

**ANALYSIS OF SCADA SECURITY USING PENETRATION TESTING: A
CASE STUDY ON MODBUS TCP PROTOCOL**

Co-authored by
Student: John Luswata
Primary advisor: Pavol Zavarsky
Secondary advisor: Bobby Swar

Project report
Submitted to the Faculty of Graduate Studies,
Concordia University of Edmonton
In Partial Fulfillment of the
Requirements for the Final
Research Project for the Degree

**MASTER OF INFORMATION SYSTEMS SECURITY
MANAGEMENT**

Concordia University of Edmonton
FACULTY OF GRADUATE STUDIES
Edmonton, Alberta
April 2018

Analysis of SCADA Security using Penetration Testing: A case study on Modbus TCP Protocol

John Luswata, Pavol Zavorsky, Bobby Swar
Information Systems Security and Assurance Management
Concordia University of Edmonton
Edmonton, Alberta, Canada

jluswata@csa.concordia.ab.ca, {pavol.zavorsky, bobby.swar}@concordia.ab.ca

Abstract—This paper presents an insight into attacks on Supervisory Control and Data Acquisition (SCADA) systems specifically focusing on systems that use the Modbus TCP protocol. A penetration testing approach is adopted using a novel penetration testing tool to (i) test the effectiveness and efficiency of the tool, (ii) examine the insider threat as well as the external threat through internal and external penetration testing respectively and (iii) rate the vulnerabilities identified through the penetration tests according to the Common Vulnerability Scoring System. The study also examines and tests the existing security countermeasures that are unique to SCADA systems and outlines some recommendations that may improve security in SCADA systems. The experimental results show that some of the attacks may severely impact integrity and availability.

Keywords—SCADA, penetration testing, Modbus, cyber attacks, industrial Internet of Things, Common Vulnerability Scoring System, smod, fuzzing

I. INTRODUCTION

SCADA systems are specialized systems which are dedicated for monitoring and controlling of industrial infrastructures such as manufacturing plants and electricity grids. Most SCADA systems that are being used to control critical infrastructure are still largely legacy systems because they are difficult to overhaul and therefore vulnerable to several threats [1]. With the emergence of the Industrial Internet of Things (IIoT) and the inevitable growth of the Internet, the external threat to these systems has increasingly become evident and this has added a new dimension of security and privacy issues that were previously not considered when these systems were initially designed and implemented.

This study adopts a penetration testing approach not only to examine and document the flaws in Modbus TCP but also to evaluate and recommend countermeasures. In the penetration process for SCADA, similar to traditional penetration testing [2], the first step is reconnaissance. This step involves collecting exploitable information about the target, such as open ports and running services on the target. The information is then used in identifying the attack surface. The second step is the enumeration, which is the process to gather information about MAC addresses, function codes and network resources on the targeted network [2].

The penetration tests in this study are conducted using Smod penetration testing tool [2]. The functions that were utilized in our experiments include functions for testing

availability (Denial of Service, Fuzzing) and integrity (Address Resolution Protocol poisoning) aspects of information security. These functions were tested in a simulated environment and the results are documented in the following sections of this paper. Furthermore, security measures that are specific to SCADA systems were also tested through the penetration testing process to find out the effectiveness of these security measures and the results show that these countermeasures may give a false sense of security and in some cases can be misleading to security teams.

This paper is structured as follows. Section II describes the work related to this study. Section III introduces and describes the simulation environment. Sections IV illustrates, describes, and categorizes the findings and results from the experiments conducted. Finally, Section V presents the conclusions.

II. RELATED WORK

This section highlights the relationship to previous studies and outlines the contribution presented in this paper.

Authors in [3] theorized attacks based on the availability of a Modbus sniffer and a packet injector with the ability to prevent, adjust and fabricate arbitrary Modbus messages and sequences of messages. The authors describe that the main entry points for the attacks would include the Master Terminal Unit (MTU), field devices and serial communication links. Instances of various attacks were examined. For example, Direct Slave Control, an attack that involves locking out a master and controlling one or more field devices, can be used to interrupt and modify a field device, as well as to fabricate a master. The authors in [3] classify the theorized attacks into three different categories, namely (1) attacks on Modbus Serial and Modbus TCP protocols, (2) attacks only on Modbus Serial protocol, and (3) attacks only on Modbus TCP. *This paper demonstrates one of the attacks theorized in [3] by leveraging the Smod penetration testing tool to perform an ARP poisoning attack, ARP poisoning can then be used to carry out a man in the middle attack and fabricate the messages sent to the slaves from the master terminal.*

The study in [4] demonstrates the effects a Denial of Service attack on Remote Terminal Units (RTU). The experiments' purpose was to compromise an RTU communicating over IEC 60870-5-104 protocol. By performing the Denial of Service attack on IEC 60870-5-104 RTU, the authors verified that this kind of attack was indeed

possible. The authors deemed it necessary to discover a mechanism to measure the tolerance limit of an RTU against a DoS attack [4]. Denial of Service attacks target network bandwidth and resources of the target machine. A DoS attack may be executed by aiming at three different layers; network layer, transport layer and application layer. The network bandwidth can be affected by flooding random IP packets at network layer and the resource starvation can be done by either flooding SYN packets at transport layer or some user data packets at application layer. *The research in [4] demonstrates attacks on devices using the IEC 60870-5-104 protocol. In our study, DoS attacks were conducted targeting devices that use Modbus TCP protocol.*

Fuzzing is used as a software testing technique to discover any flaws or weakness in the design or implementation of a given system. The authors in [5] described a novel approach to fuzzing based on a Genetic Algorithm (GA). First, the fuzzer acquires a group of normal data from the configuration software and encodes the data to form an initial test set to start operations. The next step involves using the debugger to supervise the implementation of the test data in the configuration software and feedback the execution information such as code coverage to the GA. Finally, according to the response, GA computes the fitness value to guide the test set to perform a genetic operation to form the new generation of test sets and executes the next round of tests. This process is performed iteratively until the number of iterations is reached or the test data meeting the requirements is obtained [5]. *This study builds on that by fuzzing devices using Modbus TCP with different fuzzing functions that are available in the Smold penetration testing tool.*

In addition to firewalls, Intrusion Detection Systems (IDS) for SCADA protocols have become a common defense in traditional enterprise environments. A SCADA specific IDS has been developed in recent years. The IDS known as Quickdraw that utilizes Snort preprocessors as part of its detection engine [6] was tested by penetration testing for its efficiency and effectiveness described in the following sections of this paper.

The Common Vulnerability Scoring System (CVSS) offers a methodology to capture principal characteristics of a vulnerability and produce a numerical score reflecting its severity [7]. The numerical score can then be translated into a qualitative representation to help organizations properly prioritize their vulnerability management processes. The vulnerabilities identified in this study were scored to reflect the severity and the possible impact of the attacks if they are to occur in a SCADA environment. The Base metric group represents the intrinsic characteristics of a vulnerability that are constant over time and across user environments. The base metrics are as follows [7]:

- **Attack Vector:** This metric distinguishes between local attacks which require local system access and physical attacks which require physical access to the platform to exploit a vulnerability such as a jailbreaking attack.
- **Attack Complexity:** This metric combines two issues: any software, hardware, or networking condition beyond the attacker's control that must exist or occur for

the vulnerability to be successfully exploited and the requirement for human interaction.

- **Privileges Required:** Privileges Required captures the level of access required for a successful attack.
- **Confidentiality Impact:** This metric measures the impact to the confidentiality of the information resources managed by a software component due to a successfully exploited vulnerability. Confidentiality refers to limiting information access and disclosure to only authorized users, as well as preventing access by, or disclosure to, unauthorized ones.
- **Integrity Impact:** This metric measures the impact to integrity of a successfully exploited vulnerability. Integrity refers to the trustworthiness and veracity of information.
- **Availability Impact:** This metric measures the impact to the availability of the impacted component resulting from a successfully exploited vulnerability. While the Confidentiality and Integrity impact metrics apply to the loss of confidentiality or integrity of data (e.g., information, files) used by the impacted component, this metric refers to the loss of availability of the impacted component itself.
- **User Interaction:** This metric captures the requirement for a user, other than the attacker, to participate in the successful compromise of the vulnerable component-
- **Scope:** Scope refers to the collection of privileges defined by a computing authority when granting access to computing resources [7].

In summary, the studies described above give an insight into possible attacks and methods that can be used in SCADA environments though none of the studies seek to categorize and rate the vulnerabilities. The approach in this study is to examine the effects of attacks on SCADA systems as well as rate the possible impact of the vulnerabilities in case of a successful attack.

III. SIMULATION ENVIRONMENT

In this section, a description of the test environment is provided. A virtual environment is designed and created to simulate the system architecture and its components.

A. Test bed components

1) *Master Terminal:* The master terminal is fundamentally the supervisory computer system. The master unit serves as the SCADA system's command center. The unit provides an interface and can be set to automatically regulate the system based on information from various sensors. The interface is often used to execute tasks like data collection, and sending out commands [8]. In this study the master terminal is simulated in a virtual environment using Qmod master simulator tool and a virtual machine running on windows operating system (Windows 10).

2) *Slave Terminal:* The slave is able to collect all the necessary data, but also needs to be able to receive and

interpret data. The units send the information they receive to the master unit [9]. The two most common types of slave units used in a SCADA system are Remote Terminal Unit (RTU) and Programmable Logic Controllers (PLC). For this study the slave terminal is simulated using modbuspal simulator tool running in a virtual environment with a Linux based operating system (Ubuntu 12.04).

3) *Firewall*: A firewall is software or hardware that enforces a set of rules about what data packets will be allowed to enter or leave a network. A firewall should be able to defend the system from a variety of malicious inputs, and since the system in question is critical for everyday activities, the defense should take into consideration all forms of attacks, which can either be external or insider threats. The firewall deployed in this research is modbusfw, a Modbus specific firewall configured with a basic rule. The firewall configuration is discussed in more detail in section IV.

4) *Intrusion Detection System*: An intrusion detection system, or IDS, monitors traffic moving on networks and through systems to search for suspicious activity and known threats, sending up alerts when it finds such items. In this research a SCADA specific IDS solution (Quickdraw IDS) is deployed on the slave terminal.

5) *Attack Machine*: The attack terminal is running a Linux based operating system (Kali Linux) and utilizes Smod penetration testing tool to conduct the various penetration tests.

B. Modbus TCP Protocol

The Modbus TCP protocol is used as the reference protocol to display the effectiveness of the test bed. Modbus TCP is chosen specifically for these reasons:

- Modbus is still widely used.
- Modbus TCP is simple and easy to implement for testing purposes.

The Modbus TCP packet is 7 bytes long with 2 bytes for the header (Transaction ID), 2 bytes for the protocol identifier, 2 bytes in length and 1 byte for the address (Unit ID). There are four different data blocks defined by Modbus, and the addresses or register numbers in each of those overlap. Therefore, a complete definition of where to find a piece of data requires both the address and function code. The Modbus query consists of the Modbus application data unit (ADU) containing the unit identifier and the function code indicating the type of service requested from the Modbus server. Modbus runs on port 502.

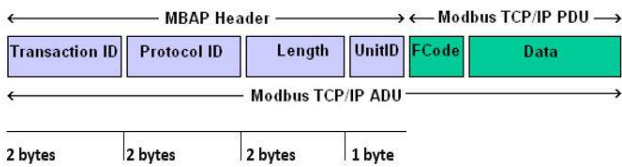


Fig 1. Modbus TCP packet

C. Testbed Architecture

The architecture of the test bed is displayed in Fig. 2. To set up this environment three simulation tools are used (i) Qmod master which acts as the master device is set up in a virtual machine running a windows operating system (Windows 10), (ii) Modbuspal which is used as the slave simulator is set up to run on a virtual machine running on a Linux platform (Ubuntu 12.04). The fundamental input fields in the master and slave are coils and registers. Each slave in a network can be assigned a unique unit address from 1 to 247 and (iii) Conpot server which is also a slave simulator is run on the slave terminal and is the primary target for the external penetration testing because it automatically creates an external IP address when it is setup which is essential for the external penetration tests. Kali Linux and Smod are installed on the attack PC. The attack machine performs the DoS, fuzzing and ARP poisoning tests. One firewall is placed between the the external network (internet) and the internal network. Another firewall is place infront of the slave terminal

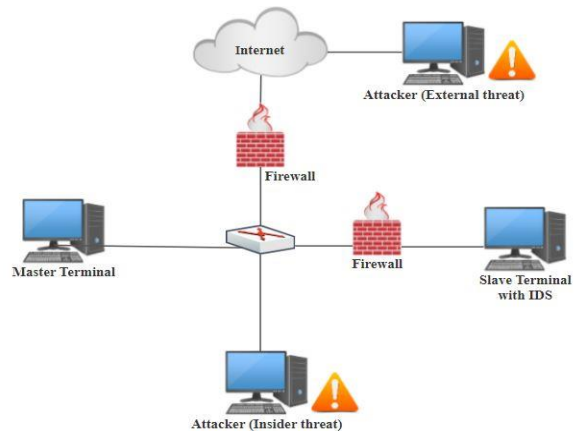


Fig 2. Experimental setup showing the system components and attack vectors.

IV. RESULTS AND FINDINGS

The study yielded substantial insight into potential vulnerabilities and the results are discussed in detail in this section.

The initial phases (reconnaissance and enumeration) discussed in the introduction were conducted by configuring the Smod tool to discover MAC addresses, IP addresses and supported function codes.

A. Internal Penetration Tests

For the initial internal penetration tests the firewall and IDS are disabled, this is meant to give a clear understanding of the full effects of an attack in an industrial environment. The base metrics for each exposed vulnerability are then summarized in a table and the base score is calculated using the Common Vulnerability Scoring System Version 3.0 Calculator [7,21] and the score is ranked according to CVSS v3.0 ratings.

TABLE I. CVSS v3.0 RATINGS [12]

Severity	Base Score
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

TABLE II. CVSS BASE METRICS (PEN-TEST I & II)

Base Metric	Metric Value
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User Interaction	None
Scope	Unchanged
Confidentiality Impact	None
Integrity Impact	Low
Availability Impact	None

1) *DoS Penetration Tests*: These tests are meant to disrupt normal operation of the slave, rendering it inoperable. DoS attacks achieve this by flooding the target with traffic, or sending it information that triggers a crash [4]. Making the system unavailable is the target for conducting DoS attacks. In all experiments, the target is the slave terminal which is communicating over Modbus TCP protocol. In this study four DoS attacks are conducted and the impact of each vulnerability identified is examined according to criteria that is outlined in the CVSS. When analyzing the impact of the attacks, we have to consider the three information security components: Confidentiality, Integrity and Availability. Analyzing each type of attack regarding these three characteristics makes it easier to identify the consequences of each attack.

a) *Pen-test I - Smod function (WriteAllCoils)*: This function floods the slave with several write coils commands. The slave device is by default setup with 64000 coil inputs. The Attack vector for this test is network based, no user privileges are required and no user interaction is needed to execute the test. Upon execution of this function we did not observe any changes in the normal functionality of the slave and master read or write functions. The master terminal was able to read all the values from the slave as well as write values successfully. Therefore there is no impact on availability since all communication is prompt and accurate. The impact on integrity is low as the values in the coils are unchanged when this test is conducted and according to the CVSS its overall impact can be categorized as medium.

b) *Pen-test II - Smod function (WriteAllRegister)*: This function floods the slave components with several write register commands. The slave device is by default setup with 64000 register inputs. The Attack vector for this test is network based, no user privileges are required and no user interaction is needed to execute the test. Upon execution of this function we did not observe any changes in the normal functionality of the slave and master read or write functions. The master terminal was able to read all the values from the slave as well as write values successfully. Therefore there is no impact on availability and the impact on integrity is low when this test is conducted thus the scope of operation is unchanged and according to the CVSS its overall impact can be categorized as medium.

Table II summarizes the results from pen-test I & II according to the criteria outlined in the CVSS using the base metric scores. The final base score is calculated using the Common Vulnerability Scoring System Version 3.0 Calculator.

Base score = 5.3 (Medium)

c) *Pen-test III - Smod function (WriteSingleCoil)*: This function targets a single coil input on the slave device by flooding it with multiple write requests. The Attack vector for this test is network based, no user privileges are required and no user interaction is needed to execute the test. When this function is executed it prevents the master from writing values to the slave, we observed that the master is still able to read coils successfully from the slave. Since integrity and availability are paramount in SCADA systems, if values can not be accurately written to the slave then the impact of such an attack is critical and the scope of operations is changed.

d) *Pen-test IV - (WriteSingleRegister)*: This function targets a single register input on the slave terminal. The attack vector is network based, no privileges or user interaction is required. When this function was executed it prevents the master terminal from writing values to the slave register field. We observed that the read registers function continues to work as the master can read values from the slave successfully. Therefore from our observations it can be concluded that the impact on integrity and availability is high and the scope of operation is changed thus this attack is classified as critical according to the criteria outlined in the CVSS.

Table III summarizes the results from pen-test III & IV according to the criteria outlined in the CVSS using the base metric scores. The final base score is calculated using the Common Vulnerability Scoring System Version 3.0 Calculator.

TABLE III. CVSS BASE METRIC (PEN-TEST III & IV)

Base Metric	Metric Value
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User Interaction	None
Scope	Changed
Confidentiality Impact	None
Integrity Impact	High
Availability Impact	High

Base score = 10.0 (Critical)

2) *ARP Poisoning Tests*: Address Resolution Protocol poisoning (ARP poisoning) refers to an attack where an adversary manipulates the Media Access Control (MAC) address, this attack can be leveraged to perform a man in the middle attack.

Smod consists of three ARP poisoning functions (i) ARP DoS, (ii) ARP watcher and (iii) ARP sniffer. The ARP DoS attacks attempts to carry out a Denial of Service. The second and third functions (ARP watcher and ARP sniffer) are similar in functionality and attempt to discover the MAC addresses of the master and slave.

When the ARP watcher and ARP sniffer are executed both yield similar results. From the results we observed that the ARP poisoning functions were able to attain the MAC address of the master terminal but were unable to attain the MAC address of the slave device. This is an indication that the tool may not be fully matured and some of the functions need to be modified before they are fully operational and effective.

B. External Penetration tests

The DoS tests conducted during the internal penetration testing phase were performed again from outside the internal network this time targeting the Conpot server on the slave terminal. From the observations made these tests did not yield any significant results and did not impact the integrity or availability of the master or slave components. Table IV summarizes the base metrics from the external penetration tests. All the tests are grouped in one table as they all yielded identical results.

TABLE IV. CVSS BASE METRICS(EXTERNAL TESTS)

Base Metric	Metric Value
Attack Vector	Network
Attack Complexity	Low
Privileges Required	Single
User Interaction	None
Scope	Unchanged

TABLE X. INTERNAL FUZZING RESULTS

Fuzzing Function	Time to first failure	Number of failures	Number of tests	Test runtime Average
Modbus/function/fuzzing	30 secs	5	10	30 mins
Modbus/function/readCoils	10 mins	2	10	30 mins
Modbus/function/readHoldingRegister	No Failure	0	15	30 mins
Modbus/function/writeSingleColil	No Failure	0	10	30 mins
Modbus/function/writeSingleRegister	No Failure	0	15	30 mins
Modbus/function/readCoilsException	No Failure	0	10	30 mins
Modbus/function/readHoldingRegisterException	No Failure	0	10	30 mins

TABLE XI. EXTERNAL FUZZING RESULTS

Fuzzing Function	Time to first failure	Number of failures	Number of tests	Test runtime Average
Modbus/function/fuzzing	29 mins	1	10	30 mins
Modbus/function/readCoils	No Failure	0	10	30 mins
Modbus/function/readHoldingRegister	No Failure	0	15	30 mins
Modbus/function/writeSingleColil	No Failure	0	10	30 mins
Modbus/function/writeSingleRegister	No Failure	0	15	30 mins
Modbus/function/readCoilsException	No Failure	0	10	30 mins
Modbus/function/readHoldingRegisterException	No Failure	0	10	30 mins

Confidentiality Impact	None
Integrity Impact	Low
Availability Impact	None

Base score = 5.3 (Medium)

C. Fuzzing tests

In this paper, a simulated slave device is fuzzed with varying results. In the test environment only seven of the twelve Smod fuzzing functions are applicable because some function codes are not supported in the test bed. The outcome of the fuzzing is shown in Tables X and XI.

If the system crashes or behaves unexpectedly, it could indicate the presence of a security flaw and further investigation is warranted [15]. If the number of failures is high and the time to first failure (TTFF) is short, the likelihood of exploitable vulnerabilities increases [16]. The average test time for the fuzzing functions is thirty minutes and the number of tests run range between 10 and 15. We found upon closer examination most of the failures involved malformed Modbus TCP packets.

In conclusion, both the fuzzing and the DoS tests indicate that these attacks can result in severe impact on system operation and stability. If an adversary can directly inject packets and control commands to SCADA components, the security and the stability of the system can be compromised. The tests conducted highlight the fact that the Modbus TCP protocol is inherently insecure. Any attacker who gets the knowledge of the Modbus Slave address can thus target it.

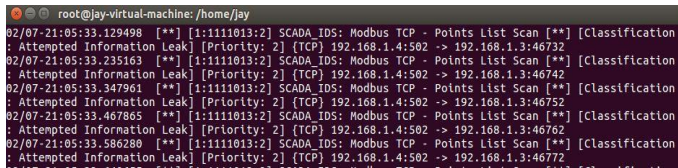
D. Defense And Mitigation Strategies

Defending against these attacks is a challenge, however; the damage is less when you know your vulnerabilities and attempt to fix or patch them. In this section we outline some attack mitigation strategies for the penetration tests that we performed. Two defense strategies are adopted (i) Intrusion Detection System and (ii) Firewall.

a) Defense I (Intrusion Detection System)

Intrusion Detection Systems (IDS) monitor and inspect traffic for any suspicious activity. In this paper an IDS designed for industrial environments was put to the test against the attack functions in Smod. The IDS was tested against five different DoS penetration test functions.

The Quickdraw IDS alerted on all the penetration tests carried out in this study although the results indicate that some of the alerts may be misleading. For instance, during a DoS attack using the write all registers function, the IDS alerted a priority 2 attack, alerting that an attempted information leakage was in occurrence, this was inaccurate and may be misleading and confusing as shown in Fig 9. This kind of information has a severe impact on attack response time as well as how accurately an attack can be dealt with.



```
root@jay-virtual-machine: /home/jay
02/07-21:05:33.129498 [**] [1:1111013:2] SCADA_IDS: Modbus TCP - Points List Scan [**] [Classification
: Attempted Information Leak] [Priority: 2] (TCP) 192.168.1.4:502 -> 192.168.1.3:46732
02/07-21:05:33.235163 [**] [1:1111013:2] SCADA_IDS: Modbus TCP - Points List Scan [**] [Classification
: Attempted Information Leak] [Priority: 2] (TCP) 192.168.1.4:502 -> 192.168.1.3:46742
02/07-21:05:33.347961 [**] [1:1111013:2] SCADA_IDS: Modbus TCP - Points List Scan [**] [Classification
: Attempted Information Leak] [Priority: 2] (TCP) 192.168.1.4:502 -> 192.168.1.3:46752
02/07-21:05:33.467865 [**] [1:1111013:2] SCADA_IDS: Modbus TCP - Points List Scan [**] [Classification
: Attempted Information Leak] [Priority: 2] (TCP) 192.168.1.4:502 -> 192.168.1.3:46762
02/07-21:05:33.586280 [**] [1:1111013:2] SCADA_IDS: Modbus TCP - Points List Scan [**] [Classification
: Attempted Information Leak] [Priority: 2] (TCP) 192.168.1.4:502 -> 192.168.1.3:46772
```

Fig 3. Quickdraw IDS alert

b) Defense II (Firewall)

Modbusfw a Modbus specific firewall was configured with a basic ruleset to counter the attacks described in earlier sections of this paper. The firewall is placed between the master and slave terminals as show in Fig 2. The ruleset is as follows;

- Rule 1; Drop all write single register requests
- Rule 2; Drop all write single coil requests
- Rule 3; Drop all read coil requests
- Rule 4; Allow only master terminal to read register

The results from the firewall tests indicated that all the rules performed as expected and if configured correctly the modbusfw firewall can be an effective countermeasure against possible attacks in SCADA environments that use the Modbus TCP protocol.

Security practices and standards such as ISA/IEC 62443, NISTIR 7628 Revision 1 (Guidelines for Smart Grid Cybersecurity) and NIST Special Publication 800-53 (Security and Privacy Controls for Federal Information Systems and Organizations) suggest that firewalls should be deployed between any two networks of differing security requirements [13]. SCADA systems may be negatively impacted by latency introduced by firewalls, thus any implemented firewall in the network should attempt to minimize the latency it introduces [13].

The V-Model for computer security risk management process [22] defines the relationship between lifecycle phases of requirements specification, design, integration, installation, and operation, and how verification and validation activities relate to development activities. This model can be incorporated into SCADA development to guarantee that security is considered at every level of the system development lifecycle [20].

V. CONCLUSION

In this paper a penetration testing approach was presented as a means of quantifying possible attacks on SCADA systems that communicate using the Modbus TCP protocol. The study used a novel penetration testing tool to examine some of the possible attacks on SCADA. Furthermore, the study goes on to use the CVSS to score and rate some of the vulnerabilities. From the observations made some of the attacks have a high impact on availability and integrity and are therefore classified as critical. Finally, the study examines some of the defenses that are specific to SCADA systems and the results indicate that though the defenses serve their purpose more can be done to improve the accuracy and functionality of these defense tools, for instance the Quickdraw IDS alerted on all the tests that were carried out but some of the alert statements did not correspond to the type of the attack in occurrence. Future work involves extending the test bed to include more SCADA components as well as incorporating SCADA hardware such as PLC components. Testing proprietary SCADA security appliances such as the Tofino security appliance against the Smod tool is also in consideration to gauge its effectiveness.

REFERENCES

- [1] C. Sandberg and B. Hunter, "Cyber security primer for legacy process plant operation," 2017 Petroleum and Chemical Industry Technical Conference (PCIC), Calgary, AB, 2017, pp. 97-102
- [2] SCADA Penetration Testing: Do I need to be prepared [online] Available: <http://research.aurainfocsec.io/scada-penetration-testing/>
- [3] P Huitsing, R Chandia, M Papa, ujeet Shenoï "Attack taxonomies for the Modbus protocols" International Journal of Critical Infrastructure Protection 1(1):37-44 · December 2008
- [4] R. Kalluri, L. Mahendra, R. K. S. Kumar and G. L. G. Prasad, "Simulation and impact analysis of denial-of-service attacks on power SCADA," 2016 National Power Systems Conference (NPSC), Bhubaneswar, 2016
- [5] B. Wu, L. Yun, X. Jin, B. Liu and G. Wei, "Study on the fuzzing test method for industrial supervisory control configuration software based on genetic algorithm," 2016
- [6] D. Peterson, "Quickdraw: Generating Security Log Events for Legacy SCADA and Control System Devices," 2009 Cybersecurity Applications & Technology Conference for Homeland Security, Washington, DC, 2009, pp. 227-229.
- [7] Common Vulnerability Scoring System v3.0: Specification Document [online] Available: <https://www.first.org/cvss/specification-document>
- [8] S. Garg, V. Kumar and Z. Saquib, "A testbed for SCADA cyber security and intrusion detection," 2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC), Shanghai, 2015, pp. 1-6.
- [9] M. M. Ahmed and W. L. Soo, "Supervisory Control and Data Acquisition System (SCADA) based customized Remote Terminal Unit (RTU) for distribution automation system," 2008 IEEE 2nd International Power and Energy Conference, Johor Bahru, 2008, pp. 1655-1660

- [10] Common Vulnerability Scoring System v3.0: Specification Document [online] Available: <https://www.first.org/cvss/specification-document>
- [11] H. Yoo and T. Shon, "Grammar-based adaptive fuzzing: Evaluation on SCADA modbus protocol," 2016 IEEE International Conference on Smart Grid Communications (SmartGridComm), Sydney, NSW, 2016, pp. 557-563
- [12] Vulnerability Metrics: NVD Vulnerability Severity Ratings [online] Available: <https://nvd.nist.gov/vuln-metrics/cvss>
- [13] D. Zvabva, P. Zavorsky, S. Butakov, J. Luswata "Evaluation of Industrial Firewall Performance Issues in Automation and Control Networks."
- [14] M. Hermann, T. Pentek and B. Otto, "Design Principles for Industrie 4.0 Scenarios," 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, 2016, pp. 3928-3937
- [15] Alert (ICS-ALERT-11-186-01) Siemens SIMATIC Controllers Password Protection Vulnerability [Online] Available: <https://ics-cert.us-cert.gov/alerts/ICS-ALERT-11-186-01>
- [16] M. M. Hasan and H. T. Mouftah, "Latency-aware segmentation and trust system placement in smart grid SCADA networks," 2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), Toronto, ON, 2016, pp. 37-42.
- [17] S. Nazir, S. Patel and D. Patel, "Autonomic computing meets SCADA security," 2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), Oxford, 2017, pp. 498-502.
- [18] Modbus Application Protocol Specification [online] Available: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf
- [19] P. J. Lee, H. Guo and B. Veeravalli, "Enhancing CII firewall performance through hash based rule lookup," TENCON 2017 - 2017 IEEE Region 10 Conference, Penang, 2017, pp. 2285-2290.
- [20] Computer Security Techniques For Nuclear Facilities [online] Available: <https://www-ns.iaea.org/downloads/security/security-series-drafts/tech-guidance/nst047.pdf>
- [21] Common Vulnerability Scoring System Version 3.0 Calculator [online] Available: <https://www.first.org/cvss/calculator/3.0>
- [22] Computer Security Techniques for Nuclear Facilities [online] <https://www-ns.iaea.org/downloads/security/security-series-drafts/tech-guidance/nst047.pdf>