International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

http://ijcjournal.org/

Supervisory Control and Data Acquisition (SCADA) System Forensics Based on the Modbus Protocol

John Onyiego[a*], Odira Elisha Abade[b]

[a,b]*School of Computing and Informatics, University of Nairobi, Kenya*

[a]*Elinsco systems, Nairobi, Kenya*

[b]*Scala Institute of Advanced Computing and Digital Literacy, Nairobi, Kenya.*

[a] *Email: jonyiego@gmail.com*

[b]*Email: eabade@uonbi.ac.ke*

Abstract

Supervisory Control and Data Acquisition (SCADA) has been at the cored of Operational Technology (OT) used in industries and process plants to monitor and control critical processes, especially in the energy sector. In petroleum sub-sector, it has been used in monitoring transportation, storage and loading of petroleum products. It is linked to instruments that collect and monitor parameters such as temperature, pressure and product densities. It gives commands to actuators by the use of the application programs installed on the programmable logic controllers (PLCs). Earlier SCADA systems were isolated from the internet, hence protected by an airgap from attacks taking place on interconnected systems. The recent trend is that SCADA systems are becoming more integrated with other business systems using Internet technologies such as Ethernet and TCP/IP. However, TCP/IP and web technologies which are predominantly used by IT systems have become increasingly vulnerable to cyberattacks that are experienced by IT systems such as malwares and other attacks. It is important to conduct vulnerability assessment of SCADA systems with a view to thwarting attacks that can exploit such vulnerabilities. Where the vulnerabilities have been exploited, forensic analysis is required so as to know what really happened. This paper reviews SCADA systems configuration, vulnerabilities, and attacks scenarios, then presents a prototype SCADA system and forensic tool that can be used on SCADA. The tool reads into the PLC memory and Wireshark has been to capture network communication between the SCADA system and the PLC.

Keywords: SCADA; IT; DCS; ICS, OT; PLC; Modicon.

----

* Corresponding author.

1. Introduction

Modern Critical national Infrastructure such as electricity, Oil and gas, water and manufacturing industries rely heavily on computer systems, networks, control systems and embedded devices in order to provide safe and reliable operations. Supervisory Control and Data Acquisition (SCADA) systems form part of a process control, which are used in manufacturing processes, chemical plant and refinery operations and even for automation and climate control in buildings. In a typical SCADA system, sensors acquire data pertaining to process behavior. This data is passed to control algorithms implemented in the SCADA systems [1]. SCADA systems are used to automate industrial processes, such as power generation and distribution, gas and oil pipelines as well as water and waste management [2] . Their primary design requirement is safety, which typically requires real-time response to changes in the monitored processes and an ability to handle harsh working environment; they were never designed to withstand cyber-attacks of any kind. Early SCADA systems were deployed in specialized isolated networks, which are not connected with corporate networks or the Internet. Thus, they were protected from remote cyber-attacks by virtue of not being accessible over the Internet. SCADA systems let you monitor and control various remote functions and processes by using modem communication links between master and remote locations. The smooth and reliable operation of SCADA systems is vital for such sectors of critical infrastructure such as energy, water and transportation where both data acquisition and control are critically important. Major SCADA systems collect data, monitor and control field instruments by the help of Programmable Logic Controllers (PLCs) that act as the slaves to the SCADA system. A PLC has an application program, a firmware and the hardware itself that has several security vulnerabilities from its design. As they cover large geographical areas, SCADA systems typically use Wide Area Networks (WAN). The communication infrastructure may be satellite, radio, power line based and any combination of the above. These systems are designed to provide real time sensor data on the physical dynamics. A SCADA architecture comprises two levels: a master or client level at the supervisory control center and a slave or data server level that interacts with the processes under control. In addition to the hardware, the software components of the SCADA architecture are important. SCADA systems generally consist of sensors, actuators, programmable logic controllers (PLCs), and a human machine interface (HMI) [3]. A PLC is deployed at a remote field site to provide immediate monitoring and control of a physical process, it functions as a slave. It sends control signals to the device under control, acquires data from them which it transmits to the master terminal units. HMI and other SCADA services (such as engineering workstation and historian) run at a control center and provide the means for operators to remotely observe and control the processes [4]. Worms such as Stuxnet [5] which was uncovered in 2010, proved a new sophisticated level era of attacks against a PLC based system. It was able to exploit other machines to initialize attacks on them. It demonstrated a real sophisticated cyber security attack that catastrophically affected several areas of PLC-BS. By attacking the software (HMIs—WinCC, PLC codes—Siemens Simatic Step7, PCs—Windows, etc) and faking values, Stuxnet severely affected crucial field devices taking advantage of multiple Windows zero days vulnerabilities [6]. There are many other malware attacks on PLCS such as Flame, Guass, Duqu, wiper and black energy [5] which have been reported. Owing to the nature of processes controlled by SCADA systems, such as nuclear reactors in electricity and pipelines in oil and gas, a cyber-attack on them can not only lead to loss of resources but also negatively impact on human health, safely and environment. In fact, these compromises will almost likely be fatal. The operational technologies such as SCADA, which are used to manage the national Critical Infrastructure are at the center of economic wellbeing of a nation and therefore needs to be secured from cyber-attacks. This has never been more urgent than in this age of terrorism, industrial espionage and international geo-political hostilities that are prevalent today. In the events that such attacks occur, it should be possible to re-construct what took place before, during and after the attacks. It therefore calls for a need to have technologies, toolkits and mechanism in place to protect such infrastructure. The challenge however is that while a lot of research has been done in IT and a plethora of cyber security tools and technologies have been developed to deal with vulnerabilities and other cybersecurity issues in the IT domain, not a similar effort has been seen in the domain of Operational Technology. This is mainly because, as mentioned earlier, OT networks have never been interconnected with the business networks and also of the fact that for a long time OT has never been built on top of TCP/IP protocol suite. Incidentally, the most common OT communication protocols such as Modbus [7] are known to be inherently vulnerable to commons cyber-attacks since they send data in plaintext. This paper seeks to bridge this gap between IT and OT cybersecurity practices with specific reference to SCADA systems. We review SCADA systems configuration, vulnerabilities, and attacks scenarios then present a prototype SCADA system and forensic tool that can be used on SCADA. The tool reads into the PLC memory and Wireshark has been to capture network communication between the SCADA system and the PLC. The rest of this document is organized as follows. In section two, we review vulnerabilities in SCADA system and SCADA forensics as well as related research works then in section three, we present the prototype SCADA system and the toolkit we have used to demonstrate SCADA forensics. In section four we present and discuss results from the toolkit. Section five presents our conclusion and recommendations for further research works.

2. Related information and research works

2.1 Vulnerabilities and threats in SCADA Systems

Most of the modern SCADA systems have moved from the isolated case, to more interoperable and open standards for cost efficiency and integration into management IT systems. Communication is now common over Ethernet TCP/IP including more standardized control protocols and applications. Thus, SCADA systems are now susceptible to external attacks and IT based vulnerabilities [8]. The major attack vectors in SCADA control systems are through; Backdoors and holes in the network perimeter, especially in the configuration of “Air Gaps” or links to corporate enterprise IT infrastructure. Further, the operating system of PLCs, often known as a firmware lacks security features such as certificates and cryptography or intrusion detection through expected response found in normal IT operating systems [9]. If the Operating System (firmware) is compromised by a hacker, all the devices controlled by PLCs, can be completely taken over; opening the door to varieties of malicious attacks and threats. Another point of vulnerability in the SCADA system is the Human Machine Interface (HMI). This can be an application software or a user interface terminal which includes the Display Terminal Unit (DTU) which is used by the operator to control and monitor the system and the Historian Terminal Unit (HTU) which is a logging device. These two units are based on servers or PC units and run on the operating system of the PC its installed in. Like any other computers in the IT domain, they are vulnerable to threats within the network and inherit all the vulnerabilities of the underlying Operating System. For instance, HMIs have become generic or off-shelf software products that are built on or shared common architecture computers or IT systems like Windows OS, ActiveX, Java, [5]. Attacking any HMI, including its related database, could lead to severe consequences on software (including but not limited to deleting or manipulating codes, alarms, or database records) as well as on hardware. Attacks on software typically take advantages of unsecured networks or infected devices to create software manipulation or to steal confidential information [5,10].

2.2 Forensic artefacts from a SCADA system

An artefact is a piece of data that may be relevant to the investigation. This artefact can be obtained from various levels of a SCADA system as depicted on figure 1. These artefacts may include:

a. PLC application program and settings
b. Network traffic
c. Engineering workstation memory and program
d. Field devices such as process logs, date and time
e. HMI Logs and settings

Modification of a PLC memory can be recorded as a digital forensic artefact, which can be used to reconstruct a timeline of events [11]. Encase scripts have been used to carry out network acquisition to test the effects of a forensic tool on a SCADA system. The result showed that the test on network acquisition have minimal effects on the normal operation of a SCADA system. Memory addresses in a PLC are rarely modified. If a change happens then this can be recorded as a change in the programming code or logic. A capture of a PLC memory addresses would form a useful forensic artefact, that will assist to capture volatile data for digital forensic on a SCADA system.

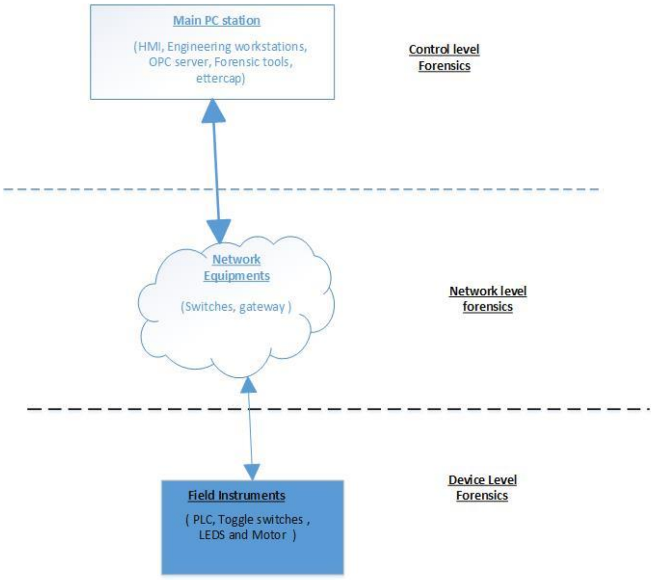


Figure 1: Levels of forensics to be used on the test bed

2.3 Related research works in SCADA Security and forensics

2.3.1 Digital Forensic Analysis of Industrial Control Systems Using Sandboxing: A Case of WAMPAC Applications in the Power Systems

The use of digital forensics in the domain of operational technology is quite limited, most of the research focuses on SCADA systems, the power of connectivity and control of power systems especially the substations makes them more challenging to protect this critical control systems. Asif Iqbal and his colleagues suggested the use of Sandboxing to test and analyze Malware in an OT environment [12], they implemented and improvised a forensic test bed by implementing a sandboxing technique in the context of a real time hardware in the loop setup. this is done by running a suspicious code without impacting the host, in OT systems, the stability of the system is key to the normal running of the controlled environment. The authors were able to map different attacks in a WAMPAC application, by designing a real time hardware forensic testbed to be used to conduct investigations on power systems especially on substations, the authors were able to carry out analysis on the power substation testbed, by analyzing the attacks from a worm. the study however could not be used to capture the PLC memory which holds key forensic artefacts especially after an attack.

2.3.2 SCADA Systems: Challenges for Forensic Investigators, when Security incidents occur in SCADA systems

Several challenges exist while conducting forensic investigation in systems which are required to be running full time to control and monitor industrial and infrastructure processes such as pipeline operations, power generations, fuel depots , building managements systems , hospitals and other critical installations, resent attacks on a SCADA systems such as the Stuxnet and Flame malwares highlighted the need to carry out forensic investigations on a SCADA system [6]. Irfan and his colleagues [13] divided the SCADA system into 5 layers, the authors indicated that forensic analysis on SCADA system takes place on layer 0-2. This layer entails the field instruments, PLCS and RTUs and the Local HMIs. Irfan and his colleagues [13] outlined the manager challenges experienced while carrying out SCADA forensics due to its nature of operation and set up, this includes:- live forensics, the SCADA system must be continuously operational a forensic investigator cannot tun it off, this affects the capturing of volatile data, SCADA systems also has a deterministic network traffic, that is the component’s communicate with others in a predefined manner, an installed intrusion detection system can be configured to consider a certain communication pattern as normal. This may lead to rising of false alarm or lead to the firewall blocking the forensic investigators machine and SCADA components during data acquisition. Another major challenge for a SCADA investigator is the customized SCADA components operating system kernels, resource constrained devices, such as CPU, Memory and input and output devices. Field devices such as PLCS / RTUS are generally resource constrained. In adequate logging of the SCADA systems which are geared towards process disturbances not security breaches.

2.3.3 Forensic Analysis of SCADA/ICS System with Security and Vulnerability Assessment

SCADA systems are used in industrial control systems (ICS) to remotely collect real time data to control and automate networked equipment’s. Due to this interconnection, SCADA systems do experience serious security experiences and breaches such as intrusion attacks which may not only lead to financial and data losses but serious environmental and health issues. Proper forensic investigations right after the incident is required to prevent loss of forensic evidence [14] Umit and his colleagues [14] designed a lab for the study of a SCADA system to be used to analyze SCADA/ICS for vulnerabilities , testing and exploiting the system weaknesses using penetration tools and analyzing the system for forensic artefacts. And analysis of this artefacts although the authors were able to develop a SCADA testbed, and carry out forensic analysis they lacked a forensic tool to collect this artefact from the testbed. This is a key requirement and an important step for collection of forensic artefacts from a SCADA system

3. SCADA forensic toolkit implementation

In this section we describe the testbed implementation including the testbed and the forensic toolkit.

3.1 Testbed Implementation

We set out to implement a toolkit for conducting digital forensics on SCADA systems. To achieve this, we set up a testbed to collect digital artifacts from a SCADA system, based on a prototype of a pipeline transporting petroleum product from one pump station to the next depot. The testbed is shown in Figure 2.

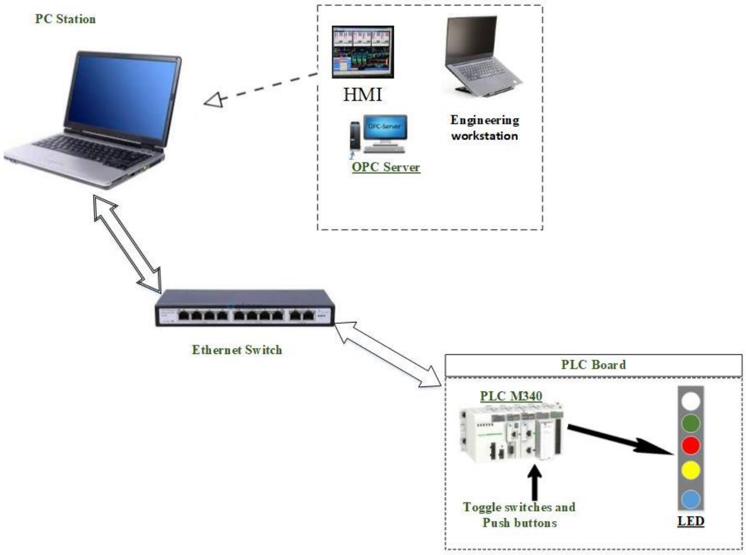


Figure 2: prototype set up

The testbed was realized using the following components:

i. PLC: We used Modicon M340 PLC consisting of a CPU, input and output modules, a power supply module, a communication module and a rack. The PLC was programmed using Ecostruxure control expert programming software.

ii. Input module: BMX DDI 1602 input modules connected to toggle switches and push buttons were used to represent field instruments

iii. BMXDDO1602 discrete output module connected to six LEDS and a Motor connected to the output port.

The PLC and the physical computer were connected via an Ethernet switch. Two Virtual Machines are running SCADA services, the HMI and the engineering workstation running Ecostruxure control.

The PC station also known as the Engineering work station runs the Ecostructure control expert, the Modicon PLC programming software and the Citec HMI software. The Engineering software can read and write into the PLC and also change the SCADA settings and parameters set.

The Master workstation has the following pieces of software installed:

i. Windows 10 professional operating system
ii. Ecostructure Control expert software
iii. Citet Vijeo Scada Software
iv. Ettercap
v. Autopsy
vi. Wireshark
vii. Unity Differencial software
viii. Developed Tool

All the power required by the PLC and the indicator switches is supplied by a 24v DV power supply. Wired to a 240V supply. The PLC was programmed to control a typical pipeline using ladder logic as per IEC 61131-3 using the Ecostruxure control expert, the petroleum product runs from one terminal to the next to be stored in a tank, the product is pumped by a booster pump, the pipeline is monitored with flow, density and temperature sensors to ensure the correct product at the correct safety levels are transferred. Alarms shown on the HMI are PLC communication failures, control power lost, and valve opening delay. A communication alarm is raised when communication is lost between PLC and the HMI. HMI acts as the Master and communicates with the PLC requesting for the information as per the commands issued

To program a PLC several languages according to IEC 61131-3, this language are as follows;

a. Ladder diagrams (LD)
b. Sequential function charts (SFC)
c. Structured texts (ST)
d. Instruction list (IL)
e. Function blocks diagram (FBD)

Wireshark was used to capture the communication between the PLC and the HMI, the results are as indicated in appendix 1, this was done prior to launch of attack and when the PLC is obtaining commands from the PLC. The HMI sent a command to the PLC initiating a valve open, Ettercap was used to generate MITM attacks on both the Master and slave, this attack depicted man in the middle attacks on the PLC and on the HMI, this combines with the filter modified any Modbus TCP command to change the signal sent to the Modbus slave, the forensic of a SCADA system can be split into three level as per the figure below The ladder logic was adopted as depicted on figure 5.3 a and b below was used, this was created using ecostruxure control expert and placed on the PLC using a USB port built on the plc.

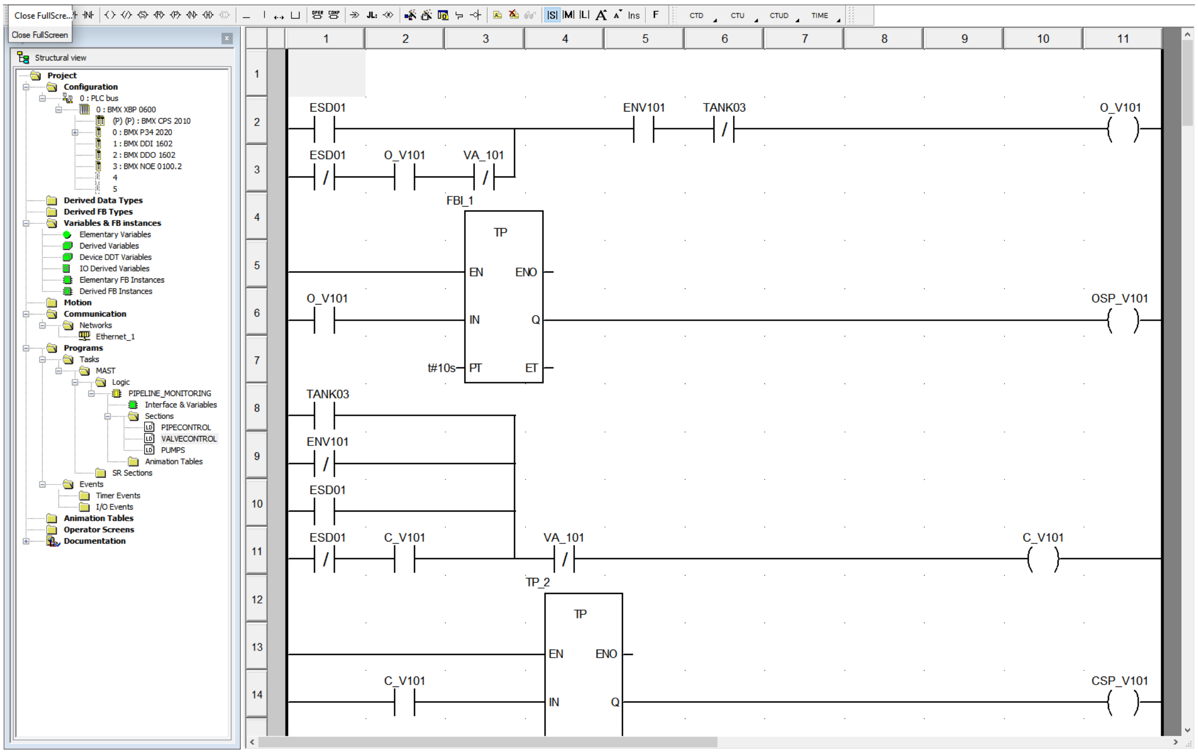


Figure 3: ladder Logic program code in ecostruxure software

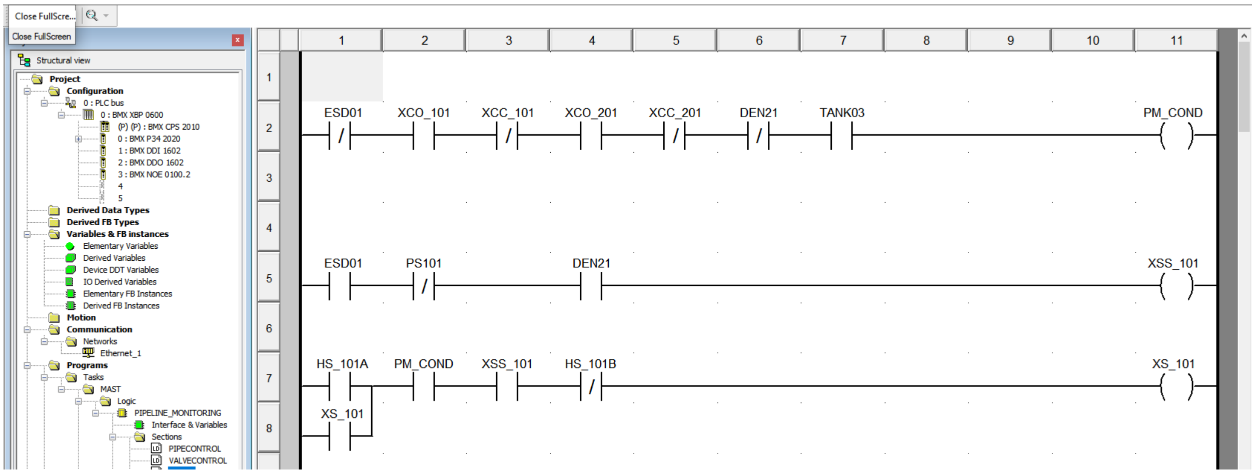


Figure 4: Ladder Logic program code in ecostruxure software

A HMI project was written on Citec SCADA and ran on the master workstation to interact with the PLC, we read and wrote into different memory . using the HMI and tool communication we captured communication and analyzed it forensically. Using the tools

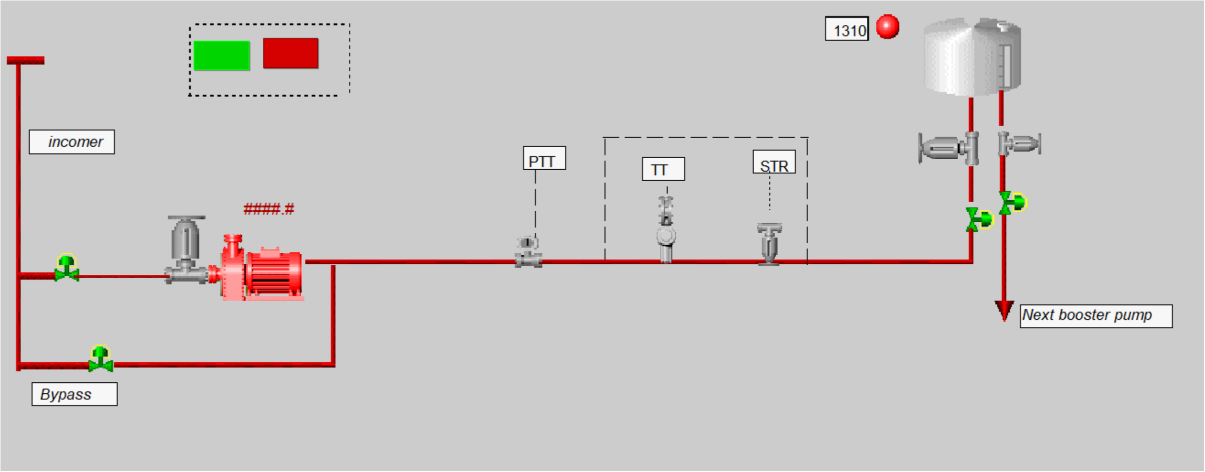


Figure 5: HMI Screen of the SCADA implementation of the prototype

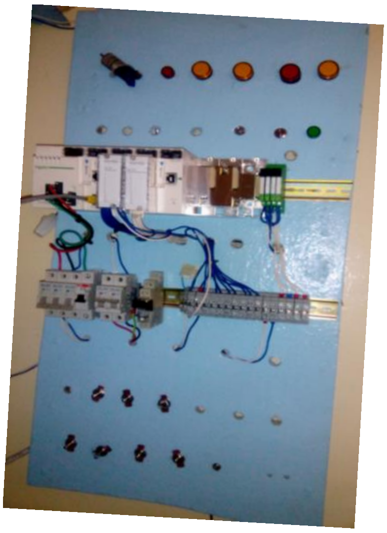


Figure 6: Test bed circuit fabrication

Unity diff software from Schneider electric was used to compare two application programs files. This is necessary when comparing ZEF, STU and XEF programs and in detecting anything added, deleted and modified. If an authorized access has taken place and program parameters edited, this program will be able to tell the difference between the two projects. This data was captured and analyzed using Hex toolkit to read into the hex data. We started by sending commands to the PLC from the HMI. The commands sent were: open-valve, close-valve and read register values.

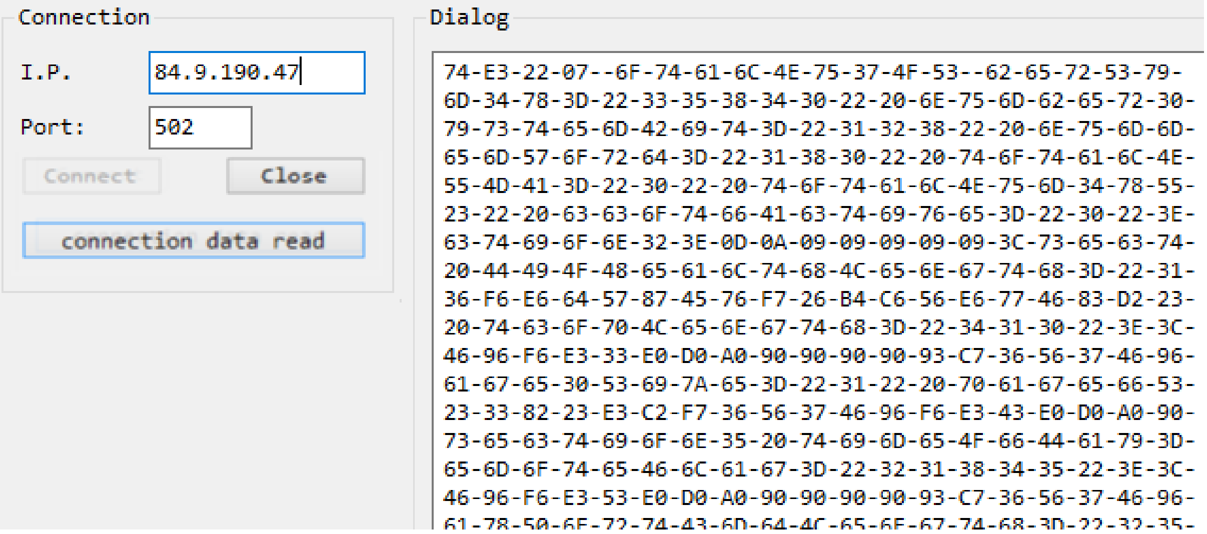
3.2 Prototype tool to read and write into the PLC

We built a tool to read the current memory addresses of the PLC to extract some artefacts from the PLC using the Modbus TCP/IP. Modbus TCP/IP communicates via port 502. Setting a PLC communication using the Easymodbus library, we can read into the PLC and capture current settings. The tool consists of 3 functional modules. Read, write and display current register setting. The application was created using C#. Once installed the IP address of the PLC must be set.

3.2.1 Application capability

The application has a capability of reading the PLC on specific registers and record current register status. To test the tool the application above was written and SCADA system run, we used the tool to read into the PLC and get current PLC settings in hexadecimals.

4. Results and Discussion



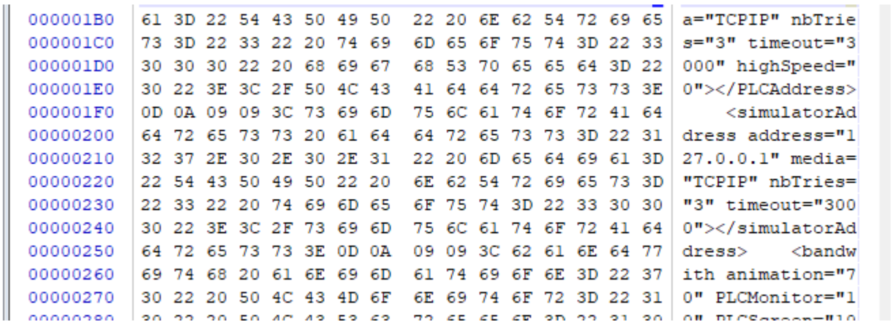
I.P. 84.9.190.47

Port: 502

```
74-E3-22-07--6F-74-61-6C-4E-75-37-4F-53--62-65-72-53-79-
6D-34-78-3D-22-33-35-38-34-30-22-20-6E-75-6D-62-65-72-30-
79-73-74-65-6D-42-69-74-3D-22-31-32-38-22-20-6E-75-6D-6D-
65-6D-57-6F-72-64-3D-22-31-38-30-22-20-74-6F-74-61-6C-4E-
55-4D-41-3D-22-30-22-20-74-6F-74-61-6C-4E-75-6D-34-78-55-
23-22-20-63-63-6F-74-66-41-63-74-69-76-65-3D-22-30-22-3E-
63-74-69-6F-6E-32-3E-0D-0A-09-09-09-09-09-3C-73-65-63-74-
20-44-49-4F-48-65-61-6C-74-68-4C-65-6E-67-74-68-3D-22-31-
36-F6-E6-64-57-87-45-76-F7-26-B4-C6-56-E6-77-46-83-D2-23-
20-74-63-6F-70-4C-65-6E-67-74-68-3D-22-34-31-30-22-3E-3C-
46-96-F6-E3-33-E0-D0-A0-90-90-90-90-93-C7-36-56-37-46-96-
61-67-65-30-53-69-7A-65-3D-22-31-22-20-70-61-67-65-66-53-
23-33-82-23-E3-C2-F7-36-56-37-46-96-F6-E3-43-E0-D0-A0-90-
73-65-63-74-69-6F-6E-35-20-74-69-6D-65-4F-66-44-61-79-3D-
65-6D-6F-74-65-46-6C-61-67-3D-22-32-31-38-34-35-22-3E-3C-
46-96-F6-E3-53-E0-D0-A0-90-90-90-90-93-C7-36-56-37-46-96-
61-78-50-6F-72-74-43-6D-64-4C-65-6E-67-74-68-3D-22-32-35-
```

Figure 7: Tool used to read PLC memory

From the engineering workstation, it was possible to obtain and analyse using a hexfile reader. STU or STA program, code. For this case WINHEX was used to read into a STU program that is generated when a programme is created in the control expert program, data below was captured

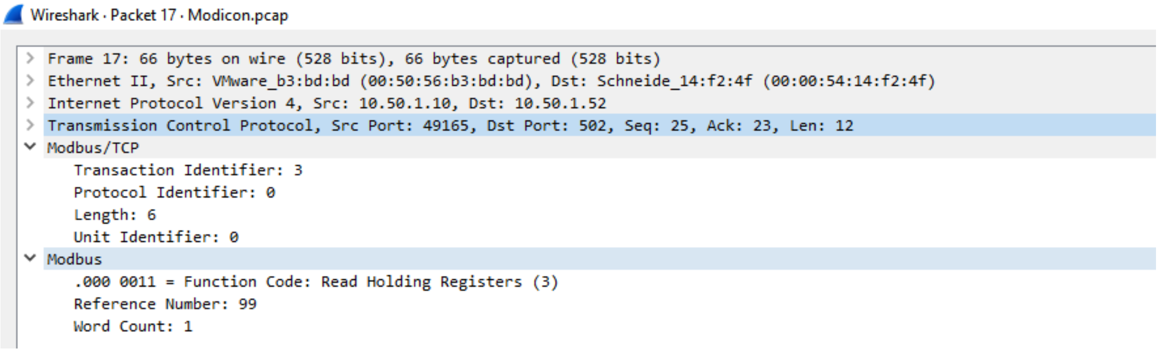


```
000001B0  61 3D 22 54 43 50 49 50  22 20 6E 62 54 72 69 65  a="TCPIP" nbTrie
000001C0  73 3D 22 33 22 20 74 69  6D 65 6F 75 74 3D 22 33  s="3" timeout="3
000001D0  30 30 30 22 20 68 69 67  68 53 70 65 65 64 3D 22  000" highSpeed="
000001E0  30 22 3E 3C 2F 50 4C 43  41 64 64 72 65 73 73 3E  0"></PLCAddress>
000001F0  0D 0A 09 09 3C 73 69 6D  75 6C 61 74 6F 72 41 64      <simulatorAd
00000200  64 72 65 73 73 20 61 64  64 72 65 73 73 3D 22 31  dress address="1
00000210  32 37 2E 30 2E 30 2E 31  22 20 6D 65 64 69 61 3D  27.0.0.1" media=
00000220  22 54 43 50 49 50 22 20  6E 62 54 72 69 65 73 3D  "TCPIP" nbTries=
00000230  22 33 22 20 74 69 6D 65  6F 75 74 3D 22 33 30 30  "3" timeout="300
00000240  30 22 3E 3C 2F 73 69 6D  75 6C 61 74 6F 72 41 64  0"></simulatorAd
00000250  64 72 65 73 73 3E 0D 0A  09 09 3C 62 61 6E 64 77  dress>    <bandw
00000260  69 74 68 20 61 6E 69 6D  61 74 69 6F 6E 3D 22 37  ith animation="7
00000270  30 22 20 50 4C 43 4D 6F  6E 69 74 6F 72 3D 22 31  0" PLCMonitor="1
00000280  30 22 20 50 4C 43 53 63  72 65 65 6E 3D 22 31 30  0" PLCScreen="10
```

Figure 8: STU Program

4.1 Analyses

Wireshark software was used to evaluate the network communication between the master and the slave, from the PCap capture below confirms that frame 17 had a unit id of 0, function code to read a register and a word count of 1.

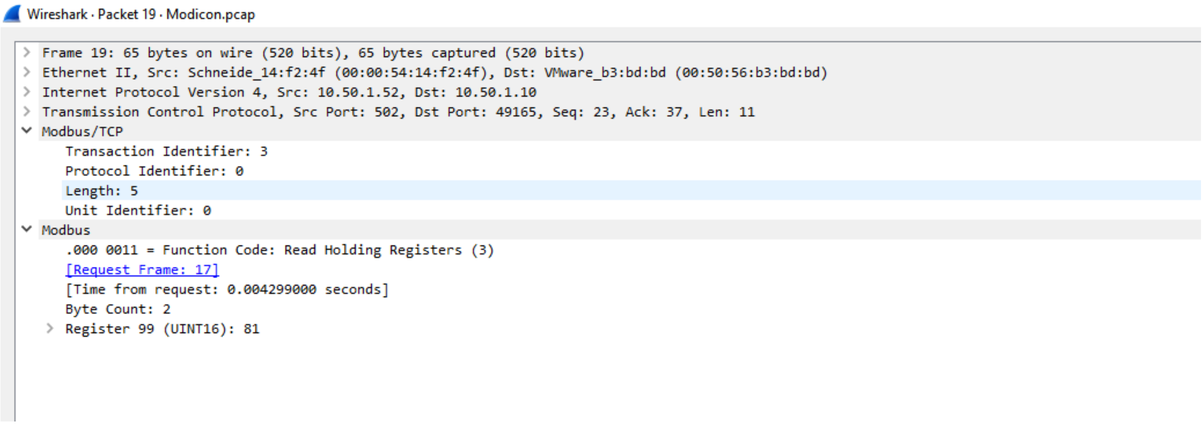


```
Wireshark · Packet 17 · Modicon.pcap

> Frame 17: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
> Ethernet II, Src: VMware_b3:bd:bd (00:50:56:b3:bd:bd), Dst: Schneide_14:f2:4f (00:00:54:14:f2:4f)
> Internet Protocol Version 4, Src: 10.50.1.10, Dst: 10.50.1.52
> Transmission Control Protocol, Src Port: 49165, Dst Port: 502, Seq: 25, Ack: 23, Len: 12
v Modbus/TCP
    Transaction Identifier: 3
    Protocol Identifier: 0
    Length: 6
    Unit Identifier: 0
v Modbus
    .000 0011 = Function Code: Read Holding Registers (3)
    Reference Number: 99
    Word Count: 1
```

Figure 9: MAster to PLC Pcap file analysis

The PLC gives a response back in frame 19 to the master to confirm register number 99 within its Modbus TCP packet as illustrated in fig

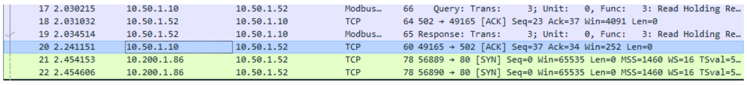


```
Wireshark · Packet 19 · Modicon.pcap

> Frame 19: 65 bytes on wire (520 bits), 65 bytes captured (520 bits)
> Ethernet II, Src: Schneide_14:f2:4f (00:00:54:14:f2:4f), Dst: VMware_b3:bd:bd (00:50:56:b3:bd:bd)
> Internet Protocol Version 4, Src: 10.50.1.52, Dst: 10.50.1.10
> Transmission Control Protocol, Src Port: 502, Dst Port: 49165, Seq: 23, Ack: 37, Len: 11
v Modbus/TCP
    Transaction Identifier: 3
    Protocol Identifier: 0
    Length: 5
    Unit Identifier: 0
v Modbus
    .000 0011 = Function Code: Read Holding Registers (3)
    [Request Frame: 17]
    [Time from request: 0.004299000 seconds]
    Byte Count: 2
  > Register 99 (UINT16): 81
```

Figure 10: PLC to Master packet analysis

Finally, in frame 20 an acknowledgement from the master workstation to the PLC was sent to convey that the Modbus TCP communication was complete. The entire communication of query response and acknowledgement is as in the figure 6.23 below.



```
17 2.030215   10.50.1.10    10.50.1.52   Modbus…  66    Query: Trans:    3; Unit:   0, Func:   3: Read Holding Re…
18 2.031032   10.50.1.52    10.50.1.10   TCP      64 502 → 49165 [ACK] Seq=23 Ack=37 Win=4091 Len=0
19 2.034514   10.50.1.52    10.50.1.10   Modbus…  65 Response: Trans:    3; Unit:   0, Func:   3: Read Holding Re…
20 2.241151   10.50.1.10    10.50.1.52   TCP      60 49165 → 502 [ACK] Seq=37 Ack=34 Win=252 Len=0
21 2.454153   10.200.1.86   10.50.1.52   TCP      78 56889 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TSval=5…
22 2.454606   10.200.1.86   10.50.1.52   TCP      78 56890 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=16 TSval=5…
```

Figure 11: query response and acknowledgement

The created forensic tool has a capability of reading a program task currently running on a PLC and the values of the all registers on the PLC. The tool is written in C# and is designed in three parts one is the GUI part the other offers communication t6o the PLC to avoid a delay or affecting the PLC cycle time, with the tool we were able to acquire the program from the PLC in binary. This code can be analyzed using a hex tool. this digital artefact can be used to analyze the program on the PLC to check for any change on the application

4.2 Performance Evaluation of the tool

This is done by evaluating the time it takes the tool to capture the hex file from the PLC. The time is calculated from the time the connect button is hit until the time the data is captured by the tool. the tool takes around 8 seconds to get the data from a plc

5. Conclusions and further research

Modern day attackers are not only focusing on IT but also on Operational technology that includes SCADA systems and DCS systems, due to the interconnection of the SCADA systems into the internet , their interconnected systems such as the PLC also forms part of the targeted system as it holds the application that collects and controls field instruments,. PLC and industrial control protocols don’t have security as part of their configuration. From our research once an attacker gains access to the SCADA network they can manipulate start and stop the PLC and change the programs therein. In this research we were able to capture a control program from a plc that may be analysed using a hex program, to connect the plc from the tool we did use the PLC IP address the Modbus TCP/IP Ethernet port 502. We were also able to capture communication between the PLC and the SCADA system and analyze it. This captured digital artefacts can be used to retrace the attack trail on this critical system. In addition to collect digital artefacts using Modbus TCP/IP the tool was tested using Modicon M340 PLC. Further research is needed to test the tool with other PLCS that support Modbus TCP/IP and with other industrial control protocols.

6. Recommendations

The tool was used to capture the program code saved in Modcon M340 PLC used as a slave by the SCADA system. The tool needs to be tested on other PLCs using Modbus TCP/IP protocol for communication. Further research needs to be done and reconfigure the tool to capture artefacts from other ICS protocols such as Profibus, DNP3 and EthernetI/P

7. Research Limitations

This research was limited to a Modicon M340 PLC using Modbus TCP/IP communication protocol.

References

[1]. Kilpatrick, T., Gonzalez, J., Chandia, R., Papa, M., Shenoi, S, “An architecture for SCADA network forensics, in Advances in Digital Forensics II,,” Springer, Boston, Massachusetts,, p. pp. 273–285 , 2006.

[2]. E. Ancillotti, R. Bruno, M. Conti, “The role of communication systems in smart grids: architectures, technical solutions and research challenges,,” Computers and Communication, vol. 36, p. 1665–1697, 2013.

[3]. Stouffer, Keith, Joe Falco, and Karen Scarfone., “Guide to industrial control systems (ICS) security,” NIST special publication, vol. 800, no. 82, pp. 16-16, 2011.

[4]. SaranyanSenthivel,IrfanAhmed,VassilRoussev, “SCADA network forensics of the PCCC protocol,” Digital investigations, vol. 22, pp. S57-S65, 2017.

[5]. Abraham Serhane, Mohamad Raad, Willy Susilo,, “Programmable logic controllers-based systems (PLC-BS): vulnerabilities and threats,,” SN Applied Sciences, vol. 1, 2019.

[6]. Nicolas Falliere, Liam O Murchu, and Eric Chien, “W3.Stuxnet DOssier v1.4,” Symantec security response, 2011.

[7]. Vinay M. Igure,Sean A. Laughter, Ronald D. Williams, “Security issues in SCADA networks.,” Computers and Security , vol. 25, p. 498–506, 2006;25(7):.

[8]. Joe Stirland, Helge Janicke,Kevin Jones, Tina Wu, “Developing Cyber Forensics for SCADA Industrial Control Systems,” in The International Conference on Information Security and Cyber Forensics , Kuala Terengganu, Malaysia, 2014.

[9]. Z. Zhang, W. Susilo and R. Raad, “Mobile ad-hoc network key management with certificateless cryptography,,” in 2nd International Conference on Signal Processing and Communication Systems, , Gold Coast,, 2008.

[10]. S. K. a. M. Wei, “SCADA Testbed for Vulnerability Assessments, Penetration Testing and Incident Forensics,,” in 7th International Symposium on Digital Forensics and Security (ISDFS),, Barcelos, Portugal, , 2019, pp. 1-6,.

[11]. Syed Ali Qasim, Juan Lopez,Irfan Ahmed, “Automated Reconstruction of Control Logic for Programmable Logic Controller Forensics,” in 22nd Information Security Conference (ISC’19), , New York, 2019.

[12]. Asif Iqbal, Farhan MAhmood, Mathias Ekstedt, “Digital Forensic Analysis of Industrial Control:Systems Using Sandboxing,” vol. 12, 2019.

[13]. Irfan Ahmed, Sebastian Obermeier, Golden G. Richard III, Martin Naedele, “SCADA Systems: Challenges for Forensic Investigators,” in Computer, vol. 45, pp. 44-51, 2012.

[14]. Umit Karabiyik,Faruk Yildiz, James Holekamp,,Khaled Rabieh, “Forensic Analysis of SCADA/ICS System with Security and Vulnerability Assessment,” in ASEE Annual Conference & Exposition, 2018.

[15]. Vinay M. Igure,Ronald D. Williams Sean A. Laughter, “Security issues in SCADA networks. Computers and Security,” 200.

[16]. Joe Stirland, Kevin Jones,Helge Janicke, Tina Wu, “Developing Cyber Forensics for SCADA Industrial Control Systems,” in The International Conference on Information Security and Cyber Forensics (InfoSec2014, 2014.