International Journal of Engineering & Technology, 7 (2.14) (2018) 153-159

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

Simulating command injection attacks on IEC 60870-5-104 protocol in SCADA system

Qais Saif Qassim [1] *, Norziana Jamil1 [2], Maslina Daud [3], Norhamadi Ja'affar [3], Salman Yussof [2], Roslan Ismail1 [2], Wan Azlan Wan Kamarulzaman [4]

[1] Institute of Informatics and Computing in Energy, Universiti Tenaga Nasional, Malaysia
[2] College of Computer Science and Information Technology, Universiti Tenaga Nasional, Malaysia
[3] Cybersecurity Malaysia, Malaysia
[4] Tenaga Nasional Berhad, Malaysia
**Corresponding author E-mail: qaissaif@uniten.edu.my*

Abstract

IEC 60870-5-104 is an international standard used for tele-control in electrical engineering and power system applications. It is one of the major principal protocols in SCADA system. Major industrial control vendors use this protocol for monitoring and managing power utility devices. One of the most common attacks which has a catastrophic impact on industrial control systems is the control command injection attack. It happens when an attacker injects false control commands into a control system. This paper presents the IEC 60870-5-104 vulnera-bilities from the perspective of command and information data injection. From the SCADA testbed that we setup, we showed that a success-ful control command injection attack can be implemented by exploiting the vulnerabilities identified earlier.

Keywords: *SCADA; IEC 60870-5-104; Cyber-Security; Vulnerability.*

1. Introduction

At the core of electric power system is SCADA, supervisory control and data acquisition. A SCADA systems are typically incorporate sensors, actuators and control software that are deployed in widely dispersed locations. Sensor's readings are being transmitted from remote locations to a central control room where the information is analyzed, decisions are made and actions are reverted back to the intended remote location [1]. SCADA systems have historically been isolated from other computing resources [2], [3]. However, the use of TCP/IP as a carrier protocol and the trend to interconnect SCADA systems with enterprise networks introduce serious security threats [4]. Most SCADA protocols were designed without any security mechanisms. Therefore, an attack on the TCP/IP carrier can severely expose the unprotected SCADA protocol [5]. Furthermore, attacks on an interconnected corporate network could tunnel into a SCADA network and wreak havoc on the industrial process [6]. Thus, it is of vital importance that any vulnerabilities of SCADA system must be identified and mitigated.

The main focus of this work is to investigate the security weaknesses of one of the most commonly used communication protocol in electrical power SCADA system, IEC 60870-5-104. This protocol was intentionally chosen in this study because it is crucial for the communication between the control stations and distribution stations in many electrical power facilities around the world. Moreover, as a proof-of-concept, a simple control command injection attack is implemented and executed in a simulated SCADA environment. The remainder of this paper is organized as follows. Section 2 presents a brief overview of the IEC 60870-5-104 protocol. Moreover, it discusses the vulnerabilities of the examined protocol. Section 3 describes the environment in which the control command injection attack is to be executed. The execution procedure and results of the command injection attack is presented in section 4. The conclusion of this work is given in section 5.

2. IEC 60870-5-104 protocol

IEC 60870-5-104 (also known as IEC104) is one of the IEC 60870 set of international standards released by the International Electrotechnical Commission (IEC). The IEC 60870 standard defines systems used for tele-control in electrical engineering and power system automation applications [7]. It specifies a communication profile for sending basic tele-control messages between two systems over standard TCP/IP network, which allows simultaneous data transmission between several devices and services.

The security of IEC 104 has been proven to be problematic [5, 8]. According to recent security advisories [7], multiple security weaknesses associated with this protocol have been reported [3] such as the lack of strong data encryption and proper authentication could allow an unauthenticated, remote attacker to inject, hijack or spoof network communications or exploit input validation flaws on vulnerable systems using the IEC 104 protocol [7].

Although the IEC have published a security standard, IEC 62351. The security standard is intended to implement a security measures to protect the IEC tele-control protocol series. The standard suggested methods for data transfer authentication and end-to-end encryption which would prevent many potential attacks such as replay, man-in-the-middle and packet injection [9]. Unfortunately, due to the increase of implementation complexity vendors are reluctant to roll this out on their networks [3, 10]. Security vulnerabilities of the IEC 60870-5-104 communication protocol can be summarized as follow [10-11]:

- Absence of checksum field in IEC 60870-5-104 protocol, as it is completely dependent on lower carrier communication layers to protect the data integrity.
- Lack of inbuilt security mechanisms for providing data protection at the application layer and data link layer such as encryption and authentication.
- The IEC 104 protocol can only transmit 255 bytes at any one time. This indirectly limits the number of security bits that can be added during data transmission. Furthermore, the communication medium used may or may not have security mechanism implemented. The use of communication medium such as twisted pair copper cable or radio waves (both are common in SCADA implementation) makes it easy for data transmitted to be eavesdropped or fake data to be injected into the medium.

3. Simulation environment

The vulnerabilities of the IEC-60870-5-104 protocol have been investigated and exploited using our laboratory-scale SCADA system. Our SCADA testbed emulates a generic electrical power SCADA system. The experimental setup is as shown in Figure 1. It includes several SCADA key components: real-time digital simulator to emulate the power system, generate data and receive commands, master and local HMI, an RTU and several engineering workstations for testing and analysis purposes. In this testbed, an Opal-RT model OP5600, a specialized hardware/software simulator, is used to simulate IEEE New England 39-Bus power system in real time. At the bay level, the functions of both the controller and RTU are modelled using the Real Time Application Platform (RTAP) which is a proprietary platform used for modelling and simulating industrial control system devices. The state of the emulated power system and RTU can be monitored and controlled through master (station level) and local HMI (bay level), which are modelled using Station Level Operator Interface (SLOI). SLOI is a proprietary platform used to visualize the power system which is simulated by the Opal-RT simulator and to receive and send commands from RTU and Controller (RTAP). In this work, the SCADA network is designed to bridge the substation (bay level) and the control center (station level) through a network switch. Therefore, all the SCADA testbed components are considered in the same network. As well as the components are connected to a local area network through an Ethernet switch.

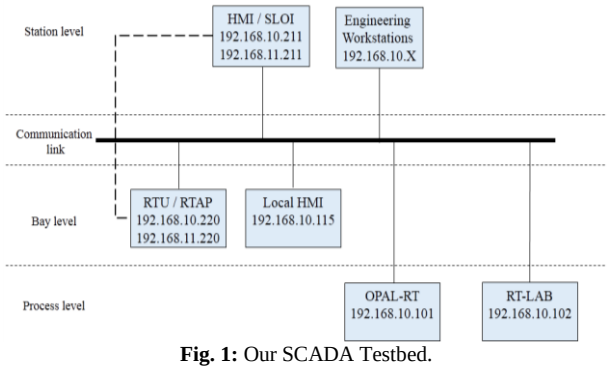

Fig. 1: Our SCADA Testbed.

4. Control command injection attack

This type of attack have been considered to demonstrate the impact of lack authentication and encryption method in IEC104 protocol [2]. Packet injection is a form of computer network attack that refers to the process of interfering with an established network connection, by means of constructing and injecting packets of malicious intent into a network in such a way that the crafted packet appears as if they are part of the normal communication stream [2].

Packet injection is commonly used in man-in-the-middle and denial of service attacks. Command injection attacks inject false control commands into the controlled system such as opening or closing a circuit breaker. Command injection can be broadly classified into two categories.

- Attacks against human operated systems: In this category, control devices are monitored and manually controlled by human operator. The attacker presume that the human operators oversee control systems and occasionally intercede with supervisory control actions. Consequently, they may attempt to inject false supervisory control actions into a control system network.
- Attacks against automatic controlled systems: The second category contend with remote terminals and intelligent electronic devices monitor and control the physical process directly at a remote site. Such devices are generally programmed to carry out the monitoring and controlling processes automatically and systematically. This programming takes the form of ladder logic, C code, with registers which hold key control parameters such as high and low limits gating process control actions. Hackers can use command injection attacks to overwrite RTU programming and remote terminal register settings.

Command injection attacks can result in catastrophic consequences. For example, an adversary can alter control set point to make devices operate in critical levels. Moreover, he/she can alter alarm values stored in PLC registers to disable alarms by changing alarm set points levels to values in-line with the altered high and low set points. An attacker can also forge reading values on the control center to trick system operators and let the attack go unnoticed.

4.1. Attack goal and prerequisites

As mentioned earlier many industrial control system protocols including IEC104 lack authentication mechanism to validate the origin of packets as well as the absence of encryption mechanism. This enables attackers to capture and alter command, response and/or measurement packets. Additionally, attackers can craft malicious command messages and directly inject them into the control system network.

Generally, the potential impact of such attacks include interruption of SCADA control devices, interruption of device communications, unauthorized modification of device configurations and unauthorized modification of process set points. To successfully inject a control command into a controlled system, an attacker should be able to craft a valid IEC104 packets. In order to exploit the target system effectively, the following conditions should be met:

- The SCADA subsystems transmit commands and information in plaintext;
- The target system must leverage unprotected communication channel.
- No security measures have been implemented such as data encryption, firewall, deep packet inspection or intrusion detection systems.
- The channel on which the target communicates must be vulnerable to interception.

In order to carry out a successful injection attack, the TCP session between the controller and control system must be interrupted and the attacker should engage with persistence TCP connection and initiate a temporary TCP session with the controlled system. The attacker then should be able to inject arbitrary commands (such as open or close a circuit breaker) to take over the control of the target industrial device.

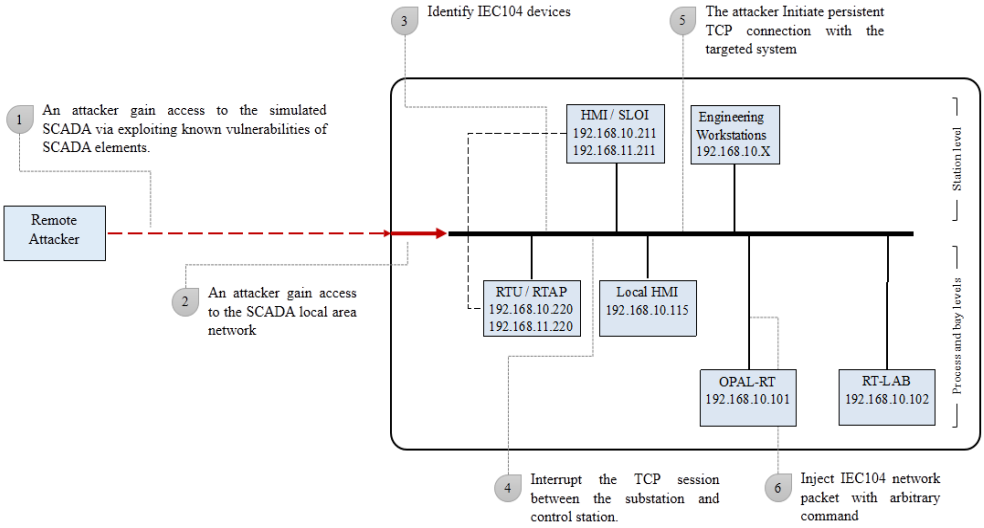

Fig. 2: Command Injection Attack Scenario.

4.2. Attack description

Figure 2 demonstrates the command injection attack scenario and highlights the main activities required to perform such attack. In general, to carry out a command injection attack, the following phases should be realized:

4.2.1. Phase 1: penetrate the SCADA system

The first step an attacker has to perform in order to carry out an attack on a system is to gain access to any of the resources of the target system. A successful penetration operation typically follows four steps:
Step 1 – Reconnaissance (Footprinting)
In order to carry out a successful attack, an attacker must first gather as much information as possible on the specified target systems. During this phase the attacker carefully analyze the victim's system, identify its strengths and weaknesses, pinpoint responsiveness to the unexpected and collect all the information required to determine and develop the attack. An attacker uses a variety of sources to learn as much as possible about the targeted system and how it operates, including Internet searches, social engineering, dumpster diving, domain name management/search services as well as non-intrusive network scanning. The activities in this phase are not easy to defend against. Information about a system, organization or individual finds its way to the Internet via various routes.
Step 2 - Scanning
Once the attacker has enough information to understand how the system works and what information of value might be available, the attacker begins the process of scanning perimeter and internal network devices looking for vulnerabilities and weaknesses they can later exploit for the targeted attack.
Step 3 – Enumeration
Enumeration is defined as a process which establishes an active connection to the target system to discover potential attack vectors. It is considered as a critical phase in system penetration process as the outcome of enumeration can be used directly for exploiting the system. In this context, for this work the information collected during the reconnaissance phase will be utilized to collect more details information about the target that needs to be compromised. Information such as, usernames, machine names, share paths, route tables, service settings and DNS details as well as applications and banners are collected in this phase.
Step 4 - Gaining Access
The final phase is the attack itself. Using wide range of tools and various techniques identified in the enumeration phase, remote attackers exploit the vulnerability to break into the target system. Gaining access to resources is the whole point of the penetration attack whereas the usual goal is to use the network as a launch site for attacks against other targets. Therefore, the attacker at first must gain some level of access to one or more network devices.

4.2.2. Phase 2: impose control on the SCADA network

The main goal of this phase is to take control over the core of the network that connects the substations to the control station to capture the entire network traffic by the attacker's machine for further actions. Moreover, in order to launch an injection attack, the attacker must be on and have access to the same network of the target. This is accomplished though Man-In-The-Middle (MITM), ARP spoofing, ARP Cache poisoning, MAC flooding or physical attacks such as port stealing [12]. In this work, as the experiments are performed in a controlled environment the data is captured using SPAN port (mirror port). SPAN port is commonly used on a network switch to send a copy of all packets on the network to a specific port.
In real life scenario, an attacker may gain access to a machine which is connected to a SPAN port or may trick the switch into becoming a span port or may gain administrative control of the switch. Consequently, the attacker will be able to capture all the traffic passing though, including data being transmitted by the identified attack targets. Another way to capture the network traffic is by getting in between the two (or more) targets and capture data. This can be done by performing ARP spoofing attack. An attacker can send spoofed ARP messages, which associate the attacking machine's MAC address with the target's IP address. So, every packet which is addressed to the targets will arrive at the attacking machine. Accordingly, the attacker can view and edit all packets being sent between the targets. Number of alternative ways to the ARP spoofing can be used to capture network traffic such as Domain Name System (DNS) poisoning and Content Addressable Memory (CAM) table overflow attack, which overflows the memory of the switch and turns it into a basic network repeater.

4.2.3. Phase 3: discover IEC104 devices

The main goal of this step is to enumerate the operational technology (OT) system's network to discover IEC104 devices attached to it. This can be done in one of two ways; passive or active. Passive enumeration is where an interface is set into promiscuous mode, which it will accept all packets on the wire. The packets can be monitored for IEC 104 traffic with tools such as Wireshark and tcpdump. On the other hand, active detection is where packets are sent out from the machine to try to invoke a response from an IEC104 device. Active detection can be accomplished by sending a test Application Protocol Data Unit (APDU) and waiting for confirmation message from the targeted systems. Once the attack-

er is able to passively capture and analyze IEC104 packets through SPAN port or other means, the connected IEC104 devices can now be identified using network packet sniffing tools such as Wireshark. One of the main features of Wireshark is that it enables packet filtering. One can filter the captured traffic to only display specific protocol or host. Utilizing this feature to display only IEC104 packets is shown in Figure 3. This makes packet analysis to find useful information much easier. Using the filtered traffic, the TCP conversation associated with IEC104 protocol is as shown in Figure 4. This confirms that the machine that is associated with 192.168.10.101 IP address has the role of RTU as it uses port TCP/2404, which is defined as IEC 60870-5-104 process control over IP based on the Internet Assigned Numbers Authority (IANA) description. On the other hand, the machine of 192.168.10.211 acting as the control station or similar functionality as it continuously receives IEC 104 data from 192.168.10.101 machine.

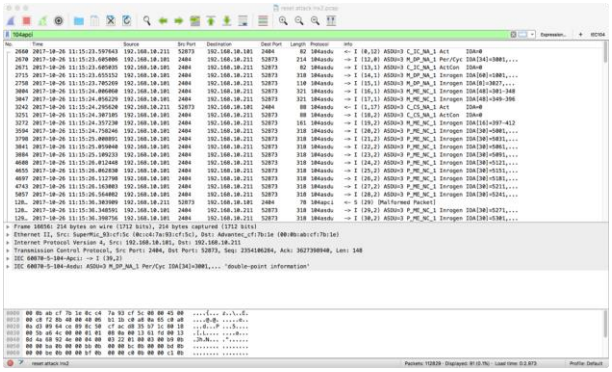

Fig. 3: IEC104 Packets Captured and Filtered Using Wireshark.

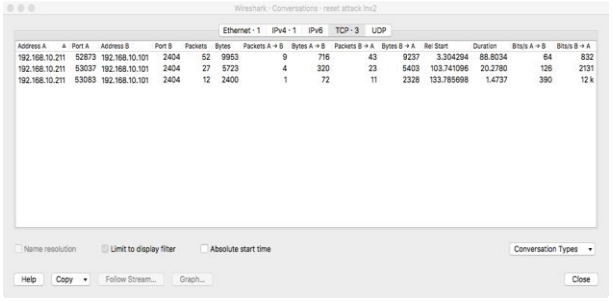

Fig. 4: Network Conversation between IEC104 Devices.

Figure 5 illustrates an IEC104 message in monitor direction generated by substation to update the master station with the current state of system's devices. Each monitor point in the controlled system is identified by its information object address (IOA), which is unique for each common ASDU address in the network. From the illustrated packet, an attacker is able to identify the address of each monitor point (e.g. 3001, 3002, 3003 ... 3034) to be used in order to craft a valid packet with bogus device state. At this stage, an attacker needs to identify the range of the monitor points configured in the system which is 34 monitor points addressed as 3001 to 3034. Figure 6 illustrates an IEC104 message in control direction generated by master station to instruct the control substation to connect (close) the associated control point (e.g. circuit breaker) addressed by IOA1001, the control action is specified by the double command output (DCO) field. Based on the concluded range of the monitor points, an attacker is able to determine the address range of control points which is started from 1001 to 1034. This is information is remarkably helpful to maliciously control the system by injecting vicious control packets.

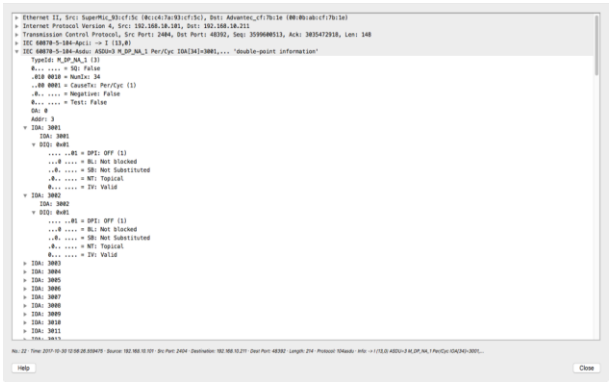

Fig. 5: IEC104 Monitor Packet Dissection.

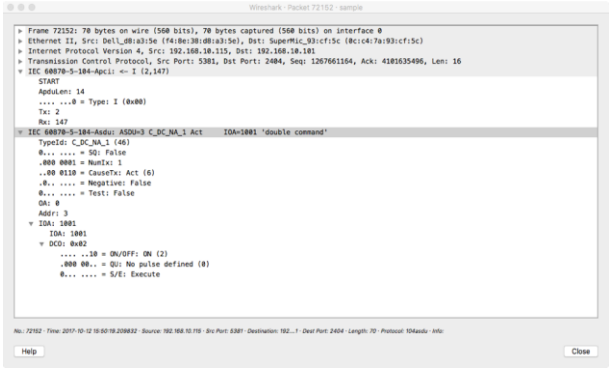

Fig. 6: IEC104 Control Packet Dissection.

4.2.4. Phase 4: reset an existing TCP session

IEC104 communication protocol is designed specially to run over TCP/IP. TCP connections (sessions) are used for all IEC 104 communications, whereas a TCP 3-way handshake process is set up and a connection is established before any device on the network transmits or receive data. Although multiple devices can be connected to the master station at the same time, only one active connection can be used for data traffic (controlled via the "STARTDT Act" and "STARTDT Con" telegrams). Therefore, for the attacker to successfully inject IEC104 command the existing TCP connection to the target system should be interrupted.
In order to carry out a successful injection attack, the TCP session between the controller and control system must be interrupted and the attacker should engage with persistence TCP connection and initiate a temporary TCP session with the controlled system. The attacker then should be able to inject arbitrary commands (such as open or close a circuit breaker). At this stage, an in house developed tool called tcpTerm written in C language was used to interrupt the existing TCP session. The syntax and usage are illustrated in Figure 7, while Figure 8 demonstrate an example of how to execute the TCP session rest attack. Figure 9 illustrates an example of TCP session reset using the developed tool. Although the TCP reset message is generated and injected by the attacker machine which has IP address 192.168.10.198, the message is spoofed in a way that it is appeared to be generated by the control station which has IP address 192.168.10.211 (Master HMI) using the same port number with proper TCP sequence number. Figure 10 demonstrates the flow graph between the two systems.

4.2.5. Phase 5: initiate a persistent TCP connection

IEC 104 telegrams are carried over TCP protocol that is connection oriented service that requires a valid connection (three-way handshake) and sequence numbers, which is chosen pseudo-randomly upon connection initiation. Without valid sequence numbers, a blind injection of packets into the network will be rejected by the recipient station. Therefore, a more sophisticated injection tools to be considered which manage TCP connection conversation and permitting the attacker to specify the malicious payload.

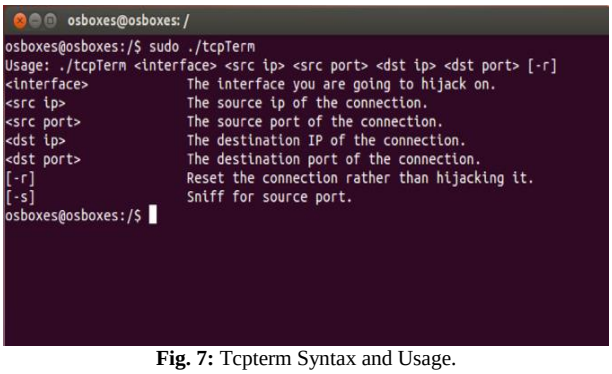

Fig. 7: Tcpterm Syntax and Usage.

Fig. 8: Tcpterm TCP Reset Attack Execution.

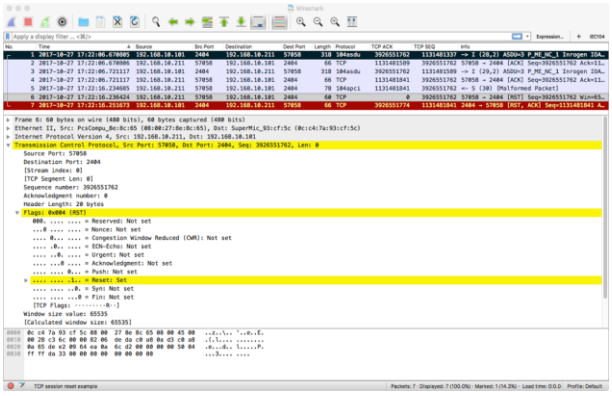

Fig. 9: TCP Session Reset Example.

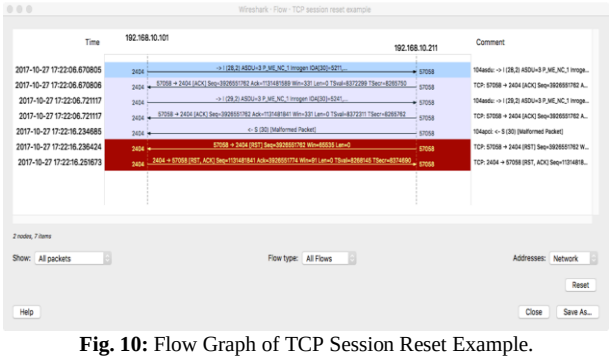

Fig. 10: Flow Graph of TCP Session Reset Example.

Once the TCP connection between the controller and controlled station is interrupted, an attacker initiates a temporary persistent TCP connection with the controlled station to inject the spurious command and control messages to the industrial control system which may have an adverse effect such as causing serious damage or impeding production. Packet Sender, an open source utility to allow sending and receiving TCP and UDP packets, has been used to inject the bogus IEC104 packets and manage the TCP traffic. It supports persistent TCP connections via a separate user interface dialog, which is enabled by checkbox on the main window or from the settings dialog as illustrated in Figure 11. Packet Sender efficiently manages the SYN and ACK values and it initiates three-way handshake before the packet can be injected to prevent detection by security systems and avoid packet rejection by the kernel's TCP/IP stack of the recipient station. Wireshark was used in this stage to monitor the traffic and ensure the correctness of the crafted packet.

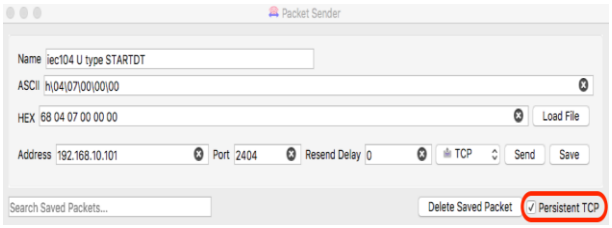

Fig. 11: Packet Sender Interface.

Figure 12 demonstrates flow graph of the forged TCP session establishment and the communication of the data between the attacker machine and controlled system. The attacker machine takes over the master HMI and starts to receive sensors readings and other related sensitive information. At this stage, an attacker is able to inject/execute any command of his/her choice on the target controlled system.

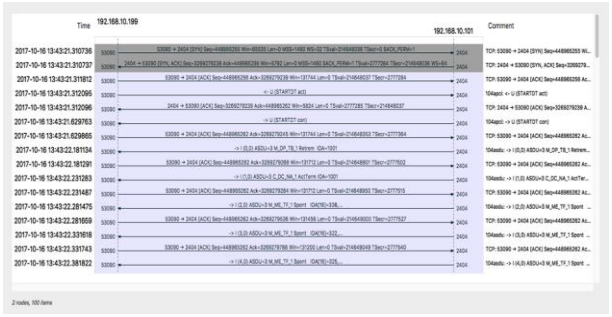

Fig. 12: Flow Graph of the New TCP Connection.

4.2.6. Phase 6: packet injection

At this stage the potential target has been identified, the attacker has gained a remote access to the network and a TCP session between the attacker's machine and the target has been established. Consequently, the attack is ready to be launched. This can be accomplished through two steps:

Step 1: Craft a valid message

Once potential targets have been identified, it is possible then to craft an IEC104 message. There are various tools available to initiate, construct and inject network packets into a network, for this work Packet Sender have been used. In general, to construct a valid IEC104 packet, the following activities should be realized.

- Create raw socket: A raw socket is an internet socket that allows direct sending and receiving of Internet Protocol packets without any protocol-specific transport layer formatting. Raw socket facilitates the construction of network packets.
- Create a buffer for the packet: A buffer is the placeholder in which all information like the Ethernet header, IP header, TCP header and data will be put together. The buffer represents the constructed packet.
- Create the Ethernet header: Once the raw socket and the buffer are ready, the Ethernet header can be constructed. The MAC addresses are required in this step.
- Create the IP/TCP headers: In this step, IP and TCP headers are constructed. The destination port number are set to 2404, which is the standard TCP port allocated for IEC60870-5-104 protocol.
- Create the payload: In this work, the payload is IEC 60870-5104 telegram which is referred to as application protocol control information (APDU). The APDU requires to define the type of telegram, the cause of transmission, common address and information object address (IOA). For example, a command to control a specific circuit breaker the type of

telegram should have the value 46 (Double Command), the cause of transmission is 6 (activation), the common address represents the station address which is 3 in our SCADA testbed. Lastly, the IOA is the specific address of the targeted circuit breaker.

- Assemble the packet: Final step is to use the buffer to bind the constructed headers together with the payload. The packet now is ready to be injected into the network.

Step 2: Deploy into the wire

As soon as the required packet is ready, the packet is injected into the network to be routed to its intended destination. For a successful attack implementation, the attacker has to establish a proper TCP connection and maintain a network conversation in order to enable the two systems exchange data. At first, the attacker must send "STARTDT act" message and wait for the "STARTDT Con" reply from the target system. Afterwards, the attacker may able to send control commands, receive sensor reading and interrogate for device state. Wireshark was used in this stage to monitor the traffic and ensure the correctness of the crafted packet. As soon as a TCP session has been established, an attacker injects a control message trying to initiate a communication with the controlled system. Figure 13 shows the proper setting for injecting (STARTDT Act) control command while the dissection of the sent packet is illustrated in Figure 14. As a response to that message, the controlled system should return (STARTDT Con) which is an acknowledgment signal indicating a confirmation for communication channel has been successfully initiated. Afterwards, the attacker may able to send control commands, receive sensor reading and interrogate for device state.

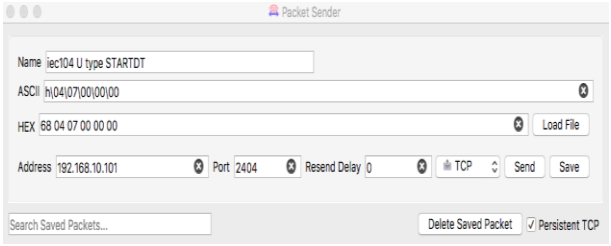

Fig. 13: Settings of Injecting STARTDT Command.

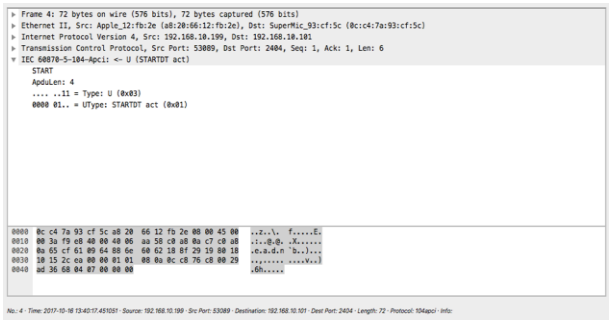

Fig. 14: STARTDT Command.

The main intention of this type of attack is to inject a control command of malicious intent to be executed by the controlled system. For this work a non-spoofed control command have been injected from the attacker's machine to be executed by OPAL-RT to change the state of a particular circuit breaker. Figure 15 shows the appropriate setting for control command injection. The HEX field represent the data to be placed as packet payload. The dissection of the injected packet is demonstrated in Figure 16. As shown, the target control point is set to 1001 with "double command" control command which instruct the target device to change its state according to data given in DCO field.

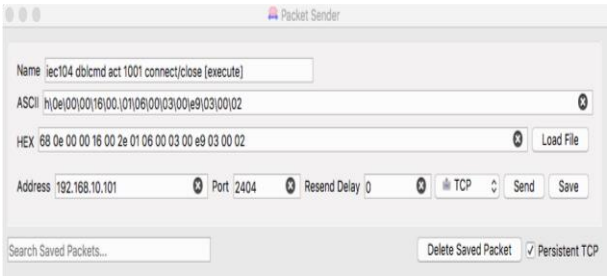

Fig. 15: Packet Sender Settings of Injecting Control Command.

It is worth to mention that IEC command is usually generates multiple communication response messages as illustrated in Figure 17. For example, if a master station requires to commit a modification on one of the controlled systems' components, the master station must send a master control message of "control command" type addressed to the specified component. The message should has cause of transmission (COT) field value equal to 6 (Act), which represents command activation state. The opposite side should reply the message with COT field value equal to 7 (ActCon) which denotes a confirmation for the activation request as illustrated in Figure 18.

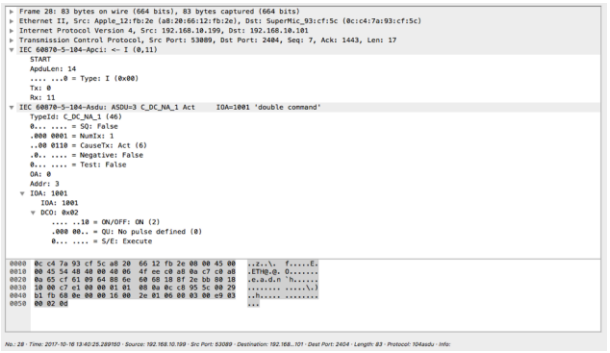

Fig. 16: IEC104 Control Message for Activation Request.

Fig. 17: Flow Graph of IEC104 Command Messages.

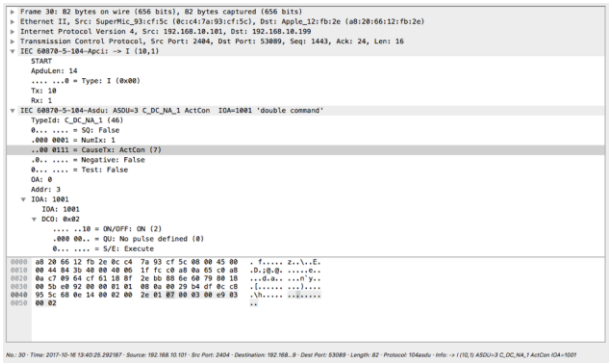

Fig. 18: IEC104 Control Message for Activation Confirmation.

Afterwards, the control action is executed on the controlled system (remote substation) which in turn will update the control station with two message; the first with COT equal to 11 (Retrem), is sent to return information caused by a remote command as shown in Figure 19. While the second message, is sent to terminate the current activated request. Request termination is achieved by sending a message addressed to the control command originator with COT

equal to 10 (ActTerm). Unlike the rest of command communication messages which are IEC104 control command, Retrem message is a point information response that retrieves the state of a specific component of a system. This can be utilized in Man-In-The-Middle related attacks by injecting false responses or intercepting this particular packet. The impact of this attack is monitored via Matlab's scope viewer. Figure 20 demonstrates the impact of modifying the state of a circuit breaker addressed by IOA 1001. As illustrated, the voltage level at bus 1 was set to 258.25V and after about 6 minutes of normal operation the command injection was executed resulting in disruption of the voltage level at bus 1 and all other related buses. The disruption continues until the power system reaches a balanced steady state.

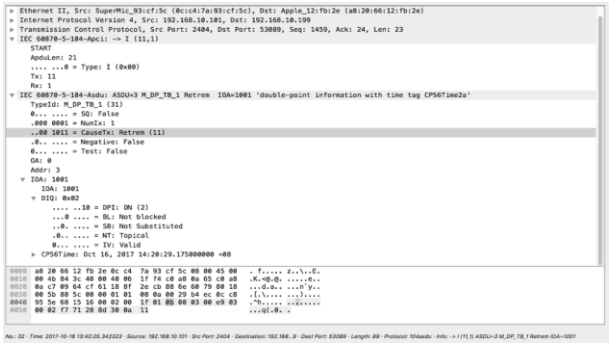

Fig. 19: IEC104 Control Message for Return Information Caused by A Remote Command.

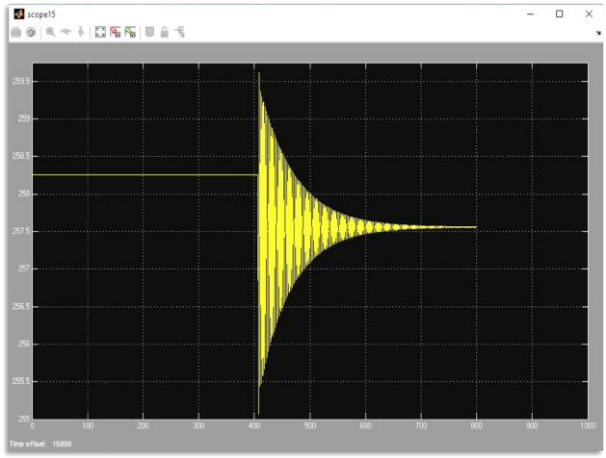

Fig. 20: Voltage Oscillation at Bus 1 after Turn-On Circuit Breaker Addressed by IOA 1001.

5. Conclusion

The lack of security protection and attack prevention mechanisms at the application and data link layers of the IEC 104 protocol implementation has indirectly resulted the critical infrastructure systems face an extreme risk of cyber-attacks. In this work, we explained the impact of the control command injection attack conducted in our SCADA simulation testbed. The experiment carried out in this work showed that IEC-60870-5-104 communication protocol used in SCADA system is insecure and can be exploited for more attacks that can disrupt normal operation of critical infrastructures. This needs tremendous effort to study on how to strengthen the security level of SCADA should the legacy communication protocol such as IEC 60870-5-104 is used.

Acknowledgement

This research study is supported by Ministry of Science, Technology and Innovation Malaysia (MOSTI) through DSTIN project 2016-2018 and Tenaga Nasional Berhad through TNB Seed Fund 2016.

References

[1] Mehta BR & Reddy YJ (2015), Chapter 7-SCADA systems. In B. R. Mehta & Y. J. Reddy (Eds.), *Industrial Process Automation Systems*. Oxford: Butterworth-Heinemann, pp. 237–300.

[2] Gao W, Morris T, Reaves B & Richey D (2010), On SCADA control system command and response injection and intrusion detection. *Proceedings of the IEEE eCrime Researchers Summit*, pp. 1–9.

[3] Nazir S, Patel S & Patel D (2017), Assessing and augmenting SCADA cyber security: A survey of techniques. *Computers and Security* 70, 436–454.

[4] Hong J, Stefanov A, Liu CC & Govindarasu M (2012), Cyber-physical security in a substation. *Proceedings of the IEEE Power and Energy Society General Meeting*, pp. 1–11.

[5] Dondossola G, Garrone F & Szanto J (2011), Cyber risk assessment of power control systems-A metrics weighed by attack experiments. *Proceedings of the IEEE Power and Energy Society General Meeting*, pp. 1–9.

[6] Igure VM, Laughter SA & Williams RD (2006), Security issues in SCADA networks. *Computers and Security* 25, 498–506.

[7] Hodo E, Grebeniuk S, Ruotsalainen H & Tavolato P (2017), Anomaly detection for simulated IEC-60870-5-104 traffic. *Proceedings of the ACM 12th International Conference on Availability, Reliability and Security*, pp. 1–7.

[8] Tawde R, Nivangune A & Sankhe M (2015), Cyber security in smart grid SCADA automation systems. *Proceedings of the IEEE International Conference on Innovations in Information, Embedded and Communication Systems*, pp. 1–5.

[9] Sun CC, Hahn A & Liu CC (2018), Cyber security of a power grid: State-of-the-art. *International Journal of Electrical Power and Energy Systems* 99, 45–56.

[10] Pidikiti DS, Kalluri R, Kumar RS & Bindhumadhava BS (2013), SCADA communication protocols: Vulnerabilities, attacks and possible mitigations. *CSI Transactions on ICT* 1, 135–141.

[11] Alcaraz C & Zeadally S (2015), Critical infrastructure protection: Requirements and challenges for the 21st century. *International Journal of Critical Infrastructure Protection* 8, 53–66.

[12] Chalamasetty GK, Mandal P & Tseng TL (2016), Secure SCADA communication network for detecting and preventing cyber-attacks on power systems. *Proceedings of the IEEE Power Systems Conference*, pp. 1–7.