DOI: http://dx.doi.org/10.14236/ewic/ics-csr2014.5

Towards Understanding Man-In-The-Middle Attacks on IEC 60870-5-104 SCADA Networks

Peter Maynard, Kieran McLaughlin
Centre for Secure Information Technologies (CSIT)
Queen's University Belfast
{*pmaynard01,kieran.mclaughlin*}*@qub.ac.uk*

Berthold Haberler
LINZ STROM GmbH
b.haberler@linzag.at

This paper investigates cyber attacks on ICS which rely on IEC 60870-5-104 for telecontrol communications. The main focus of the paper is on man-in-the-middle attacks, covering modification and injection of commands, it also details capture and replay attacks. An initial set of attacks are preformed on a local software simulated laboratory. Final experiments and validation of a man-in-the-middle attack are performed in a comprehensive testbed environment in conjunction with an electricity distribution operator.

Keywords: SCADA, Cyber-security, Man-in-the-middle attacks, IEC 60870-5-104

1. INTRODUCTION

This research investigates a series of experiments attacking Industrial Control Systems (ICS) that rely on IEC 60870-5-104 for telecontrol communications. Experiments investigating packet replay and man-in-the-middle attacks are presented, which illustrate the methods that may be used by attackers across a specific range of expertise and capabilities. Issues concerning the detection and prevention of such attacks are also investigated. These kinds of attack are of particular concern because of their potential to directly affect the reliable operation of the underlying ICS.

The experiments are conducted within two environments: first within a laboratory environment that uses software simulation of Supervisory Control And Data Acquisition (SCADA) master and slave device endpoints; and secondly within a test-bed environment in conjunction with an electricity distribution operator, which comprises part of the research in an EU FP7 project called PRECYSE. This project and the testbed environment will be discussed in more detail in due course.

First, we will examine the usage and cyber-security problems associated with the IEC 60870-5-104 protocol ("104" from here on), as well as the existing research in SCADA protocol related security issues.

2. BACKGROUND AND MOTIVATION

IEC 60870, is a collection of open standards written by the International Electrotechnical Commission (IEC) regarding the transmission of SCADA telemetry control and information. IEC 60870 was developed periodically in a hierarchical method between the years 1988 and 2000. It consists of six main parts and four companion sections. 60870, when discussed in the context of SCADA systems, normally referrers to the companion standard 60870-5-101. When this was released, in 1995, it detailed the complete transmission protocol, which allowed it to be used in production. This was originally written for serial communications, but with the subsequent release of the 60870-5-104 standard in 2000, it allowed for the same serial frames to be transmitted over TCP/IP. The 104 protocol is widely used in control communication for water, gas and electricity, and is particularly common in European utilities and related industries.

To the best knowledge of the authors, there is a lack of detailed publications attacking systems dependent on the 104 protocol. Possibly the closest related work has been done in Dondossola et al. (2009), in which Denial of Service (DoS) methods are used to determine the resilience of power control systems. There has been a variety of other published work on how to prevent or detect attacks, but there appears a lack of detailed information and discussion about the steps and methods used in potential attacks.

The key to successfully protecting systems is to understand the types of vulnerabilities and attacks which are possible. This paper therefore explores and explains in detail how these attacks are performed, allowing readers to see exactly what the effect on a target ICS might be, where the cyber-security issues are, and what measures may be needed to protect affected systems. Note that such information about attacks is already known to those who wish to use it, however it is generally not throughly examined in published literature.

3. EXISTING WORK

This section will cover some of the existing publications which are related to this paper. The existing work tends to be at a much higher level and does not explain in detail, with examples, how attacks would work in practice.

Much of the existing work is based on Modbus and DNP3. In comparison 104 is also vulnerable to many of the same kinds of attack. A key vulnerability for most systems is the lack of authentication or validation mechanisms for data communicated via 104, and similar protocols such as DNP3. Although standards and mechanisms exist to address these issues, their use in the real world is rare, due to operational concerns, legacy issues, costs, etc.

Robinson (2013) covers possible attack vectors such as lack of protocol security in MODBUS and DNP3. It discusses potential attacks like replay, man-in-the-middle and spoofing. It continues to list other common attacks and prevention methods, such as following good network polices. Also discussed are redundant systems to share the load if a system becomes compromised, whilst maintaining the original system to avoid down time and to monitor the compromised system. Constant monitoring of the network, specifically the source and contents of the 104 packets, will alert staff to abnormal activity. However, there are no explicit details of how the attacks could be performed.

Morris and Gao (2013) investigates attacks such as response and measurement injection, and command injection using the MODBUS protocol. The attacks have been performed in a laboratory setting, detailed in Morris et al. (2011). This paper details various levels of injection attacks ranging from naive injection which randomly injects values to complex injections, or target specific fields and values based on domain knowledge. It also outlines possible consequences, such as sporadic sensor measurements, altered system control schemes and altered actuator states. Which can result in partial communication disruptions right up to complete shutdown of devices.

Pietre-Cambacedes et al. (2011) brings to light issues which ICS system administrators might overlook, such as isolating the ICS network from the internet, being vulnerable to standard system attacks, and misconfiguration of firewalls. These mistakes could allow attackers to penetrate the network, and once they have gained access to the network they are much more likely to be able to perform man-in-the-middle attacks. More specifically related to this paper are “Obscure protocols/systems are naturally secure” and “We only have obscure protocols/systems”, if it runs over TCP/IP then it can be susceptible to attacks discussed.

Samineni et al. (2012) discusses ways in which it is possible to perform stealth and semi-stealth man-in-the-middle attacks using ARP spoofing. This is where the targets are left with no record in their ARP table of the attack. It explains what a man-in-the-middle attack is and how to prevent such attacks, Bruschi et al. (2003) details Secure ARP (SARP), which uses public/private keys digitally sign the messages to prevent ARP spoofing. Though this is not directly related to ICS, attacks like this are often overlooked, Pietre-Cambacedes et al. (2011). Yang et al (2012) details how ARP spoofing can be accomplished on smart grids. Gao et al. (2010) shows how command injection is done using ettercap and other techniques.

Timorin (2013); SCADAStrangeLove (2013) explains how to detect 104 devices on the network, they have also released python scripts which can identify and return the common address of a 104 device. The common address is an address used for all data contained within the 104 packet, used to identify the physical device. Using these existing scripts it would be possible to scan a network for specific 104 hosts. Network reconnaissance is the first step in a successful attack, covered in section 5.1. This script could be used to detect possible targets for a man-in-the-middle attack, by confirming the end device is using the 104 protocol and obtaining its address.

The attacks detailed in the above papers, replay, man-in-the-middle, spoofing and injection, despite being developed for other protocols can be applied to 104, which is also vulnerable, because 104 like MODBUS and DNP3 does not support authentication and verification of packets. In Dondossola et al. (2009, 2008) they cover the use of DoS attacks on networks running the 104 protocol and following industry security standards, such as VPN endpoint security. Similarly it is worth noting that

hardware tunnelling system for ICS which encapsulate 60870-5 using VPN have been developed by companies like Tofino. Finally, note that it is possible to access various ICS systems from the internet with the use of Google dorks[1] and search engines such as Shodan HQ[2], although it is also likely that many of these are research honeypots set up to capture attacks.

4. 104 PACKET PAYLOAD STRUCTURE

Before discussing the experiments, it is first necessary to briefly highlight the relevant features of the 104 protocol that are most significant to this work. Figure 1 shows the structure of an IEC 60870-5-104 packet payload, which is commonly refered to as the Application Protocol Data Unit (APDU). The APDU consists of two parts, the Application Protocol Control Information (APCI) and Application Service Data Unit (ASDU).

The APCI is used as a communication start and stop mechanism for the ASDU. The APCI contains a start character, 68H, a length field (containing the length of the APDU) and a control field. The ASDU contains the application data, such as the device common address and system readings, contained in information objects. Therefore this is the location of the most critical data for physical system, and this is where the focus will be for the attacks carried out in the experiments that follow.

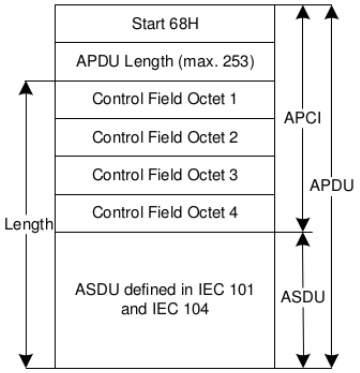

Figure 1: *APDU of the defined telecontrol companion standard*

It is worth noting that the IEC 62351 standard is specifically designed for preventing, among other things, eavesdropping, replay and spoofing. While this is compatible with the IEC 60870-5 series, it is rarely deployed in real systems due to legacy, cost and other issues.

[1]Google dork is using Google's advanced features to locate vulnerable systems.
[2]http://www.shodanhq.com/

5. ANATOMY OF AN ATTACK

We propose that attacks can be classified into distinct levels based on the knowledge required and sophistication of an attack. Table 1 visually shows the four levels of attacks. A similar system used for defining attack levels was proposed by Robinson (2013). The experiments that follow in Sections 6 and 7 are presented in such a way that an incremental level of knowledge and technical sophistication is required at each stage in order to successfully execute each attack. The motivation is to both understand the attacks and classify the kind of attacker that would be able to execute such attacks.

Threat Number	Level of attack
4	Advanced
3	Experienced
2	Novice
1	Accident

Table 1: *Attack Levels*

An attack will generally follow three steps: detection, where targets are identified; Capture, where data is collected in preparation for an attack; and finally Attack. Figure 2 visually breaks down the sequence of events, which are discussed in the following three sections.

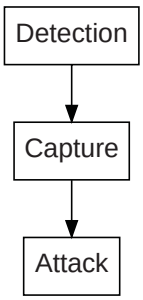

Figure 2: *Anatomy of an attack*

5.1. Detection

There are two ways to detect 104 devices attached to a TCP/IP network, passive and active. Passive is where no action is taken. An interface is set into promiscuous mode, where it will accept all packets on the wire. The packets can be monitored for 104 traffic with tools such as Wireshark and tcpdump. Once 104 traffic has been identified it can be logged for future use, such as replay attacks or parsed to identify attack targets to be used in a future attack.

Active detection is where packets are sent out from the machine to try to invoke a response from a 104 device. For example, Timorin (2013) has written a python script which probes a list of IP address, and

returns the common address of a 104 device if it exists.

It accomplishes this by sending a test APDU, the target replies with a confirmation then it sends a Start Data Transfer (STARTDT) packet. The reply is then checked for the common address, if not found it will send a C_IC_NA_1 Broadcast. This is a 104 interrogation command and is another method used to obtain the device address.

The passive method may be used by a novice hacker who is not sure what is going on in the system (threat level 2 in Table 1), or by a more sophisticated attacker trying to avoid detection. The active mode will most likely be used in conjunction with the passive technique to confirm the existence of the 104 device and to obtain detailed information. It will be used right up to threat level 4, advanced level attack.

5.2. Capture

In this stage data is captured which can be used to attack targets identified during detection. It is possible to capture data by monitoring and performing a man-in-the-middle attack. Typically data is captured using a span port, also called port mirroring. which is used on a network switch to send a copy of all packets on the network to a specific port. If an attacker is able to gain access to a machine which is connected to a span port or is able to trick the switch into becoming a span port, or gain administrative control of the switch, they will be able to capture all the traffic passing though, including data being transmitted by our targets.

Another way is to get in between the two (or more) targets and capture data. One way this can be done is with ARP spoofing, as explained in Samineni et al. (2012); Yang et al (2012). Figure 3 shows a successful ARP spoof. ARP spoofing takes advantage of the fact that ARP, like 104, does not inherently support verification and authentication. 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. Now the attacker can view and edit all packets being sent between the targets.

There are alternatives to ARP spoofing, 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. Most modern switches have mitigation support for this, provided

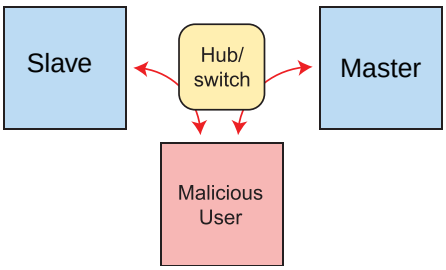

Figure 3: *Successful ARP Spoofing (CC BY-SA 3.0)*

they are configured correctly. In certain cases this attack could cause the switch to fail.

5.3. Attack

The final stage brings together all the other stages. Since this paper focuses on man-in-the middle attacks, this section is limited to directly related attacks, such as replay and injection.

A replay attack is where valid data transmissions are captured and replayed by an attacker. Packets can be captured at the source of the transmissions or, intercepted via man-in-the-middle. The data may be replayed without any modification or with. If the data is replayed with out modification, in a ICS environment, it might mean duplicate readings being sent to the monitoring station, or commands to the controlling direction. This can cause disruption to the network or even damage. Such attacks would be performed by an inexperienced attacker, or someone who does not fully understand the system but is just experimenting. Up to level 2 on the threat meter. This is covered in more detail in Section 6

Injection attacks, as detailed in Morris and Gao (2013), is where values are modified before they reach their target. A man-in-the-middle attack would be performed at threat level 2 and above. In the case of ICS a man in the middle attack can be a very serious, as readings might not be relayed correctly back to the control center, instructions may not be delivered on time or at all, (without any visible consequences). Section 7 explains it in more detail. First, however, we begin by exploring replay attacks.

6. EXPERIMENT ONE: REPLAY ATTACK

In this experiment a replay attack is performed with packets captured from software based emulators. This is a relativey simple form of attack, hence it is most likely to be performed by an attacker unfamiliar with ICS and lacking the domain knowledge. According to the classifications in Table 1, this would be level 2 or below. It is also possible

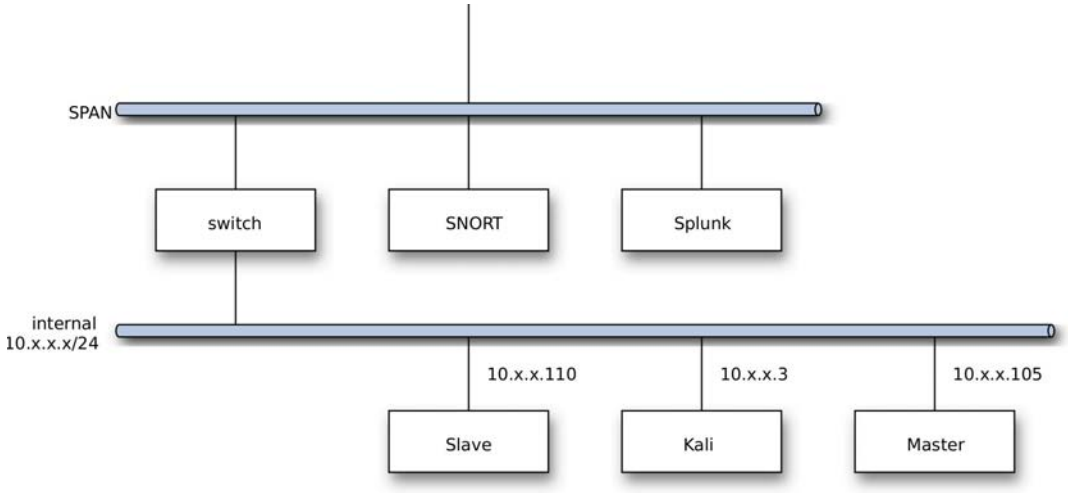

Figure 4: *Local Test Network*

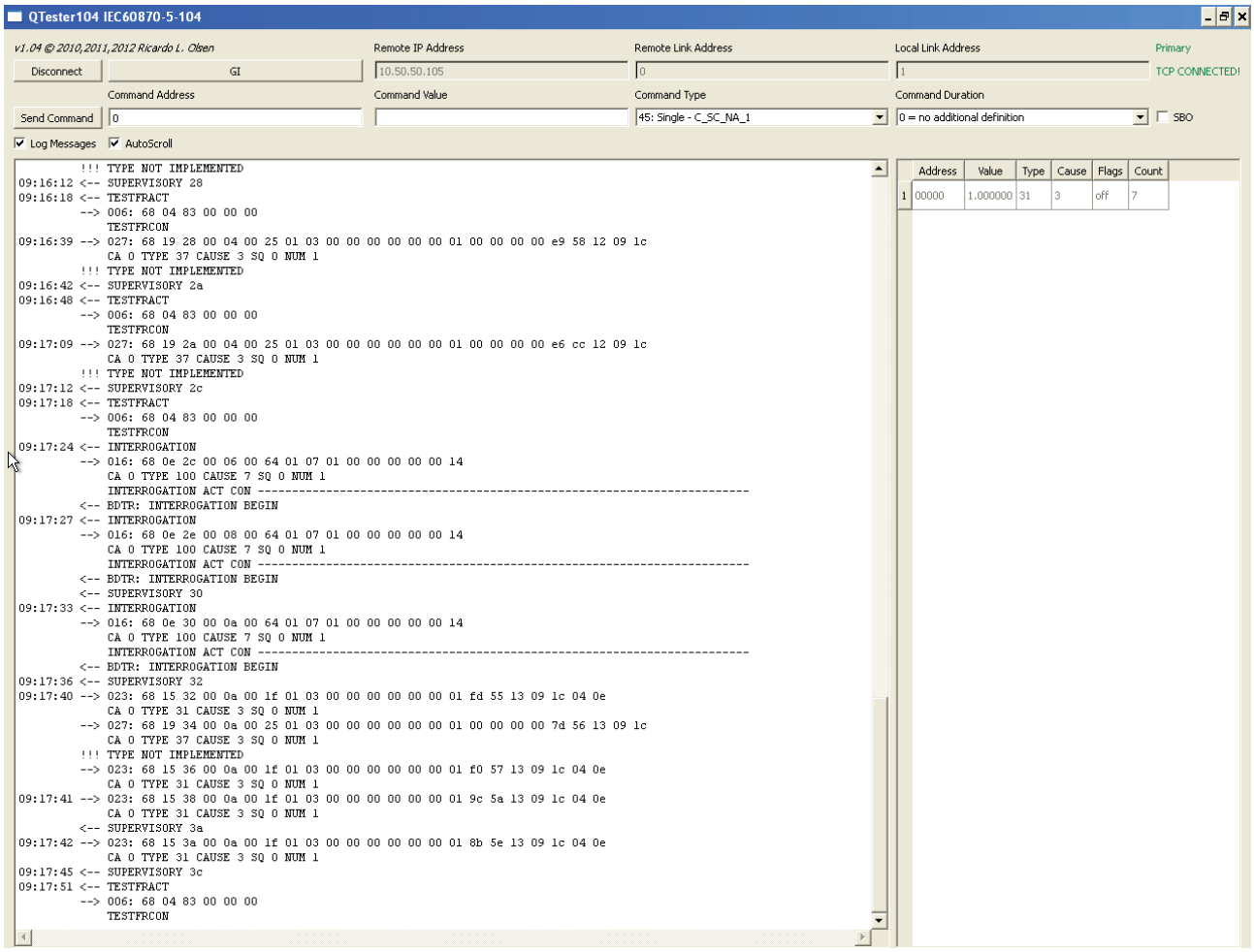

Figure 5: *Qtester connected to WinPP104 acting as the master, showing M_DP_TB_1, M_IT_TB_1 and C_IC_NA_1*

to accidentally reproduce this with misconfigured network devices, such as network switches and monitoring equipment. The goal of the attack is to capture 104 packets and replay them against the targets. This aims to simulate an inexperienced attacker messing around on the network. Figure 4 details the test network, which consists of a master and slave 104 device, SNORT IDS,

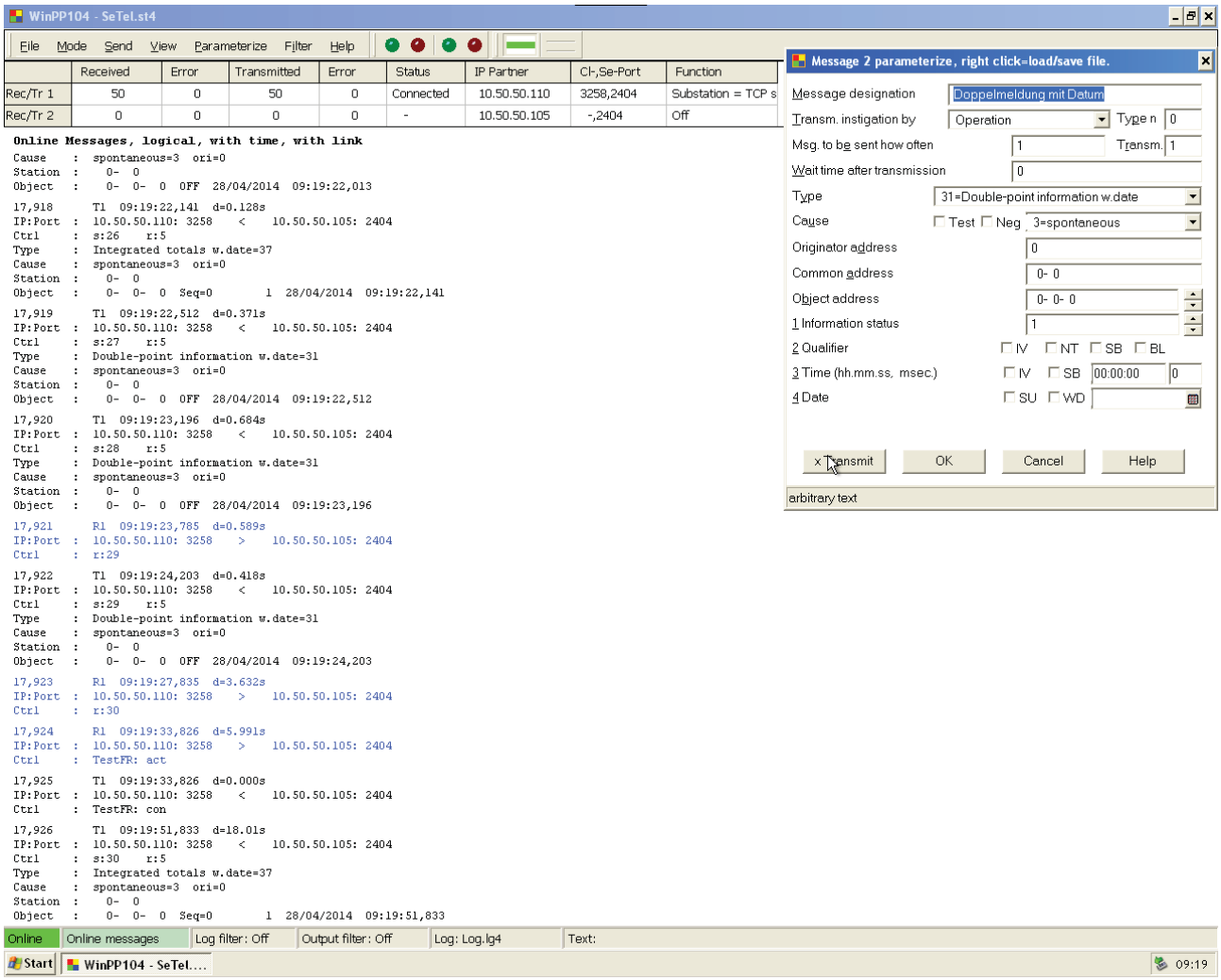

Figure 6: *WinPP104 connected to QTester acting as the slave*

Splunk log analyzer and a Kali Linux attack platform.

Qtester[3] is used to emulate the master 104 device. this emulator makes it possible to poll and view data from the substation system, and to send interrogation commands such as C_IC_NA_1. Qtester is an open source project, and supports many platforms with the use of the QT library. As stated above, it is possible to view data from the substation. It was chosen for these reasons as other systems either do not support 104 or do not visually display data received.

The substation slave device is simulated using the WinPP104[4] packet simulator. It is used to generate measurements and reply to interrogation commands from the master. WinPP104 was chosen due to its ability to test received packets and determine if they are valid, as well as being able to generate specific packets on request. These features are fundamental to confirming and developing an attack.

Figure 5 is a screen shot of Qtester running. The menu bar at the top allows us to specify the details of the slave to monitor. Such as the IP address, Link Address and our local link address. The top menu also provides the feature to send command instructions to the slave, e.g. C_CS_NA_1. The left hand main pane outputs the log messages. This details what packets have been sent and received, along with any errors. The pane on the right details the data received from the slave. It details the slave address, the value, type ID, cause of transmission, flags and packet count.

Figure 6 is a screen shot of WinPP104 running as the 104 slave, which is connected to Qtester master. The main pane of WinPP details 104 packets sent and received by the emulator. This is also where invalid packets will be displayed. The top table displays statistics of the connection, as well as the IP and port

[3]QTester http://sourceforge.net/projects/qtester104/
[4]WinPP104 http://www.ppfink.de

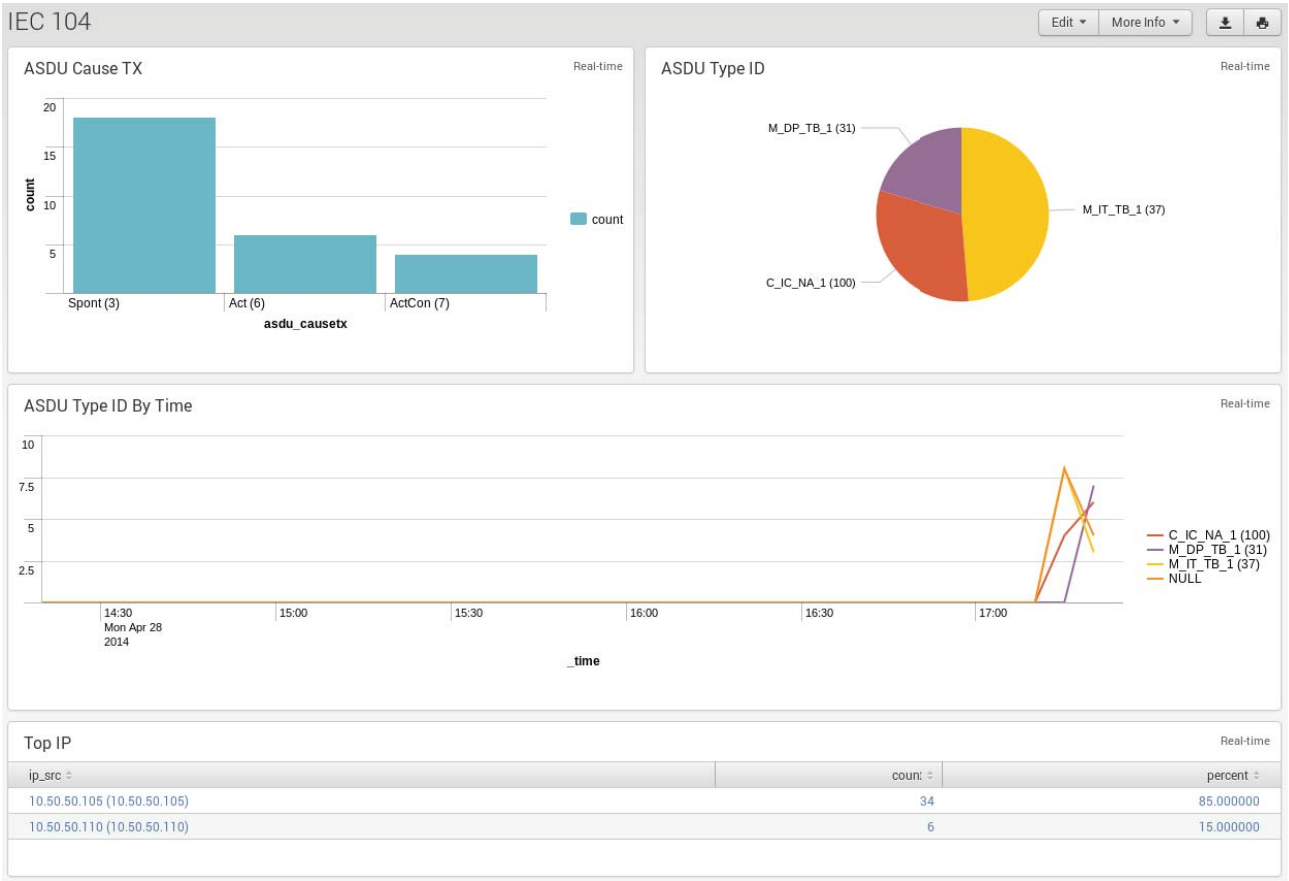

Figure 7: *Splunk showing 104 traffic between the slave and master*

address of the device it is connected to. The window at the top right, allows us to specify the 104 packet to be transmitted to the master.

From here it is possible to set the frequency of the packets to be sent or just a one off transmission. The type can be selected, in this case it is type ID 31, 'Double point information with timestamp'. The timestamp can be specified in the lower half of the dialog. The cause of transmission can also be specified as well as the originator, common and object addresses, in this case zero. Finally the information status and qualifiers can be set.

WinPP104 was chosen for these experiments because it allows great flexibility to generate an array of packets which can be sent over the network, as well as validating received packets.

Connected to the span port of the switch is SNORT[5]. SNORT is running custom rules to detect anomalous 104 traffic, which is described in Yang et al. (2013). These signatures were developed to detect any packets which do not confirm to the 104 protocol. This could be packets which look correct, but have an invalid combination of fields. Splunk is used to visually display alerts from SNORT and monitor the network for anomalies. Figure 7 shows 104 traffic between the slave and master. The top left bar chart shows a count of the different cause of transmissions detected. The pie chart on the right details the total 104 types, and the line graph at the bottom shows the number of types detected over time.

Packets used for the replay attack are captured from the span port of the switch. After capturing a packet dump of significant duration, Wireshark is used to remove non 104 packets from the capture. This leaves just the packets from the target machines, which consist of the TCP/IP handshake, initialization 104 packets, STARTDT, M_SP_TB_1 readings from the slave, and a few TESTFR packets from the master checking which is used to check the link activity.

The Kali attack platform is where the packets are replayed from. TCP Replay[6] is the suite used to replay the captured packets. The replayed packets are detected by the SNORT IDS and are seen traveling along the network. Though the replayed

[5]SNORT http://www.snort.org/

[6]tcpreplay http://tcpreplay.appneta.com/

packets are not accepted at the application layer, due to the packets being dropped by the kernel's TCP/IP stack. They are dropped because tcpreplay does not modify the SYN and ACK values before it is replayed. Figure 8 shows Wireshark detecting the replay, by highlighting it with the retransmission flag.

6.1. Discussion

It should be possible to detect the malicious activity investigated above using existing Snort rules and statistical analysis of network traffic and logs.

Furthermore, since the packets replayed will not be accepted at the application level of most systems, this level of attack should not directly affect the operation of the ICS. Even so, the attack will most likely not be caught by network firewalls, as it will appear to be valid traffic, unless a stateful IDS/firewall is used to track the TCP streams. On a low bandwidth network or a network already under stress such an attack could possibly bring down the network and increase the likelyhood of devices to timeout.

With further effort it would be possible to replay the captured packets so that they are not dropped by the kernel and are accepted at the application layer. This could be accomplished with a python script, which initiates the TCP handshake and manages the sequence numbers correctly. There are existing applications[7] which can do this, though at the time of writing the current builds were not stable enough to work with.

7. EXPERIMENT TWO: MAN-IN-THE-MIDDLE ATTACK

A man-in-the-middle attack is where traffic is intercepted between two hosts and the data monitored or modified in transit without being detected by the victims, see Figure 3. The same principle was used in the original stuxnet attack, which monitored the activity and replayed an altered value which looked similar to the original. It is more dangerous than a replay attack as it happens in real time and is harder to detect. It will most likely be used by an experienced or advanced attacker (threat level 3 and above).

This final man-in-the-middle experiment is conducted within a realistic electricity distribution SCADA environment established by the PRECYSE FP7 project. PRECYSE aims to investigate cyber threats and develop tools and methods that detect live breaches of cyber-security, to provide early warning alarms, and to issue countermeasures to mitigate incidents that threaten the operation of Critical Infrastructure systems. The development and construction of the comprehensive testbed environment has been undertaken by LINZ STROM GmbH, who are an electricity Distribution System Operator in Austria.

The Linz testbed site primarily comprises a SCADA Process Network, Substation Process Network and an Office Domain Network. The environment is presented in figure 9. PRECYSE security domains and devices are deployed to enable a series of security monitoring and countermeasure features. These components are mostly outside the scope of this paper, except for the Substation and SCADA Server network elements (highlighted in boxes A and B), which are the focus of the following experiment. These two domains in the test-bed contain RTU devices and SCADA control servers in a configuration that mirrors as close to a real operational environment as possible. The 104 protocol is used for monitoring and control of this ICS. This experiment monitors for specific 104 traffic in order to carry out a targeted attack against the core ICS control communications. Once specific SCADA packets are found they are modified and forwarded to the SCADA server where they will be accepted as if they originated from the legitimate RTU device. This attack takes advantage of the fact that the 104 protocol does not specify any authentication or verification of the packets upon delivery.

It works in the same attack structure described in section 5, which is detection, spoof and attack. Assuming the attacker has become aware that 104 devices and data are present on the network, the first stage of this attack is to initiate an ARP spoof using ettercap's ARP spoof module. This allows the attacker to view all of the packets being transmitted by the targets. The detection and modification of 104 packets is done using a custom ettercap plugin which is designed to seamlessly alter 60870-5-104 packets in transit.

Our attack plugin has been developed to monitor the two way connections for specific 104 packets, once a packet matches the specification it is dropped and another sent in its place. It is important to note that it sends TCP ACK, and essentially duplicates the TCP/IP fields to ensure the client will accept the newly crafted value, it also manages the checksums to ensure the packets are accepted.

Two implementations of the man-in-the-middle attack are now presented. The first is in the lab environment to provide a proof of concept that the plugin operates as expected. The second is in the Linz testbed site where a specific SCADA operation is targeted.

[7]Wireplay https://github.com/abhisek/wireplay and tcplivereplay, which is part of the TCP Replay project.

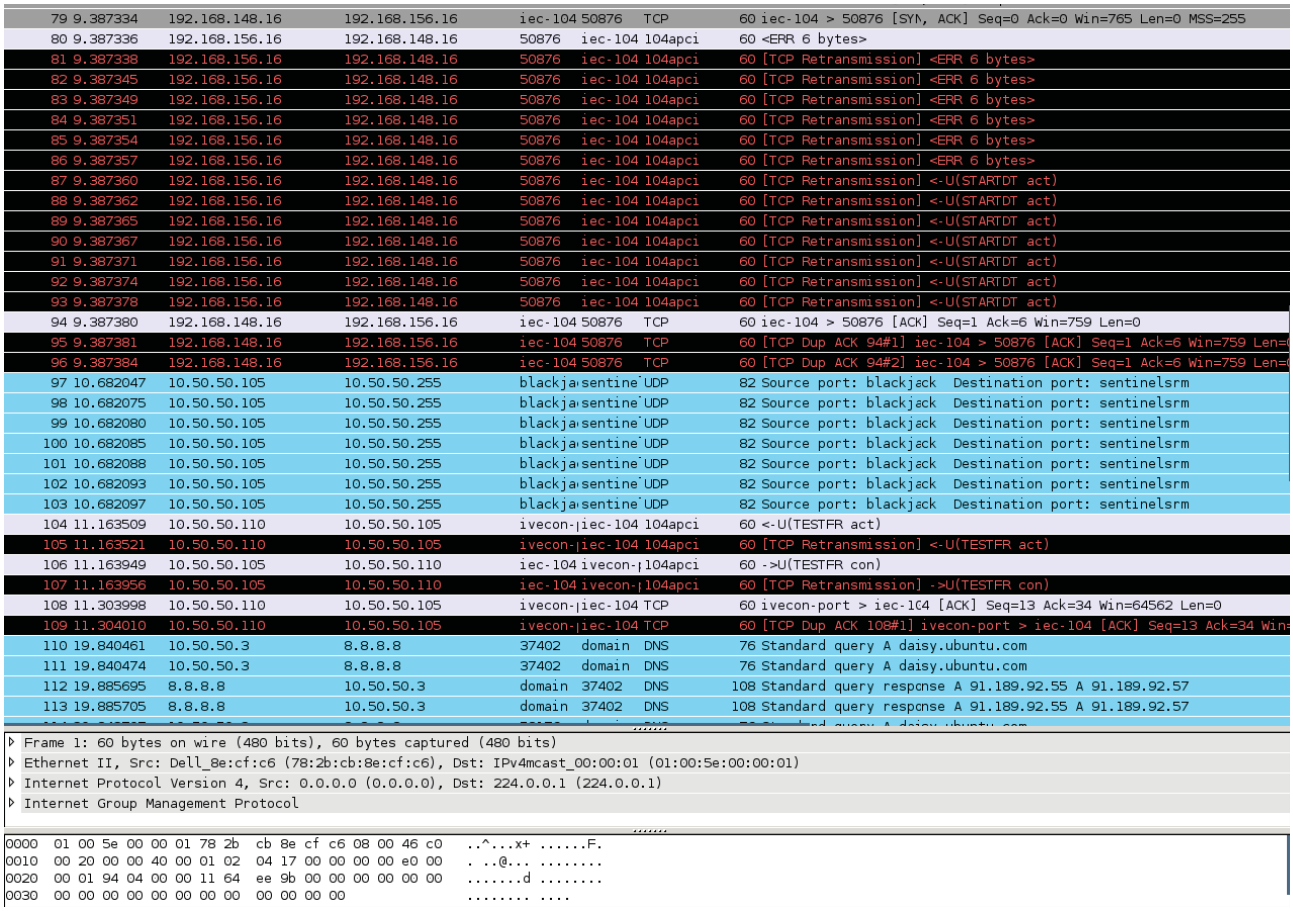

Figure 8: *Network capture of the replayed packets*

7.1. Laboratory Simulated Environment

The first implementation of our attack is to alter the Cause of Transmission (CoT) field, as introduced in previous sections. This is performed on our local test network, configured in the same manner as Experiment One, see figure 4. For the purposes of Experiment Two, the emulated slave RTU device has been configured to transmit a spontaneous 104 packet of type ID 30, M_SP_TB_1 - 'single-point information with time tag CP56Time2a', every thirty seconds. During the experiments the network is monitored using SNORT, which is configured to use the custom 104 IDS rules developed by Yang et al. (2013). When executing this attack, the attack plugin monitors for 104 packets with a type ID of 30, coming from the slave to the master. When an appropriate packet is detected, the original packet will be dropped by the man-in-the-middle and a new packet crafted, with a modified CoT. The CoT value is used to route the ASDU to the correct program or task for processing. CoT values can use the following number ranges, <1-13> and <20-41>, numbers from <14-19> and <42-43> are reserved for future use. So we shall modify the CoT from <3> to <42> to prove the plugin operates as expected.

Figure 10 shows a packet capture of an unmodified packet, begin sent from the slave to the master. As you can see the CoT value is still set to <3>, spontaneous.

Figure 11 shows a packet capture of the packet after it has been processed by ettercap. Here you can see that the CoT field is now of type <42>, which is unused and causes SNORT to trigger an alert for 'Suspicious Value of Transmission Cause Field' (Yang et al. (2013)), see figure 12. This rule checks to make sure that each packet has the correct cause of transmission based on the specification of the 104 protocol.

7.2. Real Testbed Environment

The second implementation of the attack is to intercept and modify information contained within the ASDU. Using the PRECYSE electricity distribution testbed, we have a scenario where the I/O port of an RTU is being turned on and off at regular intervals for monitoring and testing purposes. This setup is intended to mirror the communications that may occur in a real situation where an earth fault in the physical electrical grid occurs. In such a case a message will be transmitted from the fault location

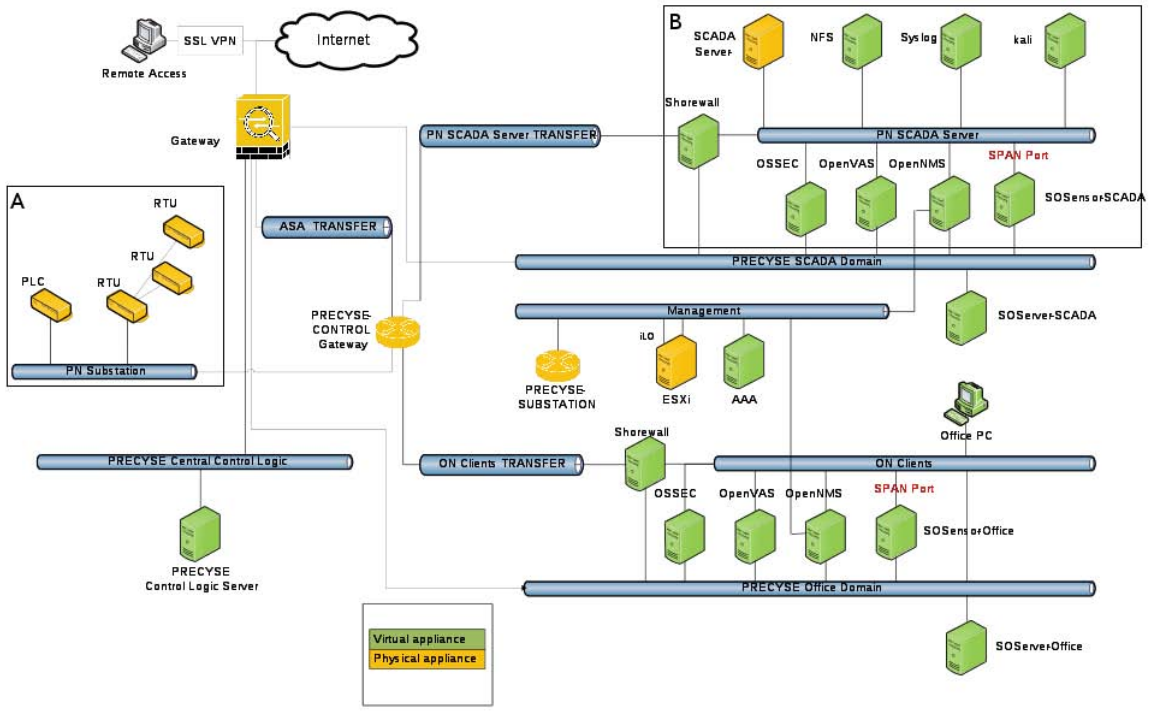

Figure 9: *Testbed Network*

```
Frame 73: 77 bytes on wire (616 bits), 77 bytes captured (616 bits)
Ethernet II, Src: RealtekU_6d:a8:fc (52:54:00:6d:a8:fc), Dst: Netgear_e3:94:bd (00:
Internet Protocol Version 4, Src: 10.50.50.105 (10.50.50.105), Dst: 10.50.50.110 (1
Transmission Control Protocol, Src Port: iec-104 (2404), Dst Port: webtie (3342), S
IEC 60870-5-104-Apci: ->I(2,1)
IEC 60870-5-104-Asdu: 0,0->0 M_SP_TB_1 Spont IOA=0 'single-point information with t
  TypeId: M_SP_TB_1 (30)
  .000 0001 = NumIx: 1
  ..00 0011 = CauseTx: Spont (3)
  .0.. .... = Negative: False
  0... .... = Test: False
  OA: 0
  Addr: 0
  IOA: 0
▽ IEC 60870-5-104-Asdu: Value
```

Figure 10: *Unmodified APDU with spontaneous CoT*

to the SCADA server which will indicate via the user interface that a fault signal has been received.

The objective for this attack is therefore to intercept the ‘ON’ value so that an ‘OFF’ value is received by the SCADA server, thus hiding the real status of the physical system from the engineer in the operations control centre. 104 stores this data within the information element, each ASDU may have zero or more information objects depending on the type of data being transmitted. This attack would be at level three and four of the skill level table, due to the required domain knowledge to accomplish it. To identify the correct information element and ASDU requires a deep understanding of the network configuration, for an attacker without this knowledge it may be possible to gain such knowledge by using network wide packet dumps and analysis. This would take time to complete and would require some skill to remain undetected over a long period.

The ON/OFF value is transmitted using the M_SP_TB_1 structure, as specified by the standard, the structure contains an information element of

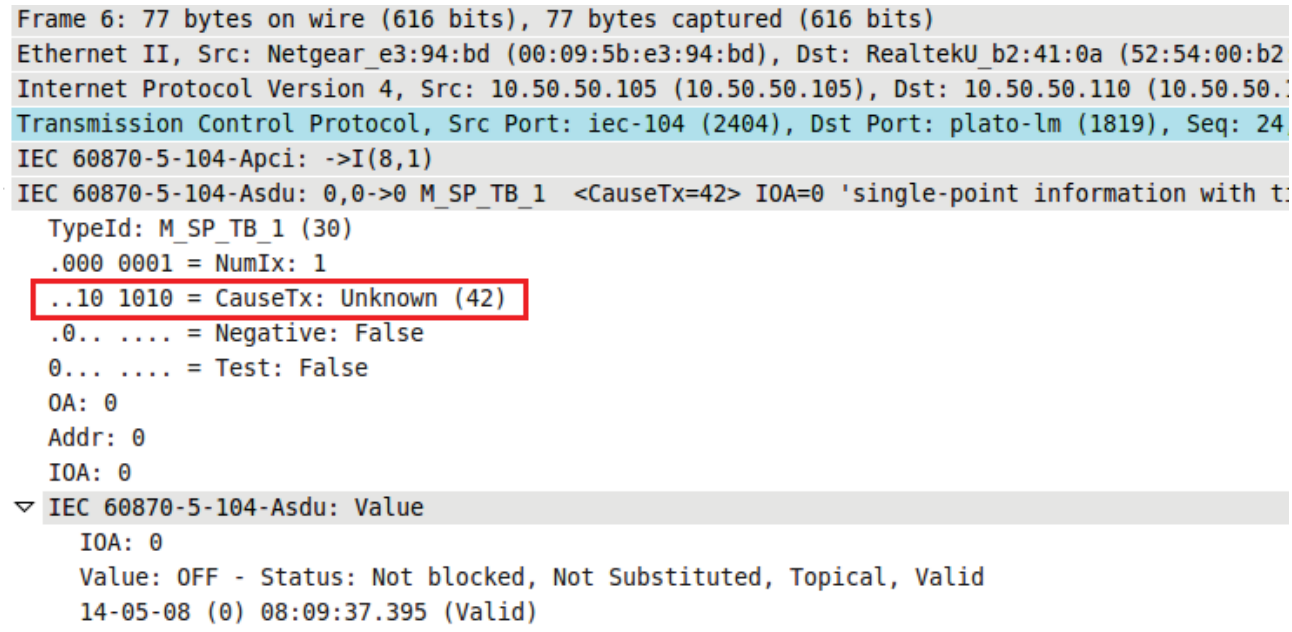

Figure 11: *Modified APDU with altered CoT Unused 42*

```
6666621 – Suspicious Value of Transmission Cause
    Field
Alert: PN_SCADA_enclave abnormal network
    behaviour. Anomalies in IEC 60870–5–104,
    suspicious mismatch of CoT and APDU Type
    Identification values in monitor direction.
     Possible protocol fuzzing or MITM attack
    behaviour. Please investigate further. Do
    not block IP, it will inhibit process
    communication.
```

Figure 12: *Description of SNORT Suspicious Value of Transmission Cause Field rule.*

Single point information (SIQ). The first bit of the SIQ is the Status (SPI) field, which is what stores the ON/OFF flag, 1 being ON and 0 OFF.

Figure 9 details the test network used. The network PN_SCADA_Server, Section B is where the control and attack machines are located. The ‘SCADA server’ is the main control unit, and is the target for this attack. Kali is used to simulate a compromised machine and is used to perform the attack, and is also located here. The monitored RTU devices are located in the PN_Substation, section A, of the network, they transmit the M_SP_TB_1 packets containing the switch ON/OFF value. The first stage of the attack is to perform a ARP poisoning against the targets, ‘PRECYSE CONTROL Gateway’ and ‘SCADA server’, using ettercap. This will allow the attacker to become a ‘man in the middle’ and monitor the communications. The next stage is to monitor for M_SP_TB_1 ASDUs which contain an information element with the SIQ’s SPI field set to ON. If the value is already set to OFF then it will be allowed to pass through. Once an ASDU is positively matched the packet will be dropped and a new packet sent, with the SPI value set to OFF. The outcome of this is that the control station will only ever see packets with the SPI value zero, OFF. Figure 13 shows the power circuit, and displays any alarms to the operators. The above attack will result in operators at the control desk being shown that the high voltage switch is off, this could cause the operator to take negative action, such as activating a backup circuit, resulting in a possible malicious physical state of the electricity grid.

7.3. Discussion

The first attack, in the lab environment, was predominantly a first step towards proving the function of the attack plugin. Nonetheless, in a real ICS it is possible that an attacker may attempt to modify SCADA packet payload fields in order to probe the system to see what happens, or as part of a slightly more sophisticated attempt to learn about the environment by employing a fuzzing technique. Such scenarios are most likely to arise if the attacker does not have a good level of domain knowledge about the ICS or the protocols in use.

In contrast, the type of targeted attack in the testbed environment requires a large amount of domain knowledge and will likely be done for specific malicious purposes by an attacker with a clear intention. This attack could only be achieved by someone with abilities of level three and above. It is important to note that the means to achieve the attack are not limited only to ARP spoofing. It is possible to perform DNS poisoning and CAM

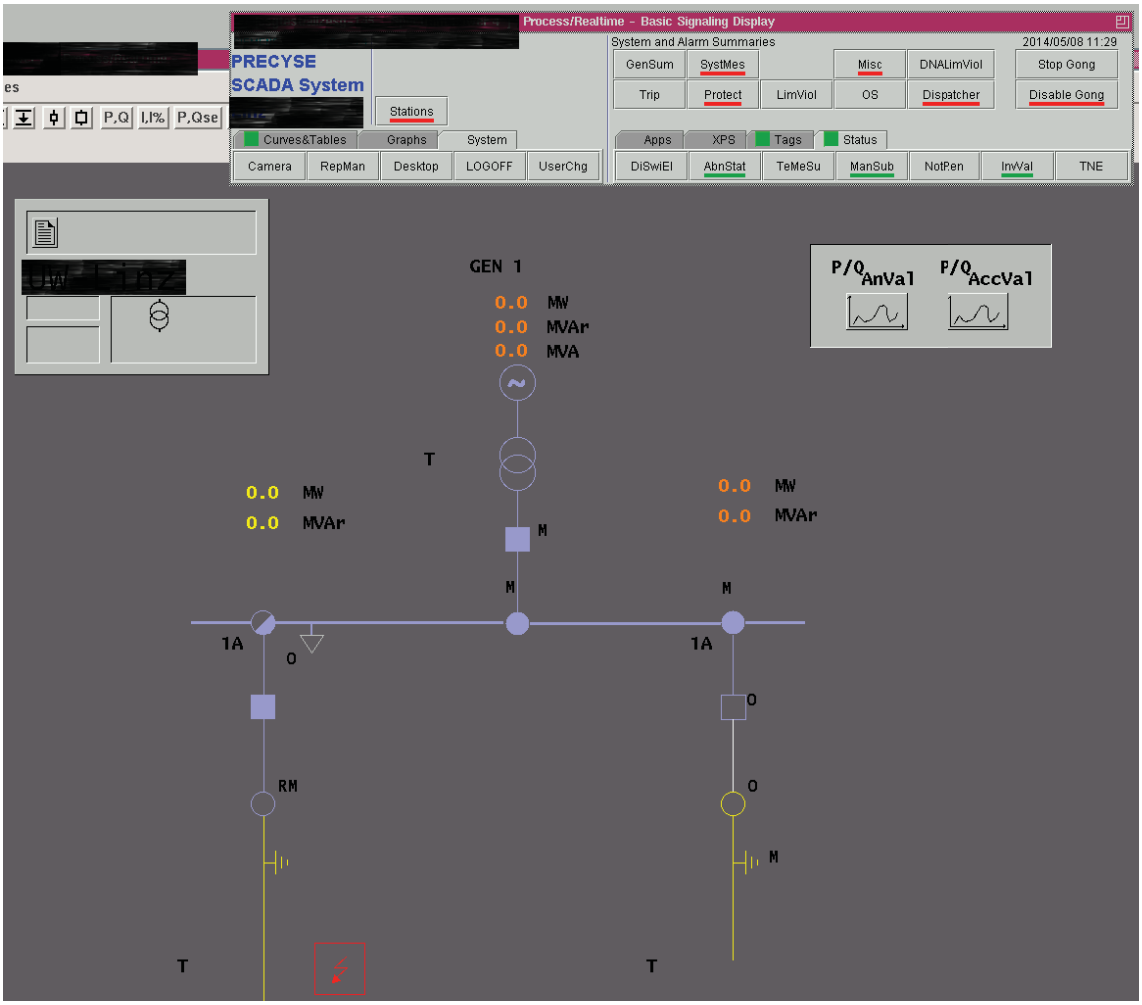

Figure 13: *GUI of the SCADA server showing a simplified example of an electric schematic*

switch overflow, which are not covered in ICS related materials to date, but are worth analysing further.

It is worth noting that the Kali machine provides a launching point for the attacks in these experiments. In an attack on a live system, this launch point is likely to be via malware infection, or some related intrusion into the network. Analysing this path of infection is not the focus of this study, rather the aim is to demonstrate exploitation of inherent vulnerabilities in the SCADA communications. Finally, we would highlight that although in this experiment we have only modified the COT and SPI fields, the plugin tools developed here allow changes to any field or the ability to substitute the packet with any other packet without limitation.

8. CONCLUSIONS

This paper has examined a series of cyber attacks against SCADA systems using the IEC 60870-5-104 protocol. These range from some basic replay style attacks to a more sophisticated man-in-the-middle attack targeting a specific operation within an electricity distribution testbed environment. One of the main purposes of this work was to enhance the existing literature in relation to SCADA based attacks; particularly man-in-the-middle attacks, which are not as well explored as DoS, packet injection, device discovery, reconnaissance methods, etc.; and particularly the 104 protocol, which is not as well covered as DNP3 or MODBUS in comparison.

Over the course of these experiments it has been shown how attackers with varying capabilities and domain knowledge can compromise the integrity of SCADA communications within an ICS. At the most serious end of this scale it has been demonstrated how a man-in-the-middle attacker could hide an earth fault condition from SCADA server in an electricity distribution scenario. Successfully tampering with such a core function of the SCADA has clear implications for system functionality, reliability and safety.

Although it is not the main intention of this paper to explore detection and mitigation strategies, it was shown that SCADA-specific IDS rules were able to indicate the presence of attacks in the replay attack experiments and in the man-in-the-middle experiment focussing on tampering with the CoT. The main intention following from this work is to develop a comprehensive strategy and set of tools to enable the detection of sophisticated man-in-the-middle style attacks, of the final type investigated. This work will focus on a defence-in-depth approach fusing information from multiple monitoring points within the SCADA network, in order to cover multi-stage attacks where different options of attack exist at each stage.

For example, the initial steps used in the presented attacks used ARP based attacks. It is important to note that vendors allow methods to detect and secure ARP from poisoning, e.g. Cisco uses 'port security' and DHCP snooping combined with Dynamic ARP Inspection. As already discussed, other Layer 2 attack points are feasible such as switch CAM table over flows, and proper monitoring and detection at this layer is clearly one important aspect. It is intended that further work will also be carried out to enhance the presented attacks to explore other options at each step in the attack.

REFERENCES

Bruschi, D., Ornaghi, A., and Rosti, E. (2003) S-ARP: A secure address resolution protocol. In: *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*. 66–74.

Dondossola, G. et al. (2008) A laboratory testbed for the evaluation of cyber attacks to interacting ICT infrastructures of power grid operators. In: *SmartGrids for Distribution, 2008. IET-CIRED. CIRED Seminar.* 1–4.

Dondossola, G. et al. (2009) ICT resilience of power control systems: Experimental results from the CRUTIAL testbeds. In: *IEEE/IFIP International Conference on Dependable Systems Networks, 2009. DSN '09.* 554–559.

Gao, W. et al. (2010) On SCADA control system command and response injection and intrusion detection. In: *eCrime Researchers Summit (eCrime)*. 1–9.

Morris, T., Vaughn, R., and Dandass, Y. S. (2011) A testbed for SCADA control system cybersecurity research and pedagogy. In: *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research, CSIIRW '11*. New York, NY, USA, 27:127:1.

Morris, T. H. and Gao, W. (2013) Industrial control system cyber attacks. In: *First International Symposium for ICs & SCADA Cyber Security Research 2013*. Leicester, U.K., 22–29.

Pietre-Cambacedes, L., Tritschler, M., and Ericsson, G. N. (2011) Cybersecurity myths on power control systems: 21 misconceptions and false beliefs. *IEEE Trans. Power Del.*, 26 (1). 161–172.

Robinson, M. (2013) The SCADA threat landscape. In: *First International Symposium for ICs & SCADA Cyber Security Research 2013*. Leicester, U.K., 30–41.

Samineni, N. R., Barbhuiya, F. A., and Nandi, S. (2012) Stealth and semi-stealth MITM attacks, detection and defense in IPv4 networks. In: *2012 2nd IEEE International Conference on Parallel Distributed and Grid Computing (PDGC)*, 364–367.

SCADAStrangeLove (2013) *SCADA StrangeLove: SCADA security deep inside* Available from http://scadastrangelove.blogspot.co.uk/2013/11/scada-security-deep-inside.html

Timorin, A. (2013) *atimorin/PoC2013* Available from https://github.com/atimorin/PoC2013

Yang, Y. et al. (2012) Man-in-the- middle attack test-bed investigating cyber-security vulnerabilities in smart grid SCADA systems. In: *International Conference on Sustainable Power Generation and Supply (SUPERGEN 2012)*, 1–8.

Yang, Y. et al. (2013) Intrusion detection system for IEC 60870-5-104 based SCADA networks. In: *2013 IEEE Power and Energy Society General Meeting (PES)*, 1–5.