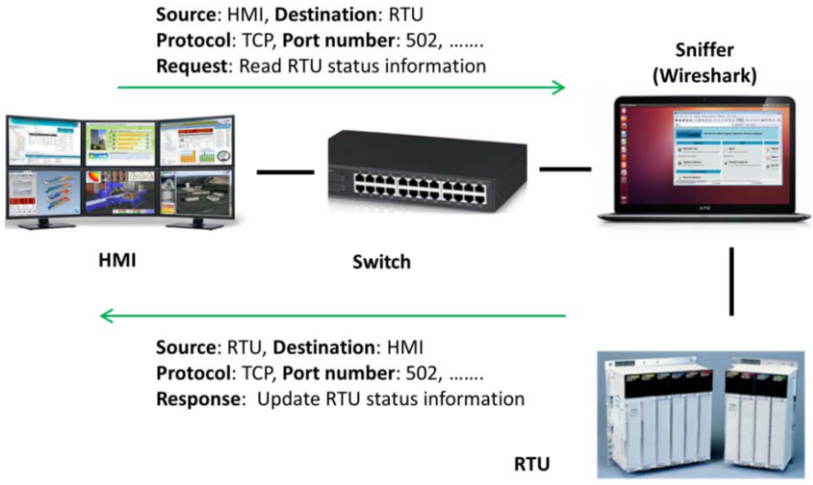

Figure 4. Packets sniffing

Packets analyzing

The next step is to perform a deep analysis of the captured packets. The attacker tries to find the packets that carrying important SCADA control instructions. Only after gathering sufficient knowledge about the network traffic and the control command functions, the attacker can further forge those control instructions according to their targeting objectives. As described above, the field devices are connected to RTU through serial communication. A control command is always transmitted over TCP/IP to the RTU first. Modbus/TCP (TCP port 502) is the most common industrial protocol to transmit such commands. Then the RTU plays as a protocol converter to re-package the packet payload based on serial communication standard, and sends the command to its destination. To recognize the control instruction related packets, the attacker can use the following characteristics to match:

- Protocol -- Typically, most SCADA commands will be transmitted over Ethernet by TCP.
- Port number -- Typically, each control command will be assigned a unique and static TCP port number. Some commands will use standard Modbus/TCP, which is TCP port 502.
- Packet timing -- Based on traffic volume, more than 90 percent of packets are periodic, because they are used to keep the SCADA system alive. The control commands may appear non-periodic.
- Other packet details -- For a TCP/IP packet, a lot of information can be used to identify itself, such as MAC/IP addresses, TCP flags, SEQ/ACK numbers, and payload length.

Once control instruction related packets are identified, the attacker needs to analyze its payload frame. This is also a critical but time-consuming reverse-engineering step. The complexity and possibility of payload frame may vary depending on the controlled functions and scale of on-site devices. If the SCADA system utilizes standard industrial protocol such as Modbus, most of them will be exactly same or very similar to standard Modbus payload frame. Statistics is the most direct and efficient methodology to analyze and summarize the payload frame so far. Hence, a successful payload analysis needs enough sniffed packets for each operation.

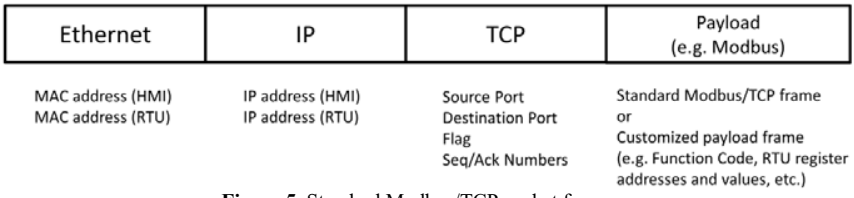

Figure 5. Standard Modbus/TCP packet frame

Packet impersonating

Let assume the attacker has already identify a single packet that carrying the instruction to disable the PSD system, then he/she can forge it and re-send it to RTU at any time he/she wants. If the packet were sent during peak hour, the passengers could be blocked from entering or leaving the train.

As shown in Fig. 5, the attacker can forge the data part based on original packets. Some examples are shown below. All of them may disrupt the normal operation of PSD or other systems behind RTU.

- Change the function code of Modbus
- Read the register values of RTU
- Write new/invalid values into RTU registers
- Remote restart PSD devices
- Clear device memory, reset counters in memory

There are many available packet generators can be used to simulate legal SCADA traffic as well as generate forged packets for attacking purpose, such as

- Colasoft Packet Builder (Windows OS) [17], an example shown in Fig.6.
- PackEth (Linux Ubuntu OS) [18]

The forged packets can be sent out from any laptop that is connected to the switch (Fig. 3). The attacker could further modify the packet payload or the packet header such as IP address and MAC address of legitimate sender, and fabricate and send a false command to RTU. The forged command could be accepted by RTU without distinguishing whether it is an authentic original command or a replay command. The RTU could execute a forged command that requests state information without realizing the ongoing attack and report its state information in a response packet.

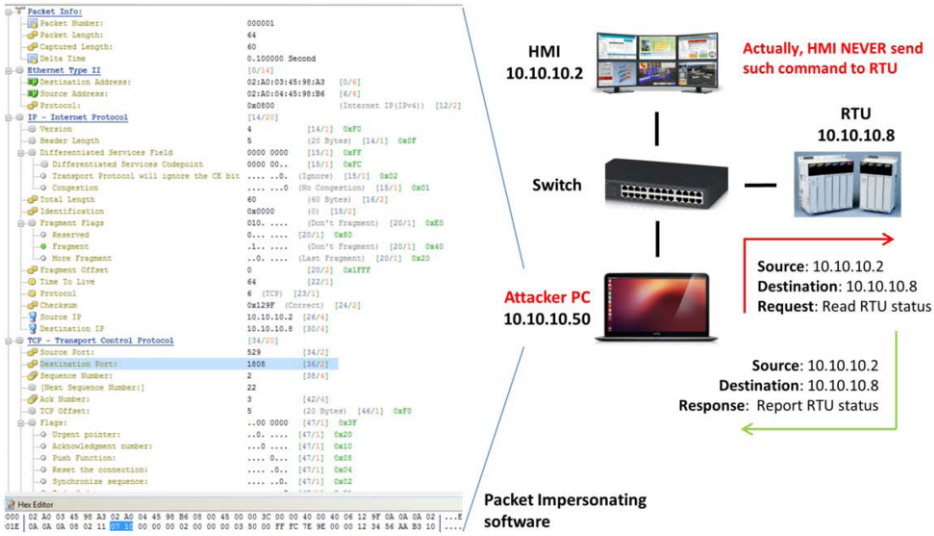

Figure 6. Customized packets generator (Colasoft) in Windows OS

5.2. Firewall Countermeasure

To protect the SCADA system against cyber attack, a firewall approach is evaluated. A firewall is a network security device that monitors and controls network traffic passing through it based on predefined security rules, and it incurs less interference to the legacy system than other approaches, such as upgrading communication protocols and installing security gateways. In the experiment, a specific firewall device is chosen [14], which has features such as plug-n-protect installation, no requirement on network configuration and

routing table's change, and no disruption to the existing control system. It can perform deep packet inspection (DPI), which inspects deeper into the application content fields of a packet, specifically to common industrial protocols, to examine command message structure or Modbus function code and determine whether the intended action makes sense. The DPI functions of the commercial firewall product do not work for non-standard proprietary protocols, which unfortunately is the case of the system under study. The firewall is placed between the switch and the RTU to filter traffic comes into or from the RTU, illustrated in Fig.3. In the experiment, the firewall can successfully block the forged packets if the attacker uses an IP address and a port number that are not existed in the pre-defined firewall rules. In a SCADA firewall, all existed legal network devices must be recorded in its rule list. Hence, the threats from unknown devices can be reduced but not eliminated. In Fig. 7, Rule 1 and Rule 2 are very basic examples to put legal SCADA field devices into whitelist. On the other hand, each SCADA command is always assigned a static TCP port number. Those port numbers will be in standby status and wait for incoming commands. All other unrelated ports are suggested to be put into blacklist of firewall. Hence, the attacker cannot invade through those ports. Most SCADA commercial firewall products are able to achieve above functions. But it does not mean perfect protection at all.

If the attacker can generate forged packets with IP addresses and port numbers which are allowed by the firewall rules, then the firewall may failed to block such packets. As described in Section 5.1, Fig. 6 and Fig. 7, a 3-step forged packet method was applied during the experiment to send attacking packets and pass the firewall successfully. Besides IP addresses and port numbers, the attacker can also calculate the correct Seq/Ack number, checksum and flags. This is relatively easy work for experienced attacker.

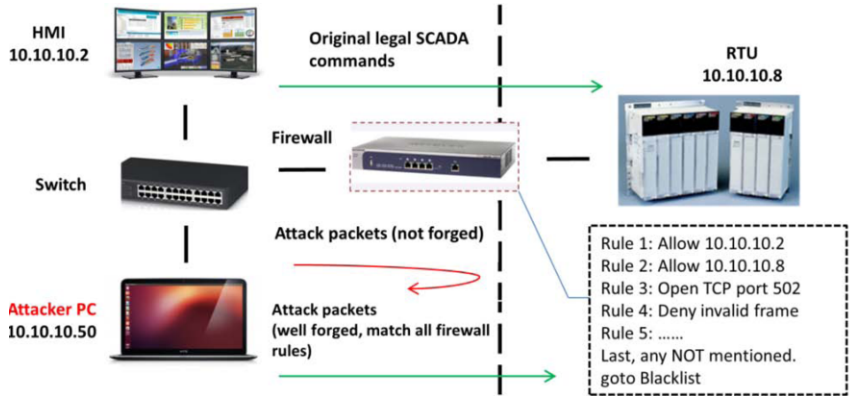

Figure 7. Forged/replay packets can evade firewall detection

The firewall could filter out the forged packets carrying undefined IP address, port number, protocols, or with abnormal packet format, but it failed to prevent the replay/fabricated message attacks as long as the packets comply with the predefined firewall rules. Most SCADA firewall products will work on single packet inspection by checking whether their packet frame match all the pre-defined rules.

The experiments show that once an attacker is able to access the ISCS network, he could launch a series attacks on the Metro system, besides on the PSD system. It is due to the weak security measures of the Metro system, as there are no message authentication

and integrity mechanisms enforced in the communication protocols. Applying firewall can block the attack packets which are not allowed by the pre-defined policy rules, but the approach has its limitations. However, a SCADA firewall can be more efficient and advanced when it performs the following functions:

- Restrict unknown devices to connect to SCADA network -- If the SCADA network uses static IP addresses instead of Dynamic Host Configuration Protocol (DHCP), it will be harder for an attacker to connect his/her attacking tools to the network.
- Only allow connections on known ports, restrict traffic from new ports -- Ports look like the gates of a network. Unnecessary gates should always be kept locked. Then the firewall can only inspect the traffic pass the known ports.
- Able to do Deep Packet Inspection (DPI) [19] on non-standard industrial protocol -- However, available commercial SCADA firewall products, e.g., [14, 15] are designed to inspect the payload of standard industrial protocols.

The firewall technology alone could not completely protect the SCADA system from cyber attack. A consolidated security approach will be more effective which offers tightly integrated multiple detection mechanisms including rule based packet filtering and sender's authenticity verification.

6. Conclusion

We described a specific PSD system in detail, and analyzed its security vulnerabilities. The attacks, in the form of forged message and message replay, could exploit the vulnerabilities of weak security communication protocols in the ISCS system and disrupt the PSD system and other equipment operation. Countermeasures were studied that include employing secure communication protocols but faces challenges as it requires significant changes to the legacy system. The secure gateway approach, requiring less change to the legacy system, could perform the security functions otherwise to be done in RTU or PLC, and apply a wide range of security technologies to protect the system. Another approach, easier to be adopted, is to deploy firewall but it has limitations. The firewall approach could filter out packets based on pre-defined rules, but fail to block packets which comply with the rules. Due to the concern of disrupting the revenue generation Metro system operation, the experiments were not performed on the working PSD system, instead on a testbed of train station SCADA system. The experiment successfully demonstrated the forged and replay packet attacks to the system devices and the effectiveness of firewall protection in the SCADA system. This work points out that we should mind not only the platform gap, but the cyber security gap in the PSD system as well.

Acknowledgements

This work was supported by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014NCR-NCR001-31) and administered by the National Cybersecurity R&D Directorate. The special thanks are also given to SMRT Corporation Ltd to provide the TDS for the experiments and provide domain knowledge and technical support.

References

[1] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, Guide to Industrial Control Systems (ICS) Security, NIST Special Publication 800-82 Revision 2, May 2015, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf

[2] C. Ten, et al, Vulnerability Assessment of Cyber Security for SCADA Systems Using Attack Tree, Power Engineering Society General Meeting, 2007. IEEE, 1-8.

[3] M. Ozturk, and P. Aubin, SCADA Security: Challenges and Solutions, June 2011, http://www2.schneider-electric.com/documents/support/cybersecurity/WP-SCADASecurity-schneider-electric.pdf

[4] A. Shahzad, "Cryptography and authentication placement to provide secure channel for SCADA communication", International Journal of Security, Vol. 6 (3), 2012, 28- 44.

[5] W. Gao, T. Morris, B. Reaves, and D. Richey, On SCADA Control System Command and Response Injection and Intrusion Detection, eCrime Researchers Summit (eCrime), 2010, 1-9.

[6] A. Bertout, and E. Bernard, Next Generation of Railway and Metros Wireless Communication Systems, Aspect-IRSE, 2012, 1-8.

[7] Simply Modbus, http://www.simplymodbus.ca/index.html

[8] Modbus Application Protocol Specification, V1.1b3, http://www.modbus.org

[9] Fersil, Copp Platfrom Screen Door Control System, http://www.fersil-railway.com/en/systems-software/platform-screen-doors-control-psdc/copp-system/

[10] NRT, Platform Screen Door System, http://www.platform-screen-doors.com/platform-screen-doors/overview

[11] ST Electronics, Platform Screen Door System, http://www.stee.stengg.com/pdf/railway_systems/Taipei_PSD-Eng.pdf

[12] I. Fovino, A. Carcano, M. Masera, and A. Trombetta, Design and Implementation of a Secure Modbus Protocol, Third Annual IFIP WG 11.10 International Conference on Critical Infrastructure Protection, Hanover, New Hampshire, USA, March 2009, 83-96.

[13] G. Hayes, and K. El-Khatib, Securing Modbus Transactions Using Hash-Based Message Authentication Codes and Stream Transmission Control Protocol, Third International Conference on Communication and Information Technology, Beirut, 2013, 179-184.

[14] Tofino, https://www.tofinosecurity.com/products/Tofino-Firewall-LSM

[15] Checkpoint, https://www.checkpoint.com/products-solutions/next-generation-firewalls/

[16] S. Jung, J. Song, and S. Kim, Design on SCADA Test-bed and Security Device, International Journal of Multimedia and Ubiquitous Engineering, Vol. 3, No. 4, October, 2008

[17] Colasoft Packet Builder, http://www.colasoft.com/packet_builder/

[18] PackEth Linux Software, http://packeth.sourceforge.net/packeth/Home.html

[19] H. Schulze, Deep packet inspection tutorial, https://ipoque.com/sites/default/files/mediafiles/documents/iss2012/ISS-Tutorial-DPI1.pdf