Defense, Mitigation and Remediation Strategies

Francisco Catarino Mendes
Department of Informatics Engineering
University of Coimbra
Coimbra, Portugal
uc2019222823@student.uc.pt

Leonardo Oliveira Pereira
Department of Informatics Engineering
University of Coimbra
Coimbra, Portugal
uc2020239125@student.uc.pt

***Abstract*—This report presents a comprehensive strategy to protect a testbed scenario, focusing on continuous vulnerability assessment, remediation, and defense mechanisms. The strategy encompasses the selection and deployment of specific tools tailored to the testbed's requirements, ensuring robust protection and ongoing vulnerability monitoring. The chosen tools are justified based on their capabilities, ease of integration, and effectiveness in detecting and mitigating potential threats. The report also proposes targeted solutions to address vulnerabilities identified in the previous assessment.**

***Index Terms*—Vulnerabilities, Mitigation, Defense, Testbed**

I. Introduction

The growing reliance on Industrial Control Systems (ICS) and Supervisory Control and Data Acquisition (SCADA) systems in critical infrastructure has made them prime targets for cyber-attacks. The first assignment provided an in-depth reconnaissance, scouting, and vulnerability analysis of a testbed scenario mimicking a real-world industrial environment. This testbed, featuring key components such as a Human-Machine Interface (HMI), Programmable Logic Controllers (PLCs), Variable Frequency Drives (VFDs), and other essential devices, was subjected to various attack simulations to identify its security weaknesses.

The findings from Assignment 1 revealed several critical vulnerabilities. The most significant among these were the potential for Man-in-the-Middle (MitM) attacks through ARP poisoning, unsecured Modbus communication protocols, and the use of insecure Telnet services. These vulnerabilities highlight the necessity for robust defense mechanisms and continuous monitoring to protect such systems from malicious actors.

In this second assignment, the aim is to develop a comprehensive strategy to safeguard the testbed environment. This includes the selection and deployment of appropriate security tools and measures to mitigate the identified vulnerabilities and protect against potential threats. Specifically, the focus will be on integrating Snort, a widely used Intrusion Detection System (IDS), to detect and prevent exploitation attempts. Additionally, we will propose solutions for the other vulnerabilities identified, ensuring a multi-layered defense strategy.

The primary objectives of this report are to:

1) Outline a strategic plan for providing protection and continuous vulnerability assessment for the testbed scenario;
2) Propose and justify solutions to address the vulnerabilities discovered in the previous assignment;
3) Document the configuration, deployment, and evaluation of Snort as a countermeasure to detect and protect against exploitation attempts.

By addressing these objectives, the effectiveness of the proposed defense mechanisms is demonstrated and the resilience of the testbed against cyber threats is ensured. This report will include detailed documentation of each step to provide a clear and comprehensive guide for securing ICS and SCADA systems.

II. Definitions

Here, the most relevant terms of this assignment are explained, in order to make available the highest possibility of understanding what is being talked about throughout the paper.

PLC - short for Programmable Logic Controller, it is a type of computer used in industrial and commercial control applications. These devices can automate a specific process, machine function, or even an entire production line. A PLC gets its data from connected sensors or input devices, goes through it, and triggers results based on pre-established parameters. Depending on the inputs and outputs, a PLC can monitor and store information such as machine productivity or operating temperature, automatically start and stop processes, generate alarms if something is malfunctioning, and others [2].

SCADA - short for Supervisory control and data acquisition, it is a system of software and hardware elements (PLCs for example) that control and monitor industrial processes by directly interacting with plant-floor machinery and viewing real-time data. These systems are really important for industrial organizations since they help to maintain efficiency, analyze data so that the best decisions are made regarding change, and communicate system issues to help mitigate downtime issues [3].

IACS - short for Industrial Automation and Control Systems, these are electromechanical and mechanical devices that perform several control, monitoring, and actuation processes on many logic devices and process-type systems. They can also track and control many processes through sensors on machines, smart devices, and software and hardware that turn sensor information into different control outputs. Finally, they improve the efficiency, quality, safety and reliability of industrial operations by automating tasks that would otherwise require manual work [4].

Modbus Protocol - an industrial protocol that was developed in 1979 to make communication possible between automation devices. Originally designed for serial cables, has been meanwhile ported to TCP/IP. It uses a master/slave relationship, where usually the "SCADA server" (or the HMI) takes the role of Master, and the PLCs take the role of slaves. The master must regularly poll the slaves to get information. However, when talking about security, this protocol is not the best, as it has cleartext communications and no authentication.

Snort - an Open Source Intrusion Prevention System (IPS). Snort IPS uses a series of rules that help define malicious network activity, uses those rules to find packets that match against them, and generates alerts for users. Snort has three primary uses, as a packet sniffer, as a packet logger — which is useful for network traffic debugging, or it can be used as a full-blown network intrusion prevention system [1].

OpenVAS - a full-featured vulnerability scanner. Its capabilities include unauthenticated and authenticated testing, various high-level and low-level internet and industrial protocols, performance tuning for large-scale scans, and a powerful internal programming language to implement any type of vulnerability test [5].

Conpot - an ICS honeypot with the goal of collecting intelligence about the motives and methods of adversaries targeting industrial control systems.

III. Discussion

In this section of the assignment, the main operations occur. To begin with, the strategy for protection and continuous vulnerability assessment is exposed, followed by the information about the solutions found to fix the vulnerabilities. Then, the configuration, deployment, and evaluation of Snort are conducted, described, and analyzed.

A. Strategy for Protection and Continuous Vulnerability Assessment

The protection of SCADA systems and ICS environments is crucial due to their role in critical infrastructure and industrial processes. A comprehensive strategy for protection and continuous vulnerability assessment ensures these systems remain secure against evolving threats. This section outlines a strategic plan incorporating various tools and techniques to safeguard the testbed scenario.

1) Protection Strategy

: **Network Segmentation and Firewalls -** The first concepts that need to be talked about are network segmentation and firewalls. Network segmentation involves dividing the network into smaller, isolated segments to limit the spread of attacks, while firewalls are used to control traffic between these segments based on predefined security rules.

For the testbed, firewalls should be deployed at strategic points in the network to create segments, rules to allow only necessary traffic between segments should be configured, and critical components, such as PLCs and HMIs, should be isolated from less secure parts of the network.

Intrusion Detection System (IDS) - Snort - once again, Snort is a powerful open-source IDS that analyzes network traffic for signs of malicious activity. It can detect various types of attacks, including those identified in Assignment 1, such as ARP spoofing and Modbus manipulation.

To protect the testbed, Snort should be installed in passive mode to monitor network traffic without impacting performance, its rules should be configured to detect specific attack patterns, including ARP spoofing and unauthorized Modbus commands, and should be deployed on a mirrored port of the network switch to ensure it captures all relevant traffic.

Snort is selected for its robust capabilities in detecting a wide range of attacks and its flexibility in rule configuration. Its open-source nature allows for customization to fit the specific needs of the testbed scenario.

Vulnerability Scanner - OpenVAS - as said before, OpenVAS is an open-source vulnerability scanner that identifies security weaknesses in systems and networks.

Regarding the testbed, regular scans of the SCADA system components should be scheduled to detect new vulnerabilities. Furthermore, OpenVAS should be configured to generate reports on detected vulnerabilities and recommended mitigation measures and finally should be integrated with a centralized logging system for continuous monitoring and historical analysis.

OpenVAS is chosen for its comprehensive vulnerability scanning features and its ability to integrate with other security tools. Its open-source nature makes it a cost-effective solution for continuous vulnerability assessment.

Modbus Security Gateway - the Modbus protocol, commonly used in ICS environments, lacks built-in security features. A Modbus Security Gateway can add encryption and authentication to Modbus communications, mitigating the risk of unauthorized access and data manipulation.

The Modbus Security Gateway should be deployed between the PLCs and other network components, should be configured to encrypt Modbus traffic and require authentication for access, and its firmware and security settings must be regularly updated to protect against new threats.

The Modbus Security Gateway is essential for securing Modbus communications, which are inherently insecure. By adding encryption and authentication, it mitigates the risk of unauthorized access and data manipulation.

Honeypot - Conpot - these are decoy systems designed to attract attackers and monitor their activities without exposing real assets to danger. Conpot, an ICS/SCADA honeypot, can simulate various industrial devices to lure attackers and gather intelligence on their methods and tools.

The Conpot should be configured to mimic the real SCADA environment, including devices like PLCs, HMIs, and other critical components, should be deployed in a segment of the network where it is likely to attract attackers without interfering with production systems, and all interactions with the honeypot must be logged and monitored in real-time.

The Conpot tool is selected due to its abilities of early detection, intelligence gathering, and presenting a reduced number of false positives.

2) Continuous Vulnerability Assessment

: **Regular Vulnerability Scans -** continuous vulnerability assessment is essential for maintaining the security of the SCADA system. Regular scans help identify new vulnerabilities and ensure timely mitigation.

In order to achieve this, weekly or monthly scans with OpenVAS should be scheduled, scan reports should be reviewed and high-priority vulnerabilities addressed immediately, and finally the scan results should be used to update firewall rules and Snort configurations.

Continuous Network Monitoring - continuous monitoring of network traffic is critical for detecting and responding to threats in real-time.

Snort should be used to monitor network traffic continuously, alerts should be put in place for suspicious activities, such as ARP spoofing and unauthorized Modbus commands, and Snort logs and alerts should be constantly reviewed to identify and investigate potential security incidents.

Incident Response Plan - an effective incident response plan ensures quick and efficient handling of security incidents, minimizing their impact on the SCADA system.

An incident response plan should be developed and documented, outlining the steps to take in case of a security breach, personnel should be trained on the incident response procedures, and tests to the incident response plan should be regularly undertaken through simulations and drills.

Honeypot Monitoring - interactions with the honeypot should be regularly monitored and analyzed to detect early signs of targeted attacks and adjust your defenses accordingly.

3) Conclusion

: The proposed strategy for protecting the testbed scenario involves a multi-layered defense approach, incorporating network segmentation, intrusion detection, continuous vulnerability assessment, and secure communication protocols. By implementing and continuously managing these measures, one can significantly enhance the security of the SCADA system and protect it against evolving cyber threats.

B. Solutions to Fix Vulnerabilities

In the previous assignment, several vulnerabilities were identified in the SCADA testbed environment. This section outlines specific solutions to address these vulnerabilities, ensuring the security and resilience of the system.

1) Man-in-the-Middle (MitM) Attacks and ARP Spoofing

: MitM attacks, particularly ARP spoofing, can be used to intercept and manipulate communication between devices on the network. In the previous assignment, these attacks were performed successfully, evidentiating the vulnerable nature of the testbed to MitM attacks.

To solve this issue, once more, the Snort IDS should be deployed and network security measures should be implemented. Starting with Snort, it can be arranged to detect ARP spoofing by monitoring ARP packets and generating alerts for suspicious activities. Also, the arpspoof preprocessor in Snort can be used to detect ARP spoofing attempts.

As for network security measures, Dynamic ARP Inspection (DAI) can be enabled on network switches to validate ARP packets and prevent spoofing, and port security can be strengthened by configuring switch ports to limit the number of devices that can connect to each port, reducing the risk of unauthorized devices.

2) Unsecured Modbus Communication

: As identified in the previous assignment, the Modbus protocol used in the testbed environment lacks encryption and authentication, making it vulnerable to interception and manipulation.

This is where the Modbus Security Gateway enters the frame once more. The Gateway can be set up to use TLS for encrypting Modbus traffic, and user authentication and access control lists (ACLs) can be configured to restrict Modbus commands to authorized users and devices.

3) Insecure Telnet Protocol

: The use of the Telnet protocol, which transmits data in plaintext, also poses a significant security risk.

Basically, the Telnet service must be disabled on all devices and replaced with Secure Shell (SSH) for remote management. SSH should be configured on all devices to ensure secure remote access and strong authentication methods should be used, such as key-based authentication and the obligation of having strong passwords.

4) Network Security Enhancements and Vulnerability Management

: This topic ends up touching on what was already discussed in A. Strategy for Protection and Continuous Vulnerability Assessment. In order to limit scouting from adversaries, tools like network segmentation and firewalls can be used, and for vulnerability management, tools like OpenVAS are of great use.

5) Conclusion

: By implementing these solutions, one can address the vulnerabilities identified in Assignment 1 and significantly enhance the security of the SCADA testbed environment. These measures will ensure robust protection against potential attacks, continuous monitoring, and timely remediation of vulnerabilities, thereby maintaining the integrity and availability of critical industrial processes.

C. Configuration and Evaluation of Snort

This section details the steps involved in configuring, deploying, and evaluating Snort as an Intrusion Detection System (IDS) to detect and protect against potential attacks in the SCADA testbed environment.

1) Configurations

: Figures 1 to 9 show all of the Snort configurations made:

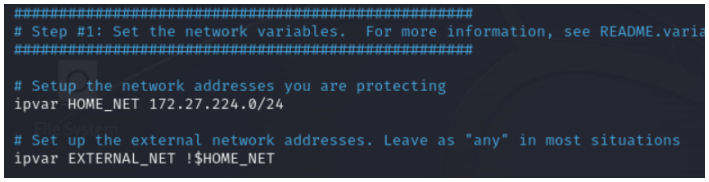

Figure 1: Setting Network Variables

The variable HOME_NET is set to the IP range of the internal network being protected. In this case, 172.27.224.0/24. As for the EXTERNAL_NET variable, it is set to any network that is not part of HOME_NET, by using the !$HOME_NET syntax.

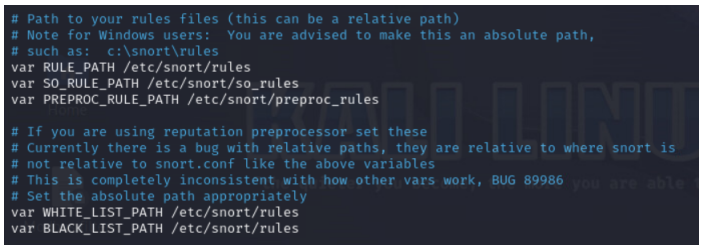

Figure 2: Path to Rule Files

RULE_PATH is the directory where Snort will look for rule files, set to /etc/snort/rules. SO_RULE_PATH is the directory for shared object rules, set to /etc/snort/so_rules. PREPROC_RULE_PATH corresponds to the directory for preprocessor rules, set to /etc/snort/preproc_rules. Finally, WHITE_LIST_PATH and BLACK_LIST_PATH are directories for whitelist and blacklist rules, both set to /etc/snort/rules.

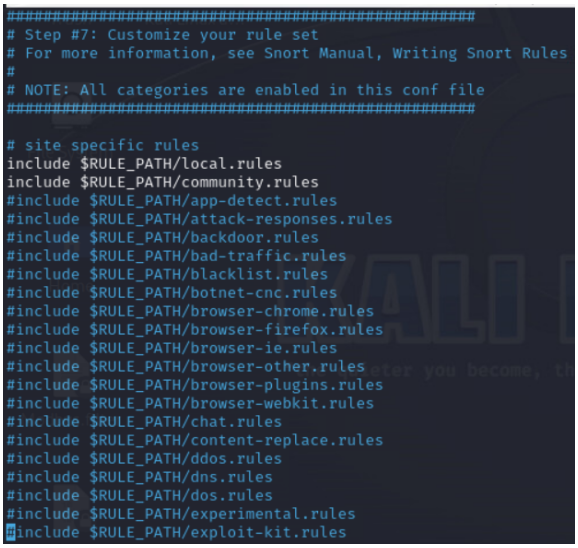

Figure 3: Customizing Rule Set

Here it is shown that both the local rules file that comes with Snort and the community rules downloaded from the Snort website were included in the rule set. All the other rules were deactivated.

```
# Configure DAQ related options for inline operation. For more
#
config daq: pcap
# config daq_dir: <dir>
config daq_mode: passive
# config daq_var: <var>
#
```

Figure 4: Configuring DAQ Options

The first command sets the Data Acquisition library to use pcap for packet capture, while the second command configures Snort to run in passive mode, meaning it will not interfere with the traffic it is monitoring.

```
# ARP spoof detection.  For more information, see the Snort Manual - Conf
preprocessor arpspoof: -unicast
preprocessor arpspoof_detect_host: 172.27.224.250 00:80:f4:09:51:3b
preprocessor arpspoof_detect_host: 172.27.224.10 00:0c:29:0e:cd:7d
```

Figure 5: ARP Spoof Detection

The first line enables the ARP spoofing detection preprocessor, while the other two specify the hosts to monitor for ARP spoofing, with their respective MAC addresses.

```
# syslog
output alert_syslog: LOG_ALERT LOG_DAEMON
```

Figure 6: Syslog Output

This command configures Snort to send alerts to the syslog using the LOG_ALERT and LOG_DAEMON facilities.

```
##################################################
# Step #8: Customize your preprocessor and decoder alerts
# For more information, see README.decoder_preproc_rules
##################################################

# decoder and preprocessor event rules
include $PREPROC_RULE_PATH/preprocessor.rules
# include $PREPROC_RULE_PATH/decoder.rules
# include $PREPROC_RULE_PATH/sensitive-data.rules
```

Figure 7: Preprocessor alerts

Here, one is just including the preprocessor rules in Snort.

The final two Figures are related to quickdraw signatures. In Figure 8, one can see the variables MODICON_CLIENT and MODBUS_CLIENT defining specific IP addresses for Modicon and Modbus clients, while the variable

```
# List of web servers on your network
ipvar HTTP_SERVERS $HOME_NET

ipvar MODICON_CLIENT [172.27.224.10/32]
ipvar MODBUS_CLIENT [172.27.224.10/32,172.27.224.251/32]
ipvar MODBUS_SERVER [172.27.224.250/32]
```

Figure 8: MODICON_CLIENT, MODBUS_CLIENT and MODBUS_SERVER variables

```
#include $RULE_PATH/web-cgi.rules
#include $RULE_PATH/web-client.rules
#include $RULE_PATH/web-coldfusion.rules
#include $RULE_PATH/web-frontpage.rules
#include $RULE_PATH/web-iis.rules
#include $RULE_PATH/web-misc.rules
#include $RULE_PATH/web-php.rules
#include $RULE_PATH/x11.rules
include $RULE_PATH/modbus.rules
include $RULE_PATH/modicon.rules

##################################################
# Step #8: Customize your preprocessor and decoder
```

Figure 9: Modbus and Modicon Rules

MODBUS_SERVER defines the IP address for the Modbus server. In Figure 9, the rules for monitoring Modbus traffic and Modicon-specific traffic are included.

All of these configurations ensure that Snort is properly set up to monitor the testbed network for various types of suspicious activities, including ARP spoofing and unauthorized Modbus communications, by setting network variables, rule paths, output settings, and including specific rule files for detecting different types of threats.

2) Tests Made *:* To begin with, an ettercap ARP poisoning attack will be performed, like it was done in the previous assignment. Figures 10 and 11 have the execution of the attack and the Snort logs, respectively:

```
┌──(root㉿kaliDEI04)-[/home/kaliucv]
└─# ettercap -T -q -i eth1 -M arp /172.27.224.250// /172.27.224.10//

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

Listening on:
  eth1 -> 00:0C:29:0F:DA:D2
          172.27.224.44/255.255.255.0
          fe80::20c:29ff:fe0f:dad2/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to EUID 65534 EGID 65534...

  34 plugins
  42 protocol dissectors
  57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Scanning for merged targets (2 hosts)...

* |==================================================>| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

 GROUP 1 : 172.27.224.250 00:80:F4:09:51:3B

 GROUP 2 : 172.27.224.10 00:0C:29:0E:CD:7D
Starting Unified sniffing...
```

Figure 10: Ettercap ARP Poisoning Attack

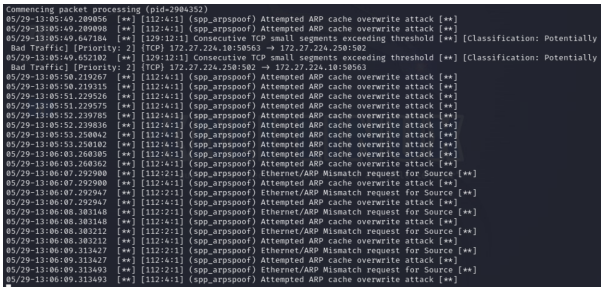

Figure 11: Snort Alerts

Although Figure 10 shows the ARP poisoning attack being executed and successfully identifying the target hosts and beginning to intercept their traffic, Figure 11 shows that Snort is detecting the ARP spoofing attempts. The alerts confirm that Snort is actively monitoring and alerting on suspicious activities, including MitM attacks.

Moving on to another attack test, this time a flood attack will be performed using the Hping3 tool. Figures 12 and 13 show the attack and the Snort logs, respectively:

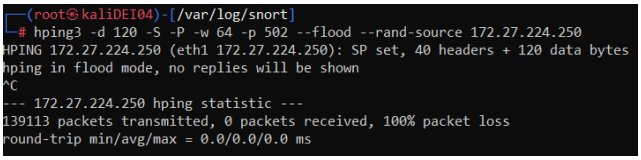

Figure 12: Hping3 Flood Attack

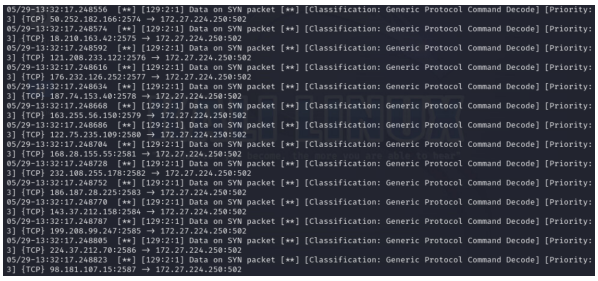

Figure 13: Snort Alerts

Figure 12 demonstrates that the hping3 tool is used to send a large number of ICMP packets to the target (testbed), creating a denial-of-service (DoS) condition by overwhelming the target with traffic. But at the same time, in Figure 13 one can see that Snort is detecting and alerting on the unusual network traffic generated by the SYN flood attack. Specifically, Snort is identifying data within SYN packets, which is atypical and flagged as potentially malicious activity. The alerts list multiple IP addresses as sources, consistent with the –rand-source option used in the hping3 command.

For the next test, the tool Nping is going to be used for another flood attack. Figures 14 and 15 show the attack and the Snort logs, respectively:

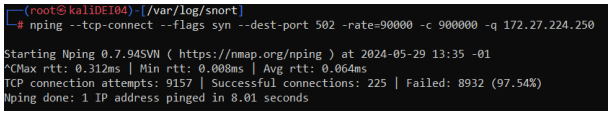

Figure 14: Nping SYN Flood Attack

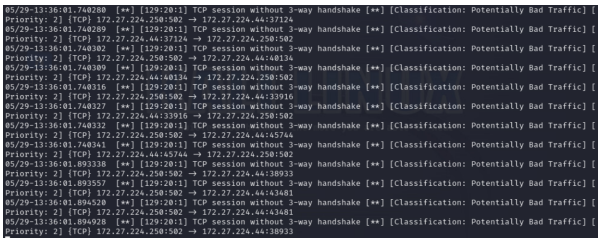

Figure 15: Snort Alerts

Figure 14 shows that the Nping tool is used to send a large number of TCP SYN packets at a very high rate to the target (testbed), overwhelming it with traffic and creating a denial-of-service (DoS) condition. The statistics show that most connection attempts failed, indicating the target is likely overwhelmed. But at the same time, Figure 15 shows that Snort is detecting and alerting the unusual network traffic generated by the SYN flood attack. Specifically, Snort is identifying TCP sessions without a complete 3-way handshake, which is characteristic of a SYN flood attack. The alerts list multiple IP addresses as sources, consistent with the high rate and volume of traffic generated by the Nping command.

Now, for another test, a Nmap Modbus Scan is going to be performed, to see if Snort also detects unauthorized scouting attempts. Figures 16 and 17 show the scouting procedure and the Snort logs, respectively:

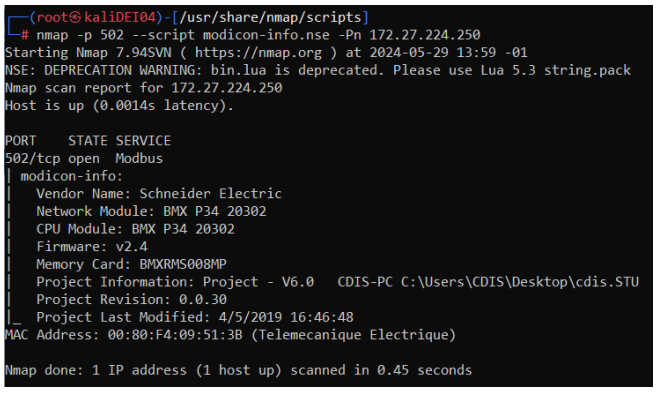

Figure 16: Nmap Modbus Scan

From Figure 16, one can verify that the modicon-info.nse script is used to gather detailed information about the Modbus device, including vendor details, hardware modules, firmware version, and project information. But at the same time, Figure 17 shows that Snort is detecting and alerting the network

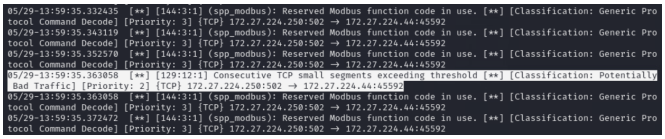

Figure 17: Snort Alerts

traffic generated by the Nmap Modbus scan. Specifically, Snort is identifying the use of reserved Modbus function codes and patterns of small TCP segments, which are flagged as potentially suspicious activities. The alerts provide detailed information about the source and destination of the detected traffic, helping in identifying and analyzing the scan activities.

Moving on to the final test, this time it is once again an Ettercap ARP Poisoning, but with a Modbus Filter. The filter is shown in Figure 18:

```
┌──(root㉿kaliDEI04)-[/home/kaliucv/Desktop]
└─# cat Modbus.filter
if (ip.proto == TCP && tcp.src == 502){
        if (DATA.data+7 == 0x03){
                msg("Found Modbus FC3 reply");
                DATA.data+22=0x27;
        }
}
```

Figure 18: Modbus Filter

Figures 19 and 20 show the attack and the Snort logs, respectively:

```
┌──(root㉿kaliDEI04)-[/home/kaliucv/Desktop]
└─# ettercap -T -q -i eth1 -M arp -F Modbus.ef /172.27.224.250// /172.27.224.10//

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

Content filters loaded from Modbus.ef...
Listening on:
  eth1 -> 00:0C:29:0F:DA:D2
          172.27.224.44/255.255.255.0
          fe80::20c:29ff:fe0f:dad2/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to EUID 65534 EGID 65534...

  34 plugins
  42 protocol dissectors
  57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Scanning for merged targets (2 hosts)...

* |==================================================>| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

 GROUP 1 : 172.27.224.250 00:80:F4:09:51:3B

 GROUP 2 : 172.27.224.10 00:0C:29:0E:CD:7D
Starting Unified sniffing...


Text only Interface activated...
Hit 'h' for inline help

Found Modbus FC3 reply
Found Modbus FC3 reply
Found Modbus FC3 reply
Found Modbus FC3 reply
```

Figure 19: Ettercap ARP Poisoning with Modbus Filter

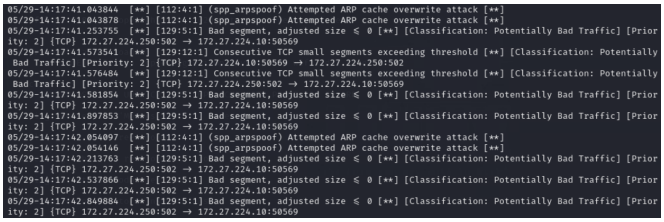

Figure 20: Snort Alerts

The Ettercap tool is used to intercept and modify traffic between these hosts using the filter, which specifically detects and modifies the Modbus Function Code 3 (FC3) replies, which are common in Modbus communication for reading holding registers, as seen in Figure 18. But on the other hand, Figure 19 illustrates that Snort is detecting and alerting the network traffic generated by the Ettercap ARP poisoning attack. Specifically, Snort is identifying ARP spoofing attempts and various anomalies in the TCP segments, such as small segment sizes and bad segments. These alerts provide detailed information about the source and destination of the detected traffic, helping in identifying and analyzing the attack activities.

3) Conclusion

: By configuring and evaluating Snort in the SCADA testbed environment, we have established an effective mechanism for detecting and responding to potential attacks. Snort's robust capabilities in traffic analysis and anomaly detection, combined with regular updates and continuous monitoring, provide a comprehensive defense against various cyber threats. This setup not only enhances the security of the SCADA system but also ensures ongoing protection through vigilant monitoring and timely response to security incidents.

IV. Conclusion

This assignment aimed to enhance the security posture of a SCADA testbed environment by implementing and evaluating a series of protective measures against identified vulnerabilities. The approach focused on deploying Snort as an Intrusion Detection System (IDS), alongside other complementary security tools and practices, to detect, monitor, and respond to various network threats.

A series of concepts were talked about regarding the strategy to provide protection and continuous vulnerability assessment for the testbed scenario and the solutions to fix the vulnerabilities found during the previous assignment, like network segmentation, firewalls, OpenVAS, Snort, Modbus Security Gateway, Honeypots (Conpots), among others. In the end, the tool chosen to be configured, deployed, and evaluated was Snort. Its configurations were presented, and a series of tests were undertaken, where one could see that Snort was indeed detecting all of the attack attempts.

This assignment has demonstrated the critical importance of a comprehensive and layered security strategy in protecting critical industrial infrastructure. Future work could be made, by configuring and testing the effectiveness of the other tools, like OpenVAS and the Conpot.

REFERENCES

[1] Snort: ”What is Snort?”, available at https://www.snort.org/.

[2] Unitronics: ”What is the definition of ”PLC”?”, available at https://www.unitronicsplc.com/what-is-plc-programmable-logic-controller/.

[3] Inductive Automation: ”SCADA: Supervisory Control and Data Acquisition”, available at https://inductiveautomation.com/resources/article/what-is-scada.

[4] MWES: ”Industrial Automation & Control Systems”, available at https://www.mwes.com/types-of-industrial-control-systems/industrial-automation-and-controls/.

[5] OpenVAS: ”Greenbone OpenVAS”, available at https://www.openvas.org/.

[6] OpenVAS: ”Greenbone OpenVAS”, available at https://www.openvas.org/.