11-2-2023

EVALUATING ATTACK SURFACE MANAGEMENT IN AN INDUSTRIAL CONTROL SYSTEM (ICS) ENVIRONMENT: LEVERAGING A RECON FTW FOR THREAT CLASSIFICATION AND INCIDENT RESPONSE

Nathalia de Sa Soares
Louisiana State University at Baton Rouge

EVALUATING ATTACK SURFACE MANAGEMENT IN AN INDUSTRIAL CONTROL SYSTEM (ICS) ENVIRONMENT: LEVERAGING A RECON FTW FOR THREAT CLASSIFICATION AND INCIDENT RESPONSE

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science

in

The Department of Computer Science

by
Nathalia de Sa Soares
B.S., University Federal Fluminense, 2017
December 2023

This thesis is dedicated to Zilma, the best mom in the world.

Know thy enemy, know thyself. A thousand battles, a thousand victories.

—Sun Tzu
The Art of War

Acknowledgments

I want to express my gratitude to those who have been instrumental in my journey toward completing my Master's in Computer Science.

First, I extend my heartfelt thanks to my advisor, Dr. Golden G. Richard III, for providing me with the incredible opportunity to pursue this program. Your guidance, mentorship, and expertise have been invaluable throughout my academic journey. Your unwavering belief in my capabilities has been a constant source of motivation.

I want to acknowledge my husband, Gabriel, whose support, patience, and understanding have supported me during this academic pursuit's demanding phases. Your belief and willingness to stand by me through thick and thin mean the world to me.

I have to give a special thank you to Devin King, who, in a short time, helped me a lot with this work, both technically and emotionally. This journey wouldn't have been as enjoyable without your unwavering support.

I am deeply grateful to my family for your enduring belief in me. Your love has been the driving force behind my ambitions. Your unwavering faith in my abilities has brought me to this point, and I am forever indebted to you.

I also extend my thanks to my professional mentor, Oliveira, for being a guiding light in my professional life. Your wisdom, advice, and willingness to share knowledge have shaped my career. Your presence has been a constant source of inspiration.

With deep gratitude,

Nathalia Soares

Table of Contents

List of Tables

List of Figures

Abstract

Protecting Industrial Control Systems (ICS) from cyber threats is paramount to ensure the reliability and security of critical infrastructure. Organizations must proactively identify vulnerabilities and strengthen their incident response capabilities as attack vectors evolve. This research explores implementing an Attack Surface Management (ASM) approach, utilizing Recon FTW, to assess an operating ICS environment's security posture comprehensively.

The primary objective of this research is to develop a tool for performing reconnaissance in an ICS environment with a non-intrusive approach, enabling the realistic simulation of potential threat scenarios and the identification of critical areas requiring immediate attention and remediation. We aim to replicate standard information-gathering techniques employed by adversaries and show the efficiency of the methods.

The research outcomes will provide valuable insights into incident response readiness and deliver an efficient, reliable, and fast tool for performing basic reconnaissance without invasive methods. The findings contribute to developing a comprehensive incident response strategy explicitly tailored for ICS environments, ultimately bolstering the resilience and security of the critical infrastructure.

Chapter 1. Introduction

An Industrial Control System (ICS) represents a complex assemblage of equipment, devices, and communication protocols designed to facilitate the delivery of specific services, manufacturing processes, or task executions [3]. Analogous to conventional Information Technology systems, the ICS domain possesses a surface susceptible to malevolent exploitation, potentially incurring financial losses, power and water outages, and potential damage to the infrastructure. In some cases, it can result in environmental issues, such as contaminated water, hazardous materials, or pollutants released into the environment, explosions causing harm to the environment and to people, along with regulatory consequences and industrial secrets, intellectual property, and sensitive information being released causing economic and security repercussions.

For instance, an illustrative case is the emergence of the ”PIPEDREAM” malware, which was scrutinized by the industrial cybersecurity firm Dragos in 2022. The actors affiliated with this cyber offensive claimed the capability to infiltrate Windows-based workstations within Operational Technology (OT) networks, subsequently wielding them for nefarious purposes [11]. This incursion disrupted mission-critical functions within liquefied natural gas (LNG) and electric power infrastructures.

Another noteworthy incident transpired in 2021 when a ransomware assault incapacitated a prominent United States pipeline, Colonial Pipeline. Although the target of this event was an ICS environment, only the IT part of the system was affected. However, even if it is not directly attacked, ICS can be affected by the IT systems. In order to mitigate the attacks on IT systems, pipeline operations needed to be shut down, resulting in a massive disruption to fuel distributions in the Gulf Coast, Eastern Coast, and Southern regions of the continental United States [12].

In 2015, a malware strain known as BlackEnergy (BE) was discovered, infiltrating the computer systems of various companies operating within Ukraine's critical infrastructure sectors. This attack had far-reaching consequences, leading to unexpected power outages within the nation. The power outages, which affected over 225,000 people, were caused by remote cyber breaches at three regional electric power distribution businesses, according to in-depth interviews with the affected companies [6]. The cyber-attack displayed a high degree of synchronization and coordination and meticulous reconnaissance of the targeted networks. Company personnel reported that these cyber-attacks on each organization occurred within a time frame of 30 minutes, simultaneously impacting numerous central and regional facilities. Throughout the course of these cyber-attacks, circuit breakers were the target of malicious activities by external actors. These attacks were enabled by configurations of the companies' Virtual Private Network (VPN), which allowed attackers remote usage of ICS software and administrative tools [6].

These instances underscore the imperative of fortifying cybersecurity measures to safeguard environments that underpin critical infrastructures and services. This infrastructure includes the fabrication of electrical power, petrochemicals, water supplies, and other needs. Comprehensive knowledge and documentation of an environment's structure are foundational to establishing both cyber-offensive and cyber-defensive postures.

The preliminary phase in the systematic acquisition of intelligence concerning a particular infrastructure is called Reconnaissance. The reconnaissance phase endeavors to unveil and chart the entirety of assets within the purview of the intended target. In the context of ICS, these assets encompass an extensive spectrum ranging from controllers and sensors to actuators, Human Machine Interfaces (HMIs), and various communication apparatuses. The reconnaissance phase can take an intrusive or a non-intrusive approach. In the intrusive approach, techniques, such as automated scans and social engineering tactics, can be executed while in the non-intrusive approach, the attacker can use publicly accessible data and open-source intelligence to serve as primary resources.

This research will utilize the open-source tool ReconFTW created by six2dez [16]. Created in 2021, ReconFTW automates the entire reconnaissance process, using techniques such as enumeration, brute force, source code, and web scraping to make the reconnaissance process easier and faster [16].

The salient contributions of this research endeavor are delineated as follows:

1. Presentation of empirical findings derived from applying ReconFTW to comprehensively map an ICS Attack Surface.
2. Design and implement a tool capable of performing Reconnaissance in a non-intrusive way tailored to the ICS environment.
3. Present the findings of the tools used in the process and rate their efficiency.

This research progresses on the critical facet of Reconnaissance within the ICS cybersecurity, engendering enhanced awareness, preparedness, and resilience in safeguarding these indispensable infrastructures.

Chapter 2. Background

2.1. ICS Architecture

To understand how an ICS works, it is important to know the basic components that exist in its architecture. A basic ICS architecture can be described as follows:

1. Sensors and Actuators: These physical devices interact with the industrial processes. Sensors gather data about temperature, pressure, flow rate, and more, while actuators control physical elements such as valves, pumps, motors, and switches. These devices are the interface between the physical world and the control system.

2. PLC (Programmable Logic Controller): PLCs are specialized computers that control industrial processes and machinery. They receive input from sensors, process this data using a set of user-defined logic, and then send commands to actuators to adjust the process accordingly. PLCs are rugged and reliable, making them well-suited for industrial environments.

3. SCADA (Supervisory Control and Data Acquisition): SCADA systems are software applications that provide a user interface for monitoring and controlling industrial processes. They collect data from PLCs and other devices, display it on operator screens, and allow operators to make manual adjustments or implement automated control strategies. SCADA systems often include features for data logging, alarming, and reporting.

4. HMI (Human-Machine Interface): HMIs are the user interfaces operators use to interact with the SCADA system. They consist of computer screens or panels that display real-time data, control options, and alarm notifications. Operators can use HMIs to monitor the system's status and make decisions based on the information presented.

5. Communication Network: A network infrastructure connects all the components in the ICS architecture, allowing them to communicate and share data. Ethernet, serial communication, and industrial field buses like Modbus or Profibus are commonly used to facilitate communication between sensors, PLCs, SCADA systems, and HMIs.

6. Server and Data Storage: In more advanced ICS architectures, there may be server systems that store historical data, perform data analysis, and support higher-level applications. These servers can help in long-term data analysis, maintenance planning, and optimizing industrial processes.

7. Firewalls and Security Measures: Modern ICS architectures incorporate security

measures, such as firewalls, intrusion detection systems, and access controls, to protect the system from unauthorized access and potential threats, given the significance of ICS systems and the potential risks associated with cyberattacks.

8. Redundancy and Fault Tolerance: To ensure the reliability of critical industrial processes, redundancy and fault-tolerant configurations are often employed. This means having backup components or systems in place to continue operation in case of failures.

Moreover, the main protocols used in the industry are described below.

1. Modbus: This was one of the first PLC-to-PLC protocols. It is widely used since the implementation does not require any license fees, and it works with various equipment, from telephones to big satellites [1]. This protocol transmits data between a primary (client) and a secondary (server) via a request-reply mechanism over port 502. A communication protocol modeled by the master-slave concept holds that a single device (primary) governs one or more other devices (secondary). In a standard Modbus network, there is one master and up to 31 slaves [21].

2. DNP3: It is another popular protocol mainly used in electrical plants and water companies. This protocol has several features such as bandwidth efficiency, time synchronization, and point-oriented objects[1]. It works on port 19999 when using transport layer security and port 20000 when not using [20]. Just like Modbus, it uses the primary/secondary principle.

3. IEC 104: Works at port 2404 and uses the primary/secondary principle. This protocol enables communication between control stations and substations via TCP/IP network [22].

Although these are the most seen protocols in the industry, it is essential to remember that several vendors have created their protocol, for example, the PROFIBUS protocol, created by Fieldbus.

A simple ICS architecture is designed to automate, monitor, and control industrial processes efficiently and safely. It is a structured framework that helps industries streamline their operations, improve productivity, and ensure the reliability and safety of their systems. As technology advances, more complex and interconnected ICS architectures may be employed to meet the specific needs of different industries.

2.2. ReconFTW

The reconnaissance phase is the first step to begin an attack or penetration test. In this phase, the attacker or the analyst will learn information about the target to perform the attacks more quickly. This phase can be made manually, using social engineering and/or phishing attacks, but it also can be performed automatically. ReconFTW is a tool built to perform reconnaissance automatically in a target domain, running a set of tools to scan and find vulnerabilities. It will scan ports, try password spraying and brute force attacks, fuzzing, and more to look into each vulnerability the target can have. The focus of this tool is to perform web scans. However, this should be something other than the focus in an ICS environment since the environment only sometimes has a web interface.

The main features of ReconFTW are:

- OSINT: In the OSINT module, it is possible to find tools such as EmailFinder and Google Dorks to aid in uncovering email addresses, user data, and hidden metadata. GitHub-specific tools like GitDorks_Go scan for sensitive information in repositories, Enumerepo, Trufflehog, and Gitleaks to help analyze and secure GitHub organizations by identifying vulnerabilities and exposed secrets.

- Subdomains: This module contains tools that offer a comprehensive array of techniques for subdomain discovery, certificate analysis, DNS reconnaissance, and cloud security assessments, like amass, subfinder, puredns, nuclei, etc.

- Hosts: In the hosts module, count with tools that can search information about the host target and related subdomains, check the ports using Nmap, perform password spray with brutespray, and not only check information about IP addresses with whois.

- Webs: The webs module looks for misconfiguration and vulnerabilities in the URLs collected. It includes javascript analysis, fuzzing, URL extraction, and web prober. Some of the tools used in this module are katana, ffuf, and nuclei.

- Vulnerability Checks: This module will look for the most common vulnerabilities in web applications, such as XSS, broken links, web cache vulnerabilities, and 4XX bypasser. Some used tools in this module are dalfox, ppfuz, and byp4xx.

Chapter 3. Related Work

One of the first things that needs to be addressed when discussing ICS and cyber security is how this type of environment deals with the CIA triad, Confidentiality, Integrity, and Availability. For IT systems, confidentiality is the most crucial factor to be secure, followed by integrity and availability. However, when looking into OT systems, the most important factor will be availability, followed by integrity and confidentiality [3]. Knowing this order can give a perspective of the ICS Cyber Kill Chain, written by Michael J. Assante and Robert M. Lee [7], where at the first stage of the chain, we have the reconnaissance phase, and as OT systems, need to be available a hundred percent of the time, they will be less updated and consequently more information online about them will be possible of being retrieved.

The Cyber Kill Chain shows two stages for ICS attacks, described as follows.

- Stage 1: Cyber Intrusion Preparation and Execution
    - Planning: Reconnaissance
    - Preparation: Weaponization and Targeting
    - Cyber Intrusion: Attempt and Success
    - Management and Enablement
    - Sustainment, Entrenchment, Development and Execution
- Stage 2: ICS Attack Development and Execution
    - Attack Development and Tuning
    - Validation
    - ICS Attack

A few works were done in the stage 1 field like the work from Paul M. Williams

[19]. In this research, the reconnaissance phase was approached using a tool called Shodan to discover detectable ICS devices online using PLC code. The findings of Paul‘s study demonstrate that PLC code may be gathered from ICS devices connected to the Internet without significantly slowing down job execution. Additionally, it illustrates a technique for classifying ICS devices with Internet access according to their functions and Critical Infrastructure sectors.

Along with the work of Mahesh Wakchaure, Satish Sarwade, and Irfan Siddavatam [18], which uses packet inspection to gather information about the system to build attack scenarios and develop methods for the defensive team to act against it. They used Deep Packet Inspection to review the contents of a packet and classify the network applications. This research used Wireshark to sniff generated network traffic and save this in a remote machine. After with TCPDump, they extracted the data to identify what was happening in the communication. It enabled them to identify the Modbus protocols and how they worked, making it possible to construct the scenarios for attackers and further build a tool to protect an ICS environment.

In addition to the aforementioned works, the research conducted by Olivier Cabana et al. [8] proposes a technique to detect, fingerprint, and track campaigns targeting ICS environments by leveraging a /13 darknet traffic. /13 refers to a specific range of IP addresses allocated for use in a network telescope, also known as a ”darknet.” A darknet is a network or a portion of the internet intentionally left unused, with no legitimate hosts or services operating. Instead, it is designed to collect and monitor unsolicited or potentially malicious network traffic passively. In the case of a ”/13 darknet,” it typically denotes a block of IP addresses from the IPv4 address space, specifically a ”/13” subnet. The ”/13” notation signifies a range of IP addresses covering half of the available IPv4 address space. In numerical terms, a "/13" block includes 524,288 IP addresses. Network telescopes like a "/13 darknet" are used for various purposes in cybersecurity and network monitoring. They help detect and analyze potentially malicious or anomalous traffic, such as scanning, probing, or unauthorized access attempts, providing valuable insights into network security threats and trends. Researchers and security professionals use data from dark networks to enhance their understanding of cyber threats and to develop countermeasures to protect networks and systems.

Olivier's work consisted of receiving a constant network traffic stream from a network telescope as input. This traffic is analyzed, clustered, and categorized. Then, every campaign fragment is output by the application. Further, they filtered the packets in two ways to see if they were part of a DoS attack: inspecting the TCP flags and UDP payloads using deep package inspection. Finally, they look into acceptable ports that correspond with ICS ports. This gives us a good perspective of protecting the systems against a reconnaissance cycle and what information attackers can get from this process.

Continuing in the Reconnaissance and attacks topic, the work of Benjamin Green, Marina Krotofil, and Ali Abbasi [10] exemplify what an attacker needs to perform a successful attack against an ICS environment, how to do the reconnaissance, prepare the attack and stay the most stealth possible during these phases. They carried out two realistic Man-in-the-Middle attacks to show what data an attacker would have to gather, compile, and understand in order to start manipulating the target's processes and avoiding discovery. These steps are very helpful in developing an easy and fast way to understand the ICS architecture and be proactive in the defense field.

There needs to be more research focusing on the reconnaissance aspect of Industrial Control Systems (ICS), resulting in a limited pool of tests and data in this domain. This gap can be attributed to the delicate nature of the ICS environment and the potential risks associated with leaving critical infrastructure exposed, as it could lead to catastrophic consequences during cybersecurity assessments.

Chapter 4. Methodology

4.1. Chosen Environment

ICS environments are typically big and very fragile, making it hard to execute tests without causing damage. With this in mind, to execute this research, a testbed was used, containing five virtual machines representing a controlled chemical plant environment.

The testbed chosen was created by David Formby, Milad Rad, and Raheem Beyah from Georgia Institute of Technology and Fortiphyd Logic and named GRFICS [9]. As said before, the environment has 5 Virtual Machines that can be used in Virtual Box software and can be described as follows:

- Simulation VM: Using a JSON API, ChemicalPlant simulates a chemical process reaction in a realistic manner, with simulated IO devices controlling and monitoring the reaction.
- PLC VM: Known as plc_2, this system runs a modified version of OpenPLC [4], using a known vulnerable earlier version of the libmodbus library.
- HMI: This machine, called SCADABR, is mostly an operator-machine interface made with free SCADABR software. This device gives commands to the PLC and keeps an eye on the process measurements being gathered by the PLC.
- Pfsense Firewall/Router: Known by its nickname, pfsense is a device that acts as a firewall and router between the ICS network and the DMZ.
- Engineering Workstation: Despite its name, this device functions more like a workstation than the engineering computer that the OpenPLC is programmed on.

4.2. Modbus Protocol

Modbus is a widely used communication protocol used in ICS environments. It was initially developed by Modicon (now Schneider Electric) in 1979 and has since become a standard for connecting electronic devices in various industries. The protocol is simple, robust, and versatile, making it a popular choice for communication between programmable

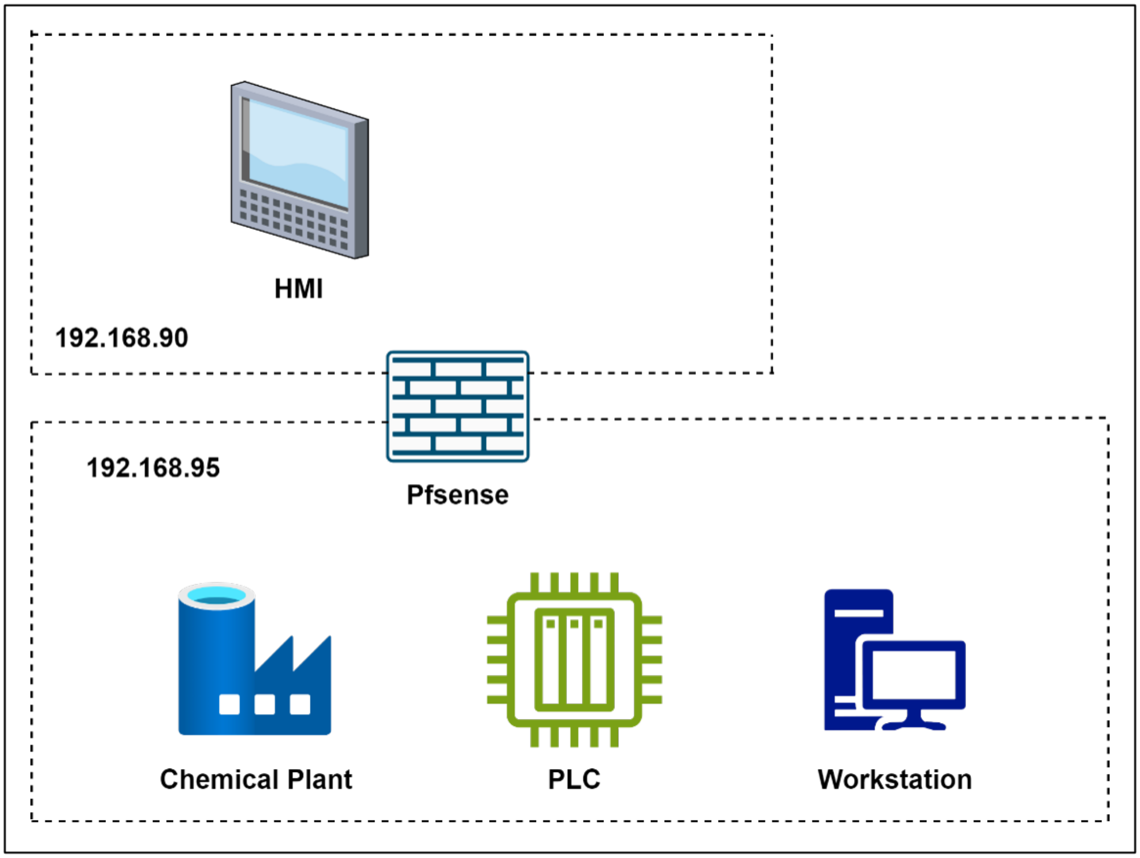


Figure 4.1. GRFICS Infrastructure

logic controllers (PLCs), sensors, and other industrial devices.

When talking about communication types, Modbus can support Serial and Ethernet communication. When working with serial communication, Modbus can use either RS-232 or RS-485 as the physical layer for serial communication. RS-232 is typically used for short-distance point-to-point connections, while RS-485 allows for multi-drop networks with longer cable runs. When working with Ethernet, TCP/IP is used for communication over Ethernet networks. It uses the standard Ethernet infrastructure and the TCP/IP protocol suite, making it suitable for local and remote communication.

As mentioned before, Modbus has a Primary/Secondary architecture where one device (the primary) initiates requests, and one or more devices (secondary) respond to these requests. The primary polls the secondary for data or sends commands. And when it comes to data type, it supports:

- Coils: Binary outputs that can be read and controlled.
- Discrete Inputs: These are binary inputs that can only be read.
- Exception Responses: Slaves can return exception responses when a request cannot be fulfilled, providing error information.

Every device in a Modbus network is identified by a unique address that ranges from 1 to 247.

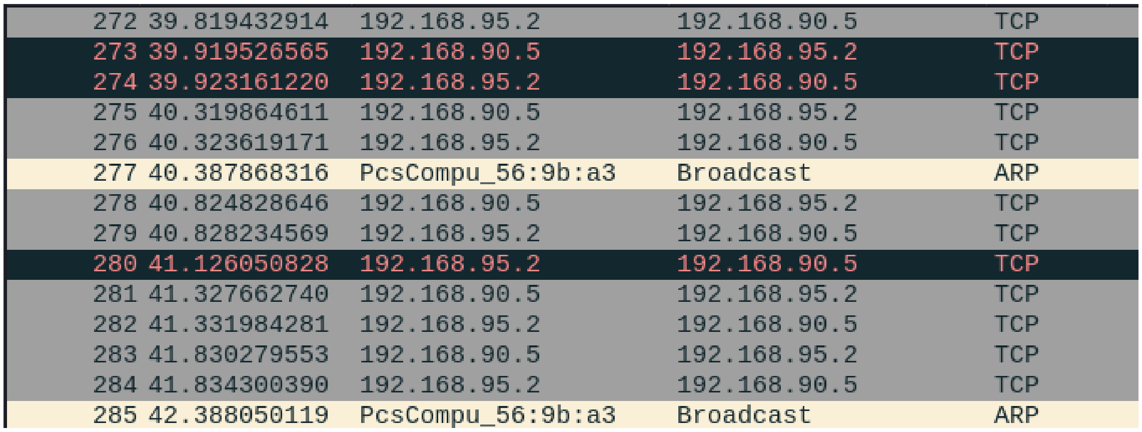
Furthermore, as security considerations, Modbus was designed without built-in security features. Therefore, when using Modbus in modern applications, it's crucial to implement additional security measures to protect against unauthorized access and data breaches.

In summary, the Modbus protocol is a widely adopted standard for industrial communication due to its simplicity and reliability. It offers versatility, making it suitable for a wide range of applications, but it's essential to consider security when implementing Modbus in today's connected industrial environments.

4.3. ReconFTW

To test how ReconFTW, without any modifications, would be able to operate in an ICS environment, some tests were done to the testbed to see the results. The first step was to set up a Kali Linux machine in the same network as the HMI and use Wireshark to intercept the communication. During the run of Wireshark, it was possible to see that only two IP addresses were communicating: the HMI address IP: 192.168.90.5 and the

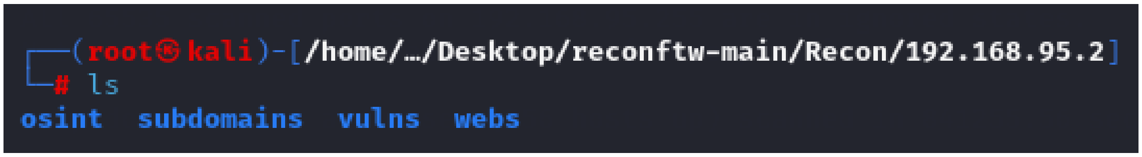
PLC IP: 192.168.95.2.



```
272 39.819432914   192.168.95.2        192.168.90.5        TCP
273 39.919526565   192.168.90.5        192.168.95.2        TCP
274 39.923161220   192.168.95.2        192.168.90.5        TCP
275 40.319864611   192.168.90.5        192.168.95.2        TCP
276 40.323619171   192.168.95.2        192.168.90.5        TCP
277 40.387868316   PcsCompu_56:9b:a3   Broadcast           ARP
278 40.824828646   192.168.90.5        192.168.95.2        TCP
279 40.828234569   192.168.95.2        192.168.90.5        TCP
280 41.126050828   192.168.95.2        192.168.90.5        TCP
281 41.327662740   192.168.90.5        192.168.95.2        TCP
282 41.331984281   192.168.95.2        192.168.90.5        TCP
283 41.830279553   192.168.90.5        192.168.95.2        TCP
284 41.834300390   192.168.95.2        192.168.90.5        TCP
285 42.388050119   PcsCompu_56:9b:a3   Broadcast           ARP
```

Figure 4.2. Communication PLC and HMI

After discovering the IP address of the PLC, ReconFTW was run, and, as expected, the results did not redirect us to any valuable information. The modules that found some results were OSINT, subdomains, vulns, and webs.



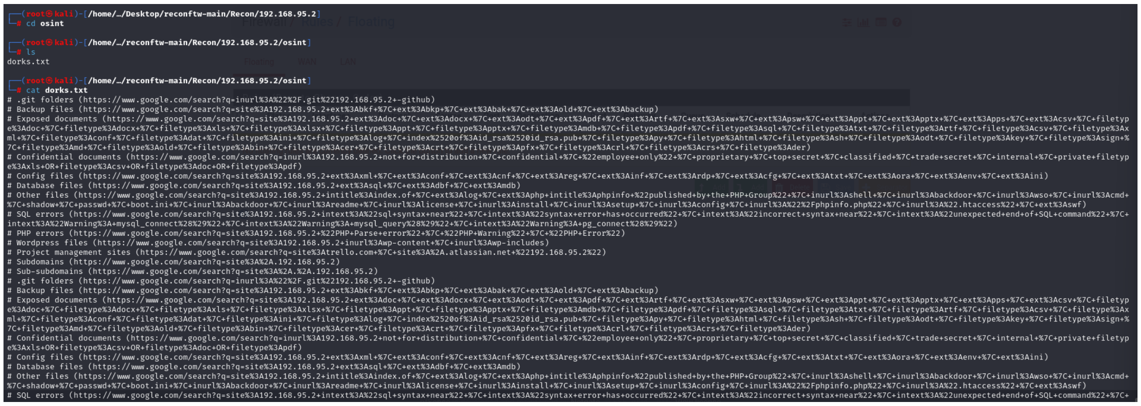
```
┌──(root㉿kali)-[/home/…/Desktop/reconftw-main/Recon/192.168.95.2]
└─# ls
osint   subdomains   vulns   webs
```

Figure 4.3. ReconFTW on PLC

Analyzing the results in the OSINT module, it is possible to see some general information about the IP. Since this IP is online because of the purpose of this testbed, the OSINT module retrieved a few information using Google Dorks. As Github information and GRFICS paper information, but at the same time, dorks that look for confidential information did not return anything about the target.

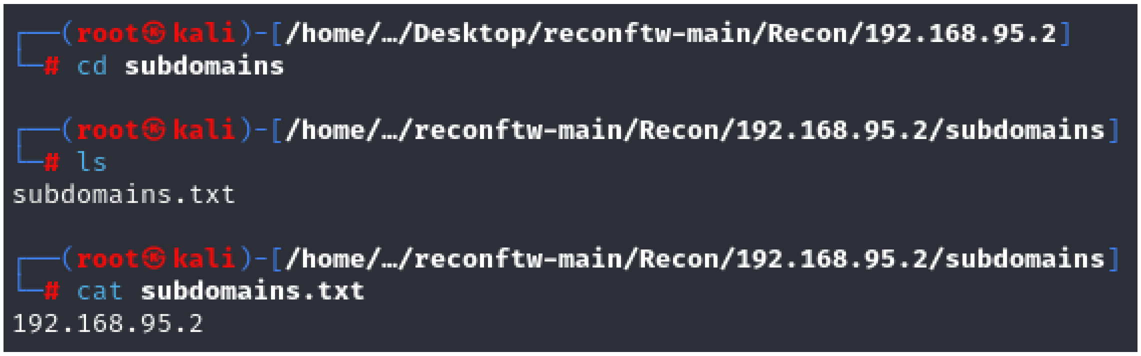
The results from the subdomains module are very straightforward since the IP has no subdomain attached to it.

Furthermore, the results from the vulns module, smuggling attacks, and SSL tests



```
┌──(root㉿kali)-[/home/…/Desktop/reconftw-main/Recon/192.168.95.2]
└─# cd osint

┌──(root㉿kali)-[/home/…/reconftw-main/Recon/192.168.95.2/osint]
└─# ls
dorks.txt

┌──(root㉿kali)-[/home/…/reconftw-main/Recon/192.168.95.2/osint]
└─# cat dorks.txt
# .git folders (https://www.google.com/search?q=inurl%3A%22%2F.git%22192.168.95.2+-github)
# Backup files (https://www.google.com/search?q=site%3A192.168.95.2+ext%3Abkf+%7C+ext%3Abkp+%7C+ext%3Abak+%7C+ext%3Aold+%7C+ext%3Abackup)
# Exposed documents (https://www.google.com/search?q=site%3A192.168.95.2+ext%3Adoc+%7C+ext%3Adocx+%7C+ext%3Aodt+%7C+ext%3Apdf+%7C+ext%3Artf+%7C+ext%3Asxw+%7C+ext%3Apsw+%7C+ext%3Appt+%7C+ext%3Apptx+%7C+ext%3Apps+%7C+ext%3Acsv+%7C+filetyp
e%3Adoc+%7C+filetype%3Adocx+%7C+filetype%3Axls+%7C+filetype%3Axlsx+%7C+filetype%3Appt+%7C+filetype%3Apptx+%7C+filetype%3Amdb+%7C+filetype%3Apdf+%7C+filetype%3Asql+%7C+filetype%3Atxt+%7C+filetype%3Artf+%7C+filetype%3Acsv+%7C+filetype%3Ax
ml+%7C+filetype%3Aconf+%7C+filetype%3Adat+%7C+filetype%3Aini+%7C+filetype%3Alog+%7C+index%2520of%3Aid_rsa%2520id_rsa.pub+%7C+filetype%3Apy+%7C+filetype%3Ahtml+%7C+filetype%3Ash+%7C+filetype%3Aodt+%7C+filetype%3Akey+%7C+filetype%3Asign+%
7C+filetype%3Amd+%7C+filetype%3Aold+%7C+filetype%3Abin+%7C+filetype%3Acer+%7C+filetype%3Acrt+%7C+filetype%3Apfx+%7C+filetype%3Acrl+%7C+filetype%3Acrs+%7C+filetype%3Ader)
# Confidential documents (https://www.google.com/search?q=inurl%3A192.168.95.2+not+for+distribution+%7C+confidential+%7C+%22employee+only%22+%7C+proprietary+%7C+top+secret+%7C+classified+%7C+trade+secret+%7C+internal+%7C+private+filetyp
e%3Axls+OR+filetype%3Acsv+OR+filetype%3Adoc+OR+filetype%3Apdf)
# Config files (https://www.google.com/search?q=site%3A192.168.95.2+ext%3Axml+%7C+ext%3Aconf+%7C+ext%3Acnf+%7C+ext%3Areg+%7C+ext%3Ainf+%7C+ext%3Ardp+%7C+ext%3Acfg+%7C+ext%3Atxt+%7C+ext%3Aora+%7C+ext%3Aenv+%7C+ext%3Aini)
# Database files (https://www.google.com/search?q=site%3A192.168.95.2+ext%3Asql+%7C+ext%3Adbf+%7C+ext%3Amdb)
# Other files (https://www.google.com/search?q=site%3A192.168.95.2+intitle%3Aindex.of+%7C+ext%3Alog+%7C+ext%3Aphp+intitle%3Aphpinfo+%22published+by+the+PHP+Group%22+%7C+inurl%3Ashell+%7C+inurl%3Abackdoor+%7C+inurl%3Awso+%7C+inurl%3Acmd+
%7C+shadow+%7C+passwd+%7C+boot.ini+%7C+inurl%3Abackdoor+%7C+inurl%3Areadme+%7C+inurl%3Alicense+%7C+inurl%3Ainstall+%7C+inurl%3Asetup+%7C+inurl%3Aconfig+%7C+inurl%3A%22%2Fphpinfo.php%22+%7C+inurl%3A%22.htaccess%22+%7C+ext%3Aswf)
# SQL errors (https://www.google.com/search?q=site%3A192.168.95.2+intext%3A%22sql+syntax+near%22+%7C+intext%3A%22syntax+error+has+occurred%22+%7C+intext%3A%22incorrect+syntax+near%22+%7C+intext%3A%22unexpected+end+of+SQL+command%22+%7C+
intext%3A%22Warning%3A+mysql_connect%28%29%22+%7C+intext%3A%22Warning%3A+mysql_query%28%29%22+%7C+intext%3A%22Warning%3A+pg_connect%28%29%22)
# PHP errors (https://www.google.com/search?q=site%3A192.168.95.2+%22PHP+Parse+error%22+%7C+%22PHP+Warning%22+%7C+%22PHP+Error%22)
# Wordpress files (https://www.google.com/search?q=site%3A192.168.95.2+inurl%3Awp-content+%7C+inurl%3Awp-includes)
# Project management sites (https://www.google.com/search?q=site%3Atrello.com+%7C+site%3A%2A.atlassian.net+%22192.168.95.2%22)
# Subdomains (https://www.google.com/search?q=site%3A%2A.192.168.95.2)
# Sub-subdomains (https://www.google.com/search?q=site%3A%2A.%2A.192.168.95.2)
# .git folders (https://www.google.com/search?q=inurl%3A%22%2F.git%22192.168.95.2+-github)
# Backup files (https://www.google.com/search?q=site%3A192.168.95.2+ext%3Abkf+%7C+ext%3Abkp+%7C+ext%3Abak+%7C+ext%3Aold+%7C+ext%3Abackup)
# Exposed documents (https://www.google.com/search?q=site%3A192.168.95.2+ext%3Adoc+%7C+ext%3Adocx+%7C+ext%3Aodt+%7C+ext%3Apdf+%7C+ext%3Artf+%7C+ext%3Asxw+%7C+ext%3Apsw+%7C+ext%3Appt+%7C+ext%3Apptx+%7C+ext%3Apps+%7C+ext%3Acsv+%7C+filetyp
e%3Adoc+%7C+filetype%3Adocx+%7C+filetype%3Axls+%7C+filetype%3Axlsx+%7C+filetype%3Appt+%7C+filetype%3Apptx+%7C+filetype%3Amdb+%7C+filetype%3Apdf+%7C+filetype%3Asql+%7C+filetype%3Atxt+%7C+filetype%3Artf+%7C+filetype%3Acsv+%7C+filetype%3Ax
ml+%7C+filetype%3Aconf+%7C+filetype%3Adat+%7C+filetype%3Aini+%7C+filetype%3Alog+%7C+index%2520of%3Aid_rsa%2520id_rsa.pub+%7C+filetype%3Apy+%7C+filetype%3Ahtml+%7C+filetype%3Ash+%7C+filetype%3Aodt+%7C+filetype%3Akey+%7C+filetype%3Asign+%
7C+filetype%3Amd+%7C+filetype%3Aold+%7C+filetype%3Abin+%7C+filetype%3Acer+%7C+filetype%3Acrt+%7C+filetype%3Apfx+%7C+filetype%3Acrl+%7C+filetype%3Acrs+%7C+filetype%3Ader)
# Confidential documents (https://www.google.com/search?q=inurl%3A192.168.95.2+not+for+distribution+%7C+confidential+%7C+%22employee+only%22+%7C+proprietary+%7C+top+secret+%7C+classified+%7C+trade+secret+%7C+internal+%7C+private+filetyp
e%3Axls+OR+filetype%3Acsv+OR+filetype%3Adoc+OR+filetype%3Apdf)
# Config files (https://www.google.com/search?q=site%3A192.168.95.2+ext%3Axml+%7C+ext%3Aconf+%7C+ext%3Acnf+%7C+ext%3Areg+%7C+ext%3Ainf+%7C+ext%3Ardp+%7C+ext%3Acfg+%7C+ext%3Atxt+%7C+ext%3Aora+%7C+ext%3Aenv+%7C+ext%3Aini)
# Database files (https://www.google.com/search?q=site%3A192.168.95.2+ext%3Asql+%7C+ext%3Adbf+%7C+ext%3Amdb)
# Other files (https://www.google.com/search?q=site%3A192.168.95.2+intitle%3Aindex.of+%7C+ext%3Alog+%7C+ext%3Aphp+intitle%3Aphpinfo+%22published+by+the+PHP+Group%22+%7C+inurl%3Ashell+%7C+inurl%3Abackdoor+%7C+inurl%3Awso+%7C+inurl%3Acmd+
%7C+shadow+%7C+passwd+%7C+boot.ini+%7C+inurl%3Abackdoor+%7C+inurl%3Areadme+%7C+inurl%3Alicense+%7C+inurl%3Ainstall+%7C+inurl%3Asetup+%7C+inurl%3Aconfig+%7C+inurl%3A%22%2Fphpinfo.php%22+%7C+inurl%3A%22.htaccess%22+%7C+ext%3Aswf)
# SQL errors (https://www.google.com/search?q=site%3A192.168.95.2+intext%3A%22sql+syntax+near%22+%7C+intext%3A%22syntax+error+has+occurred%22+%7C+intext%3A%22incorrect+syntax+near%22+%7C+intext%3A%22unexpected+end+of+SQL+command%22+%7C+
```

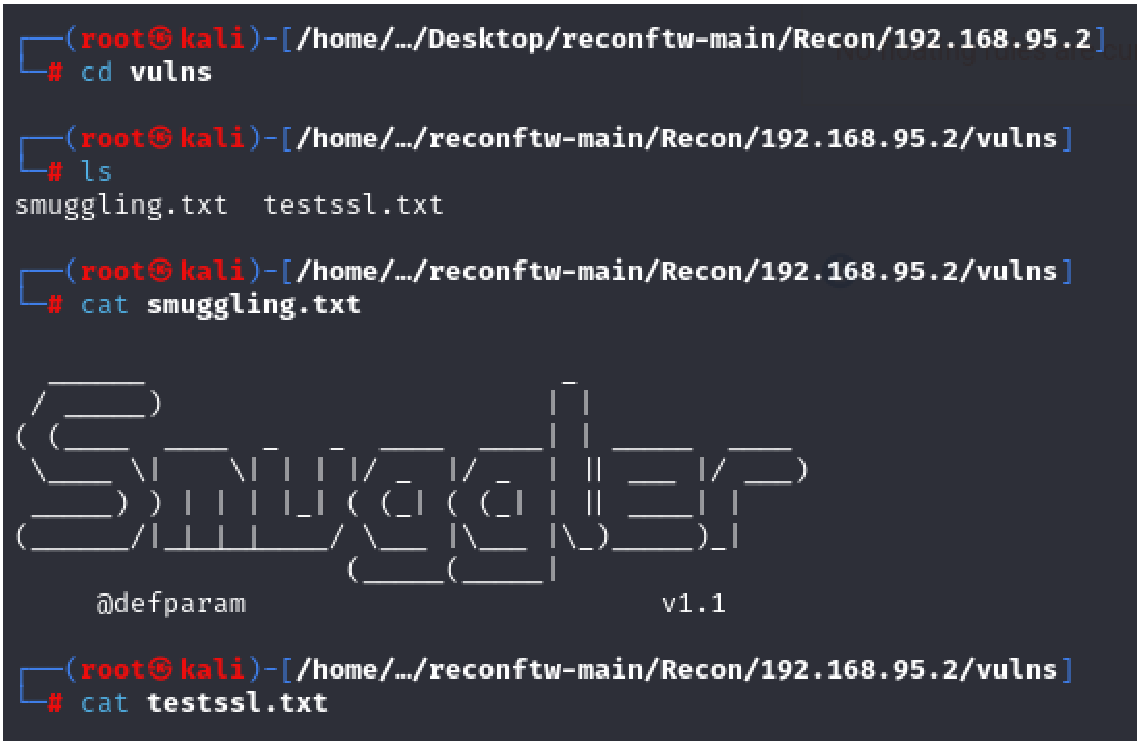
Figure 4.4. OSINT Module Results



```
┌──(root㉿kali)-[/home/…/Desktop/reconftw-main/Recon/192.168.95.2]
└─# cd subdomains

┌──(root㉿kali)-[/home/…/reconftw-main/Recon/192.168.95.2/subdomains]
└─# ls
subdomains.txt

┌──(root㉿kali)-[/home/…/reconftw-main/Recon/192.168.95.2/subdomains]
└─# cat subdomains.txt
192.168.95.2
```

Figure 4.5. Subdomains Module Results

returned zero results, too, since these tools are designed to look into web applications, which are not the type of application we have in an ICS environment.



```
┌──(root㉿kali)-[/home/…/Desktop/reconftw-main/Recon/192.168.95.2]
└─# cd vulns

┌──(root㉿kali)-[/home/…/reconftw-main/Recon/192.168.95.2/vulns]
└─# ls
smuggling.txt  testssl.txt

┌──(root㉿kali)-[/home/…/reconftw-main/Recon/192.168.95.2/vulns]
└─# cat smuggling.txt

  ______                         _
 / _____)                       | |
( (____   ____  _   _  ____  ____| | _____  ____
 \____ \ |    \| | | |/ _  |/ _  | || ___ |/ ___)
 _____) )| | | | |_| ( (_| ( (_| | || ____| |
(______/ |_|_|_|____/ \___ |\___ |\_)_____)_|
                     (_____(_____|

     @defparam                         v1.1

┌──(root㉿kali)-[/home/…/reconftw-main/Recon/192.168.95.2/vulns]
└─# cat testssl.txt
```

Figure 4.6. Vulns Module Results

In conclusion, ReconFTW showed to be a swift and easy tool to perform reconnaissance, but to work in the ICS world, it needs to have some tools specific to the ICS protocols and ports, which will be inserted and discussed in the next chapter of this work: Implementation and Results.

Chapter 5. Implementation and Results

To test and prove the results from the ReconFTW, the reconnaissance phase for this research started by implementing a Kali Linux Virtual Machine in the same network as the HMI machine. Further, a Nmap scan was done, and Wireshark was used to analyze the packets. Enabling the IP discovery referring to the PLC, the desired target. Later, running the ReconFTW, as demonstrated in chapter 4, the results were unsatisfactory.

The next step was to find tools specifically for ICS environments that could be combined into a new tool called ReconICS. This new tool proposes to reunite available open-source resources with a Command Line Interface - CLI that could be a general overview of an ICS environment. Furthermore, it helps security analysts develop defensive techniques to avoid exploiting vulnerabilities.

The chosen open-source tools were:

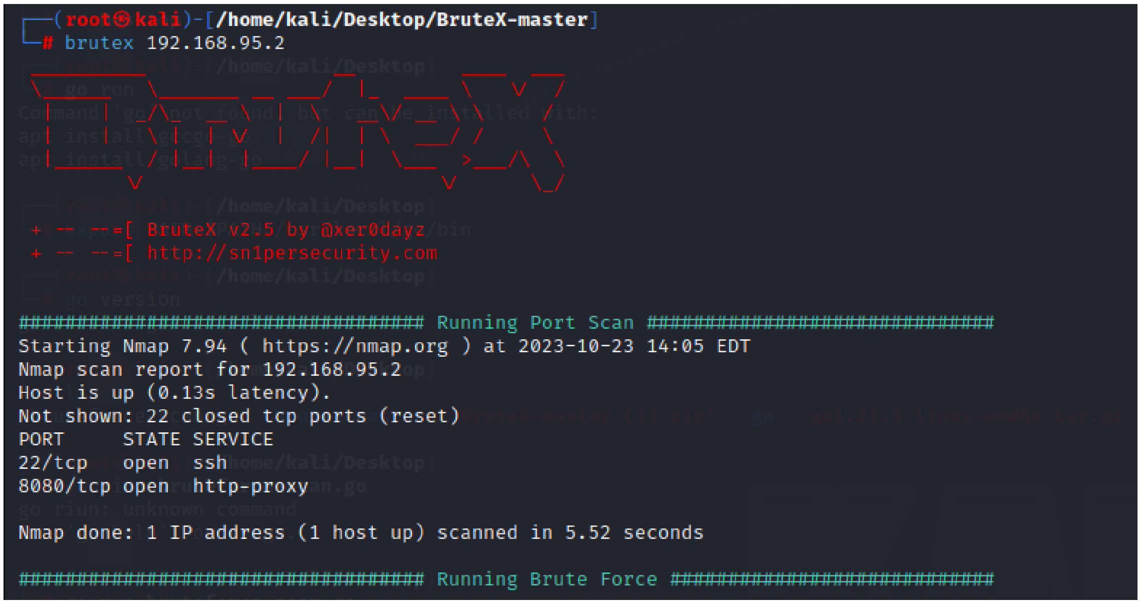
- Brute X: A brute force tool performing a Nmap scan and a brute force attack with wordlist [2].
- Icssploit: A framework similar to Metasploit to exploit ICS environments [17].
- Recon Modbus Functions: A tool for reconnaissance in ICS devices, recognizing implemented ”MODBUS” functions [5].

5.1. Brute X

By utilizing Brute X to target the PLC IP 192.168.95.2 without specifying the port, the tool identified TCP ports 22 and 8080. However, as it relies on Nmap for port scanning, it provides a limited set of results rather than conducting a comprehensive scan.

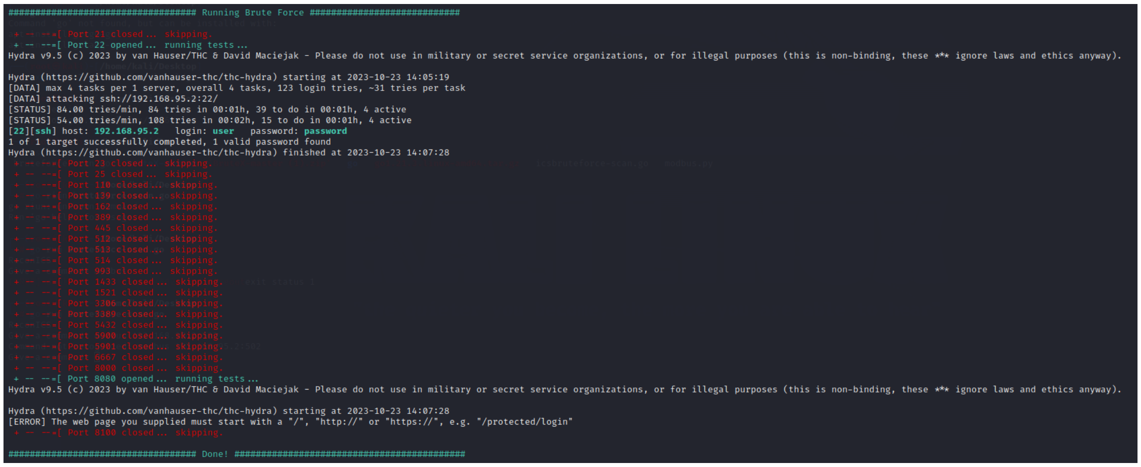
While executing a brute force attack, the tool successfully uncovered and provided both the user and password for port 22.

To ensure that the tool is working and can find the Modbus port 502, the port was



```
┌──(root㉿kali)-[/home/kali/Desktop/BruteX-master]
└─# brutex 192.168.95.2

 + -- --=[ BruteX v2.5 by @xer0dayz
 + -- --=[ http://sn1persecurity.com

################################### Running Port Scan ##############################
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-23 14:05 EDT
Nmap scan report for 192.168.95.2
Host is up (0.13s latency).
Not shown: 22 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 5.52 seconds

################################### Running Brute Force ############################
```

Figure 5.1. Scan Results - 192.168.95.2



```
################################### Running Brute Force ############################
 + -- --=[ Port 21 closed... skipping.
 + -- --=[ Port 22 opened... running tests...
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-10-23 14:05:19
[DATA] max 4 tasks per 1 server, overall 4 tasks, 123 login tries, ~31 tries per task
[DATA] attacking ssh://192.168.95.2:22/
[STATUS] 84.00 tries/min, 84 tries in 00:01h, 39 to do in 00:01h, 4 active
[STATUS] 54.00 tries/min, 108 tries in 00:02h, 15 to do in 00:01h, 4 active
[22][ssh] host: 192.168.95.2   login: user   password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-10-23 14:07:28
 + -- --=[ Port 23 closed... skipping.
 + -- --=[ Port 25 closed... skipping.
 + -- --=[ Port 110 closed... skipping.
 + -- --=[ Port 139 closed... skipping.
 + -- --=[ Port 162 closed... skipping.
 + -- --=[ Port 389 closed... skipping.
 + -- --=[ Port 445 closed... skipping.
 + -- --=[ Port 512 closed... skipping.
 + -- --=[ Port 513 closed... skipping.
 + -- --=[ Port 514 closed... skipping.
 + -- --=[ Port 993 closed... skipping.
 + -- --=[ Port 1433 closed... skipping.
 + -- --=[ Port 1521 closed... skipping.
 + -- --=[ Port 3306 closed... skipping.
 + -- --=[ Port 3389 closed... skipping.
 + -- --=[ Port 5432 closed... skipping.
 + -- --=[ Port 5900 closed... skipping.
 + -- --=[ Port 5901 closed... skipping.
 + -- --=[ Port 6667 closed... skipping.
 + -- --=[ Port 8000 closed... skipping.
 + -- --=[ Port 8080 opened... running tests...
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-10-23 14:07:28
[ERROR] The web page you supplied must start with a "/", "http://" or "https://", e.g. "/protected/login"
 + -- --=[ Port 8100 closed... skipping.

################################### Done! ##########################################
```

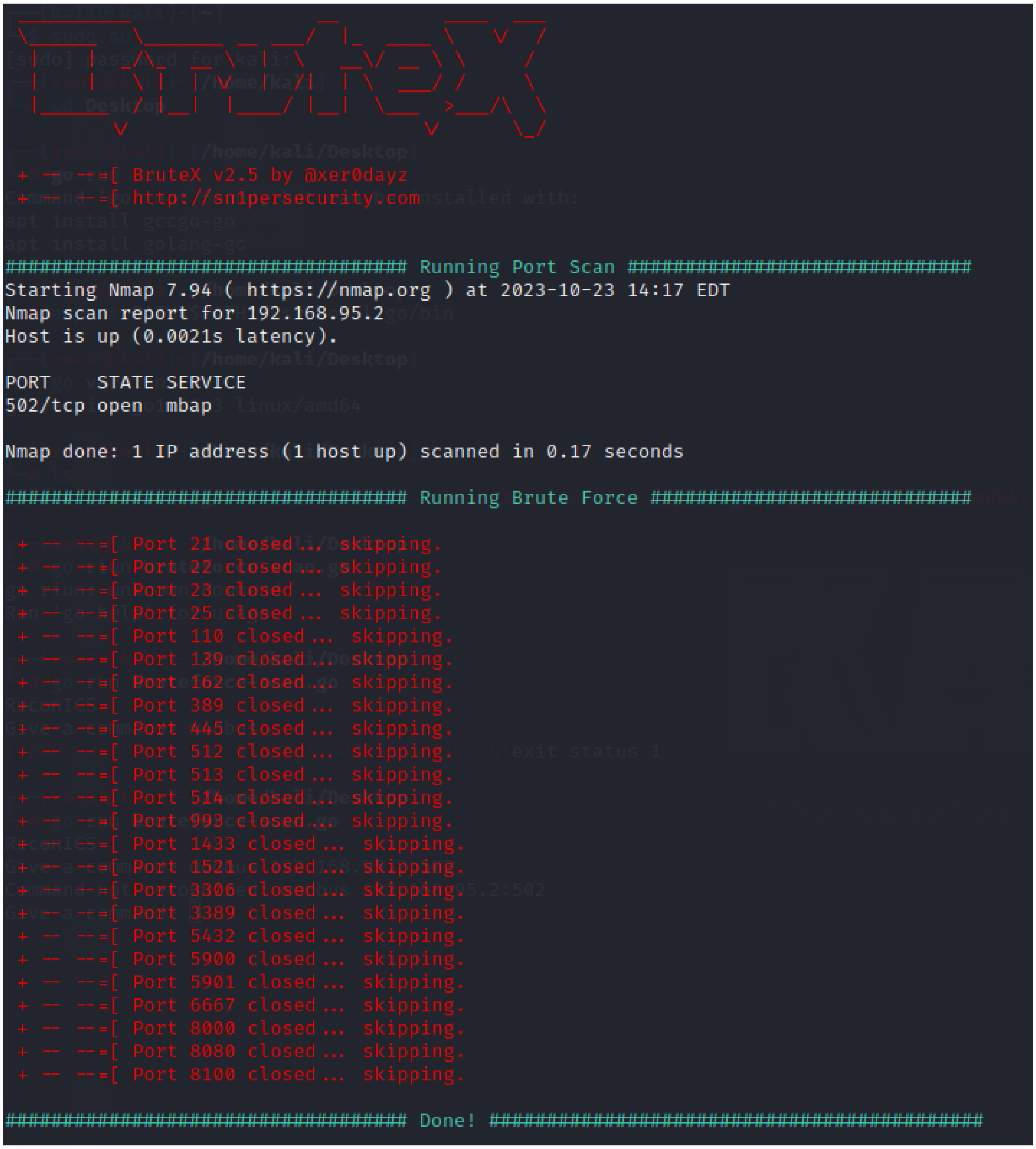
Figure 5.2. Brute Force Results - 192.168.95.2

specified in the scan, resulting in the finding and identifying the correct protocol.

5.2. Icssploit

Icssploit aims to offer a tool similar to Metasploit to perform exploitation in an ICS environment. The tool counts with modules, including:

- Creds: The creds module performs simple brute force attacks on a given target.
- Exploits: The exploits module has a different parameter for each PLC manufac-



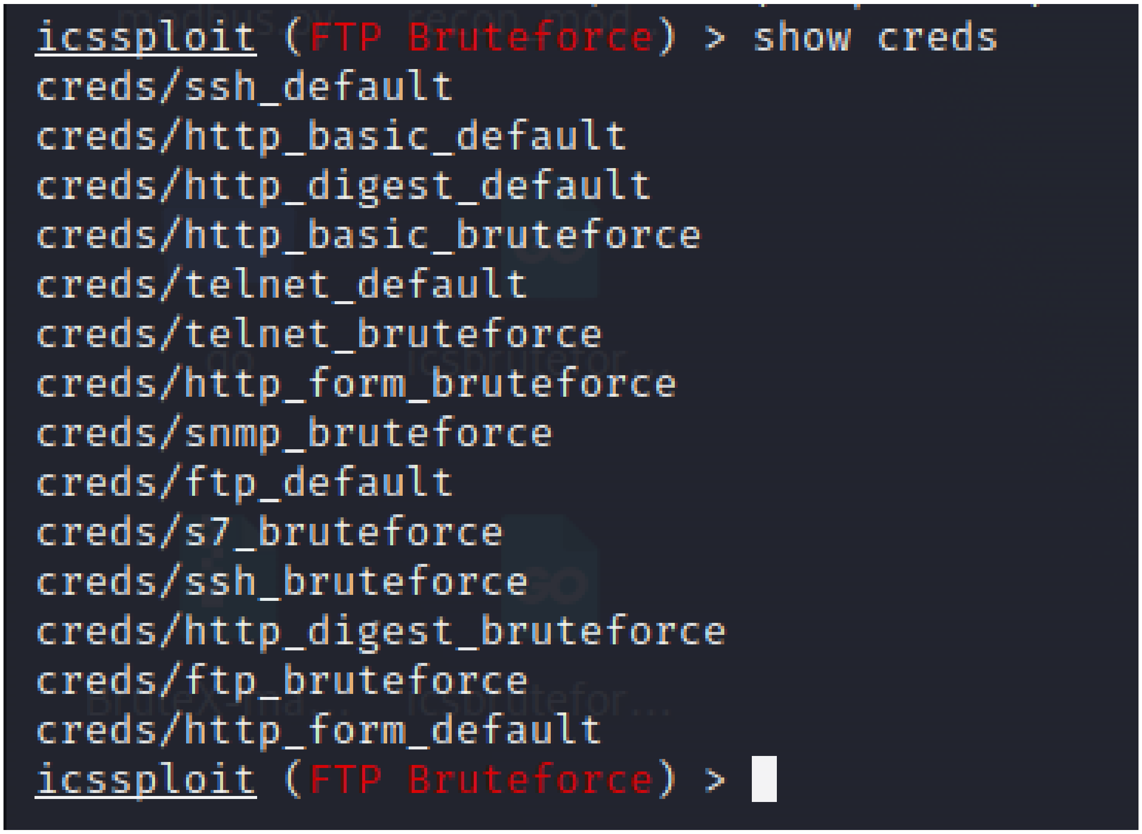
```
 + -- --=[ BruteX v2.5 by @xer0dayz
 + -- --=[ http://sn1persecurity.com

################################### Running Port Scan ##############################
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-23 14:17 EDT
Nmap scan report for 192.168.95.2
Host is up (0.0021s latency).

PORT    STATE SERVICE
502/tcp open  mbap

Nmap done: 1 IP address (1 host up) scanned in 0.17 seconds

################################### Running Brute Force ############################

 + -- --=[ Port 21 closed... skipping.
 + -- --=[ Port 22 closed... skipping.
 + -- --=[ Port 23 closed... skipping.
 + -- --=[ Port 25 closed... skipping.
 + -- --=[ Port 110 closed... skipping.
 + -- --=[ Port 139 closed... skipping.
 + -- --=[ Port 162 closed... skipping.
 + -- --=[ Port 389 closed... skipping.
 + -- --=[ Port 445 closed... skipping.
 + -- --=[ Port 512 closed... skipping.
 + -- --=[ Port 513 closed... skipping.
 + -- --=[ Port 514 closed... skipping.
 + -- --=[ Port 993 closed... skipping.
 + -- --=[ Port 1433 closed... skipping.
 + -- --=[ Port 1521 closed... skipping.
 + -- --=[ Port 3306 closed... skipping.
 + -- --=[ Port 3389 closed... skipping.
 + -- --=[ Port 5432 closed... skipping.
 + -- --=[ Port 5900 closed... skipping.
 + -- --=[ Port 5901 closed... skipping.
 + -- --=[ Port 6667 closed... skipping.
 + -- --=[ Port 8000 closed... skipping.
 + -- --=[ Port 8080 closed... skipping.
 + -- --=[ Port 8100 closed... skipping.

################################### Done! ##########################################
```

Figure 5.3. Scan Results Port 502

turer. Moreover, there are different options for exploiting each of them.

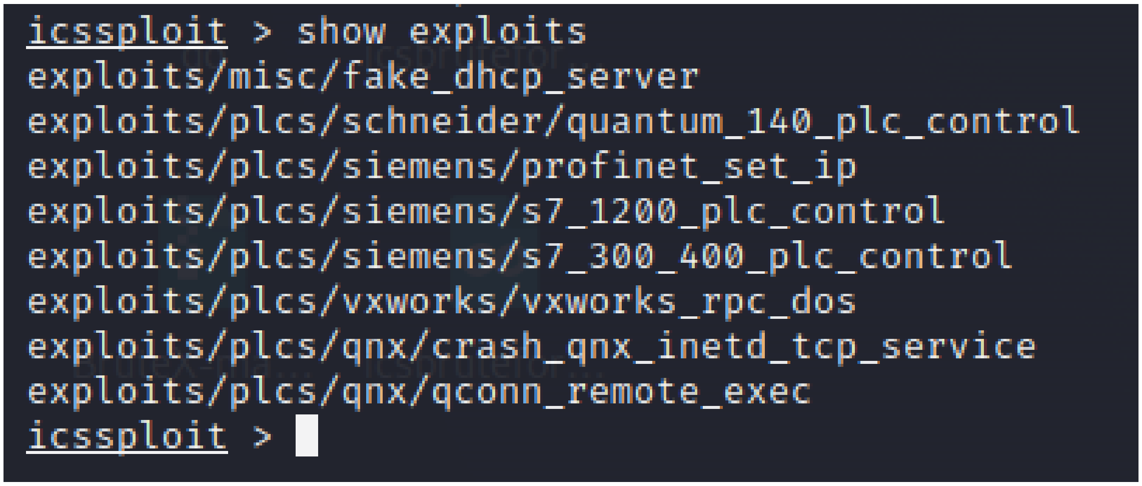
- Scanners: The scanners module will perform Nmap scanners, according to the manufacturer of the PLC.

Due to the limitations of the testbed used in this research, the only module explored was the Exploits. In the documentation of the testbed, the manufacturer of the



```
icssploit (FTP Bruteforce) > show creds
creds/ssh_default
creds/http_basic_default
creds/http_digest_default
creds/http_basic_bruteforce
creds/telnet_default
creds/telnet_bruteforce
creds/http_form_bruteforce
creds/snmp_bruteforce
creds/ftp_default
creds/s7_bruteforce
creds/ssh_bruteforce
creds/http_digest_bruteforce
creds/ftp_bruteforce
creds/http_form_default
icssploit (FTP Bruteforce) >
```

Figure 5.4. Icssploit Creds

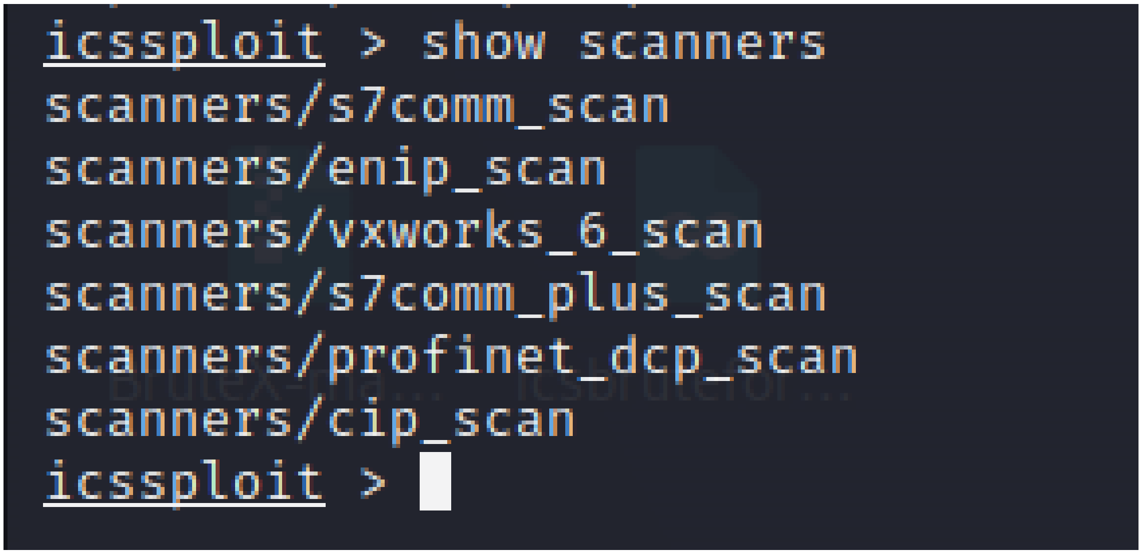


```
icssploit > show exploits
exploits/misc/fake_dhcp_server
exploits/plcs/schneider/quantum_140_plc_control
exploits/plcs/siemens/profinet_set_ip
exploits/plcs/siemens/s7_1200_plc_control
exploits/plcs/siemens/s7_300_400_plc_control
exploits/plcs/vxworks/vxworks_rpc_dos
exploits/plcs/qnx/crash_qnx_inetd_tcp_service
exploits/plcs/qnx/qconn_remote_exec
icssploit >
```

Figure 5.5. Icssploit Exploits

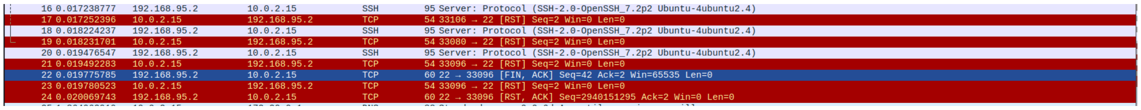
PLC is not specified, so some tests with the Icssploit were made to gather this information. Wireshark was used to record a Pcap file, and Zeek [23] was used to analyze the file.

Running the options from QNX, Siemens, and Vxworks, the connection failed initially, leading to manufacturers’ incompatibility.



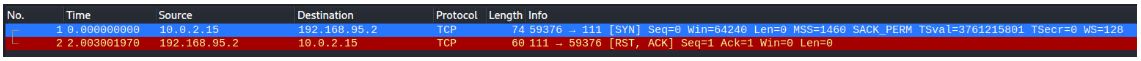
```
icssploit > show scanners
scanners/s7comm_scan
scanners/enip_scan
scanners/vxworks_6_scan
scanners/s7comm_plus_scan
scanners/profinet_dcp_scan
scanners/cip_scan
icssploit > 
```

Figure 5.6. Icssploit Scanners



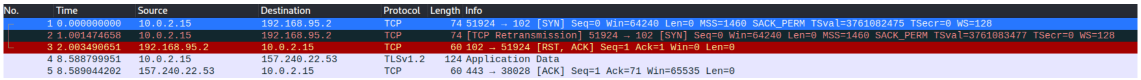
```
16 0.017238777  192.168.95.2  10.0.2.15     SSH  95 Server: Protocol (SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4)
17 0.017252396  10.0.2.15     192.168.95.2  TCP  54 33106 → 22 [RST] Seq=2 Win=0 Len=0
18 0.018224237  192.168.95.2  10.0.2.15     SSH  95 Server: Protocol (SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4)
19 0.018231701  10.0.2.15     192.168.95.2  TCP  54 33080 → 22 [RST] Seq=2 Win=0 Len=0
20 0.019476547  192.168.95.2  10.0.2.15     SSH  95 Server: Protocol (SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4)
21 0.019492283  10.0.2.15     192.168.95.2  TCP  54 33096 → 22 [RST] Seq=2 Win=0 Len=0
22 0.019775785  192.168.95.2  10.0.2.15     TCP  60 22 → 33096 [FIN, ACK] Seq=42 Ack=2 Win=65535 Len=0
23 0.019780523  10.0.2.15     192.168.95.2  TCP  54 33096 → 22 [RST] Seq=2 Win=0 Len=0
24 0.020069743  192.168.95.2  10.0.2.15     TCP  60 22 → 33096 [RST, ACK] Seq=2940151295 Ack=2 Win=0 Len=0
```

Figure 5.7. QX Exploit



```
No.  Time           Source         Destination    Protocol  Length  Info
   1 0.000000000    10.0.2.15      192.168.95.2   TCP           74 59376 → 111 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3761215801 TSecr=0 WS=128
   2 2.003001970    192.168.95.2   10.0.2.15      TCP           60 111 → 59376 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

Figure 5.8. Vxworks Exploit

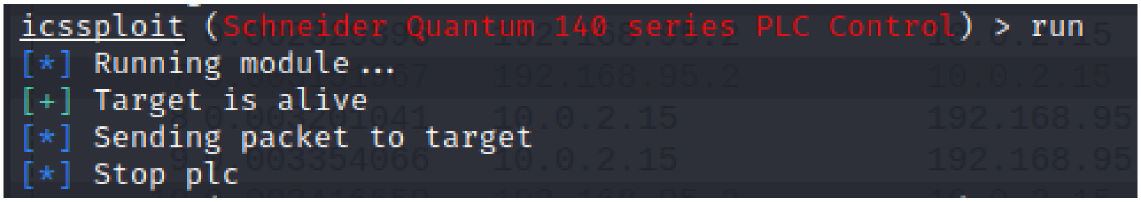


```
No.  Time          Source         Destination    Protocol  Length  Info
   1 0.000000000   10.0.2.15      192.168.95.2   TCP           74 51924 → 102 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3761082475 TSecr=0 WS=128
   2 1.001474658   10.0.2.15      192.168.95.2   TCP           74 [TCP Retransmission] 51924 → 102 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3761083477 TSecr=0 WS=128
   3 2.003490651   192.168.95.2   10.0.2.15      TCP           60 102 → 51924 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   4 8.588799951   10.0.2.15      157.240.22.53  TLSv1.2      124 Application Data
   5 8.589044202   157.240.22.53  10.0.2.15      TCP           60 443 → 38028 [ACK] Seq=1 Ack=71 Win=65535 Len=0
```

Figure 5.9. Siemens Exploit

When running the manufacturer Schneider, it was possible to connect and analyze the communication between the PLC and the exploit tool.

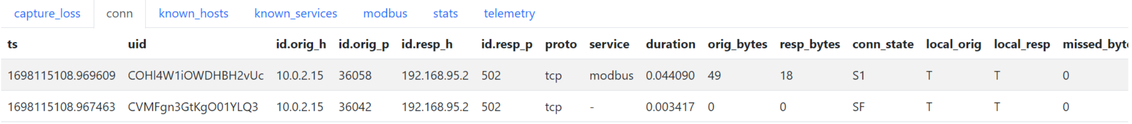
Analyzing the Pcap file from the Schneider exploit with Zeek, it was possible to check the codes in the conn section: S1 and SF. According to Zeek documentation, S1



```
icssploit (Schneider Quantum 140 series PLC Control) > run
[*] Running module...
[+] Target is alive
[*] Sending packet to target
[*] Stop plc
```

Figure 5.10. Schneider Exploit

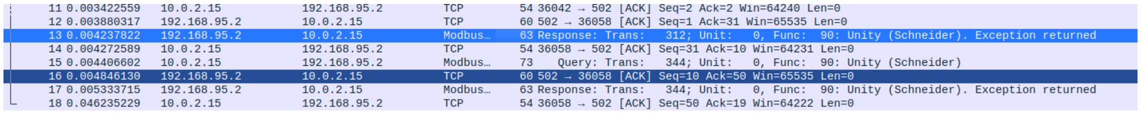
means that the connection was established but not terminated, and SF means that the connection was usually established and terminated [23].



ts	uid	id.orig_h	id.orig_p	id.resp_h	id.resp_p	proto	service	duration	orig_bytes	resp_bytes	conn_state	local_orig	local_resp	missed_byt
1698115108.969609	COHl4W1iOWDHBH2vUc	10.0.2.15	36058	192.168.95.2	502	tcp	modbus	0.044090	49	18	S1	T	T	0
1698115108.967463	CVMFgn3GtKgO01YLQ3	10.0.2.15	36042	192.168.95.2	502	tcp	-	0.003417	0	0	SF	T	T	0

Figure 5.11. Pcap Analysis Zeek - Schneider Exploit

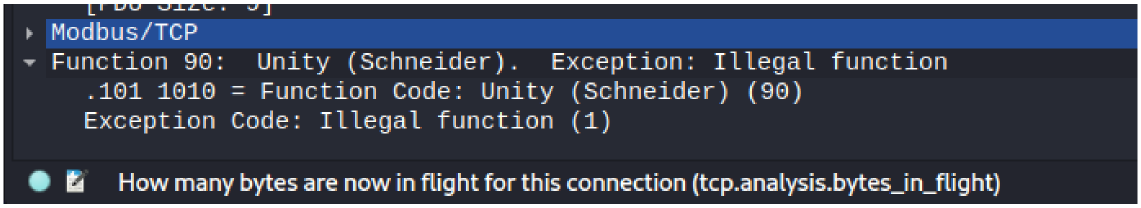
Along with the Pcap analysis, using Wireshark, it is possible to see that the response code from the secondary for the request made by the tool was: Illegal Function, which can be understood as the PLC receiving the request but not being able to process the task. This is possible as the function code was not implemented in the testbed unit and only applies to new devices. But it also can indicate that the secondary is in the wrong state to process this type of request, for instance, if it is misconfigured and is being asked to return register values. [13]



```
11 0.003422559   10.0.2.15      192.168.95.2   TCP       54 36042 → 502 [ACK] Seq=2 Ack=2 Win=64240 Len=0
12 0.003880317   192.168.95.2   10.0.2.15      TCP       60 502 → 36058 [ACK] Seq=1 Ack=31 Win=65535 Len=0
13 0.004237822   192.168.95.2   10.0.2.15      Modbus…   63 Response: Trans:   312; Unit:   0, Func:  90: Unity (Schneider). Exception returned
14 0.004272589   10.0.2.15      192.168.95.2   TCP       54 36058 → 502 [ACK] Seq=31 Ack=10 Win=64231 Len=0
15 0.004406602   10.0.2.15      192.168.95.2   Modbus…   73    Query: Trans:   344; Unit:   0, Func:  90: Unity (Schneider)
16 0.004846130   192.168.95.2   10.0.2.15      TCP       60 502 → 36058 [ACK] Seq=10 Ack=50 Win=65535 Len=0
17 0.005333715   192.168.95.2   10.0.2.15      Modbus…   63 Response: Trans:   344; Unit:   0, Func:  90: Unity (Schneider). Exception returned
18 0.046235229   10.0.2.15      192.168.95.2   TCP       54 36058 → 502 [ACK] Seq=50 Ack=19 Win=64222 Len=0
```

Figure 5.12. Pcap Analysis Wireshark - Schneider Exploit

Although the exploitation did not go through, for a reconnaissance step, the analysis was very enlightening. Knowing the manufacturer of the PLC and that the equipment is probably misconfigured or outdated can lead to a manual attack preparation that ap-



```
    [PDU Size: 5]
▸ Modbus/TCP
▾ Function 90:  Unity (Schneider).  Exception: Illegal function
    .101 1010 = Function Code: Unity (Schneider) (90)
    Exception Code: Illegal function (1)
```

How many bytes are now in flight for this connection (tcp.analysis.bytes_in_flight)

Figure 5.13. Pcap Analysis Wireshark 2 - Schneider Exploit

plies more robust techniques to achieve the target.

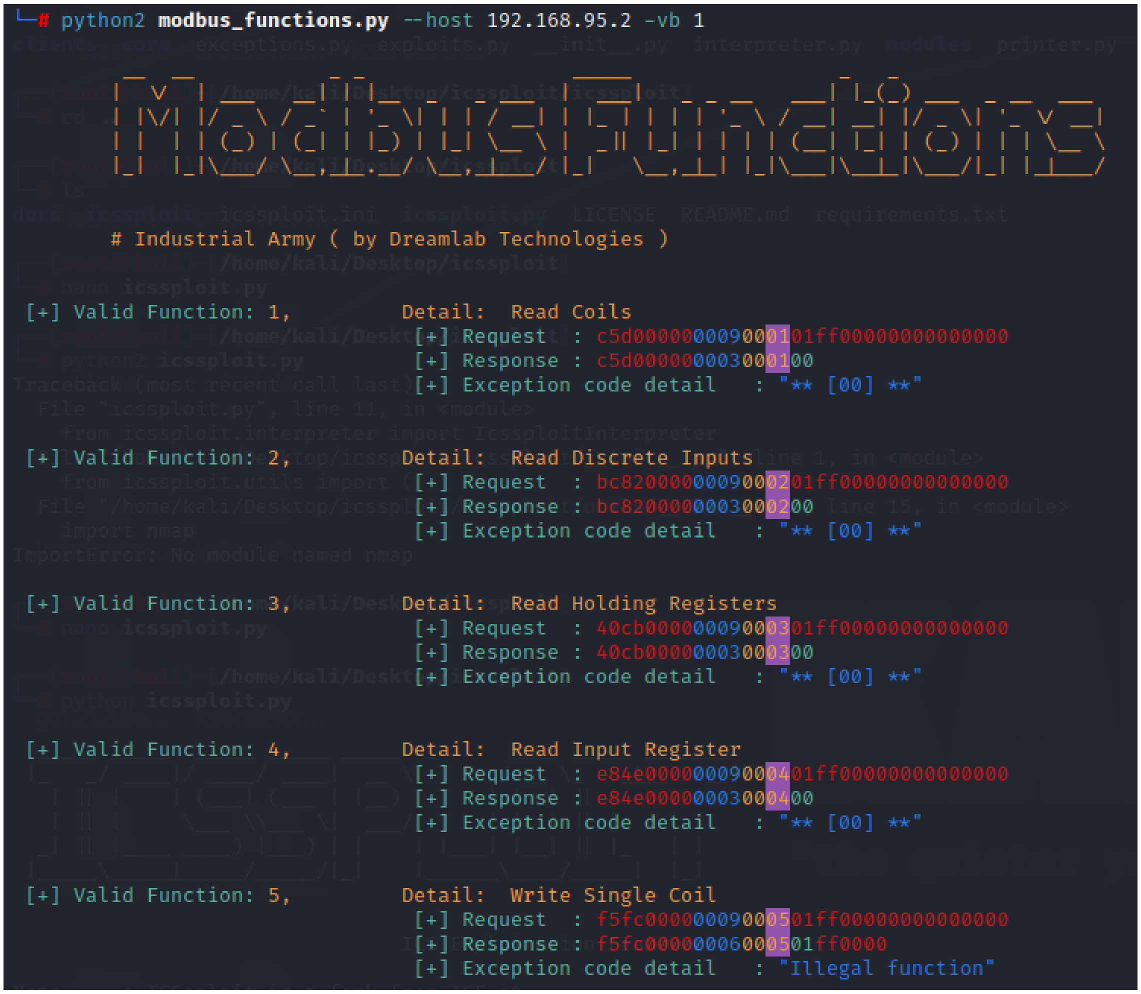
5.3. Recon Modbus Functions

This reconnaissance looks for all the functions a Modbus protocol has. The results have information about the secondary, their ID, and what they should perform, such as reading data, accepting data, and reporting status. The function table can be found in Appendix A [14].

In the results, it is possible to identify the function the secondary is performing, his ID, and all the functions the plc from the testbed has. An attacker can use this information to send requests, for example, for the secondary that performs function eight and subfunction ten to cause the server to clear its counters and the diagnostic register.

An Attacker can start his attack in the reconnaissance phase by scanning the network trying to find Modbus devices with Modbus diagnostic commands: Clear Counters and Diagnostic Register: a request sent to PLC, with function code 8 (0x08) and subfunction code 10 (0x0A), will trigger the clearing of the diagnostic register and counters on the target server. Usually, only serial devices implement this function.

Another option is to obtain data by looking for the device identity and asking the secondary to send you a package that carries out function 43. The attacker will obtain the vendor name, product name, and version number from this return. In addition, they can



```
└─# python2 modbus_functions.py --host 192.168.95.2 -vb 1

 __  __           _ _               _____                 _   _
|  \/  | ___   __| | |__  _   _ ___|  ___|   _ _ __   ___| |_(_) ___  _ __  ___
| |\/| |/ _ \ / _` | '_ \| | | / __| |_ | | | | '_ \ / __| __| |/ _ \| '_ \/ __|
| |  | | (_) | (_| | |_) | |_| \__ \  _|| |_| | | | | (__| |_| | (_) | | | \__ \
|_|  |_|\___/ \__,_|_.__/ \__,_|___/_|   \__,_|_| |_|\___|\__|_|\___/|_| |_|___/

        # Industrial Army ( by Dreamlab Technologies )

 [+] Valid Function: 1,         Detail:  Read Coils
                                 [+] Request  : c5d000000009000101ff000000000000
                                 [+] Response : c5d00000000300010 0
                                 [+] Exception code detail   : "** [00] **"

 [+] Valid Function: 2,         Detail:  Read Discrete Inputs
                                 [+] Request  : bc8200000009000201ff000000000000
                                 [+] Response : bc820000000300020 0
                                 [+] Exception code detail   : "** [00] **"

 [+] Valid Function: 3,         Detail:  Read Holding Registers
                                 [+] Request  : 40cb00000009000301ff000000000000
                                 [+] Response : 40cb0000000300030 0
                                 [+] Exception code detail   : "** [00] **"

 [+] Valid Function: 4,         Detail:  Read Input Register
                                 [+] Request  : e84e00000009000401ff000000000000
                                 [+] Response : e84e0000000300040 0
                                 [+] Exception code detail   : "** [00] **"

 [+] Valid Function: 5,         Detail:  Write Single Coil
                                 [+] Request  : f5fc00000009000501ff000000000000
                                 [+] Response : f5fc00000006000501ff0000
                                 [+] Exception code detail   : "Illegal function"
```

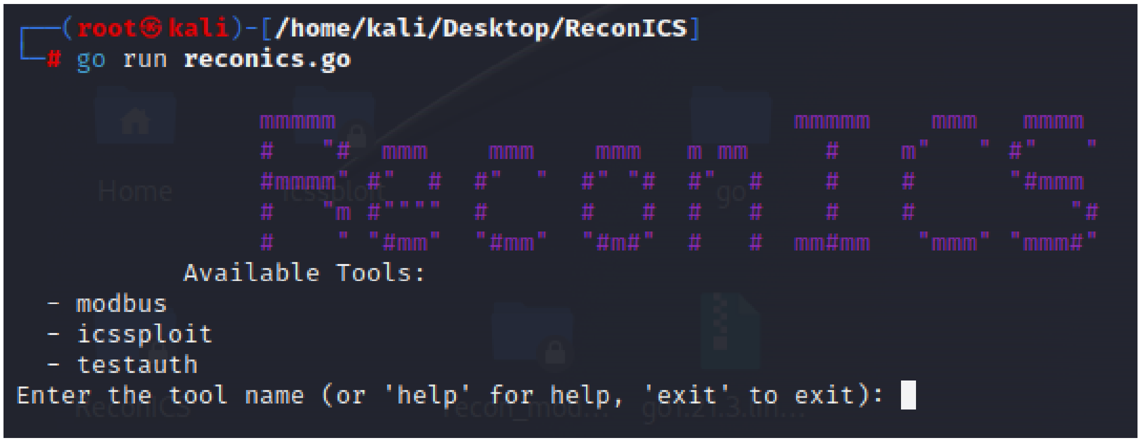
Figure 5.14. Modbus Reconnaissance Results

look for vulnerabilities unique to the model for which they have the information [15].

5.4. ReconICS - Tool

As a contribution to this work, a tool was built to compile the reconnaissance of ICS devices. The ReconICS tool counts with a Port Scanner, focused on the main ICS ports and more usual network ports used in the industry. A Brute Force attack performer, where the user can input the target and the wordlist desire to verify the security of their ICS environment.

Furthermore, Recon Modbus Functions and Icssploit are integrated to give the user a better understanding of their equipment and the possible misconfiguration or updates they need to do.



```
┌──(root㉿kali)-[/home/kali/Desktop/ReconICS]
└─# go run reconics.go

               mmmmm                            mmmmm    mmm   mmmm
               #   "#  mmm    mmm    mmm  m mm    #    m"   " #"   "
               #mmmm" #"  #  #"  "  #" "# #"  #   #    #      "#mmm
               #   "m #""""  #      #   # #   #   #    #          "#
               #    " "#mm"  "#mm"  "#m#" #   # mm#mm   "mmm" "mmm#"
          Available Tools:
  - modbus
  - icssploit
  - testauth
Enter the tool name (or 'help' for help, 'exit' to exit): █
```

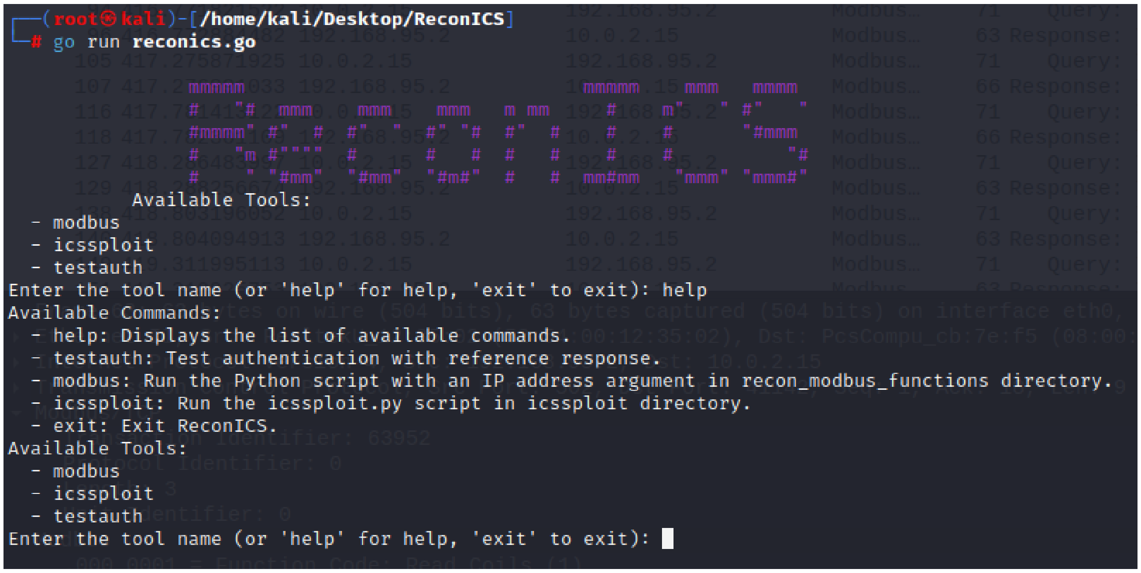
Figure 5.15. ReconICS - First Screen

ReconICS was written in Golang programming language, just like ReconFTW, and has a command line interface where the user can interact with the tool by given commands. Once the tool is started, it will automatically download the dependencies to all the integrated tools. It is an easy and fast way to start the reconnaissance process and can be used directly in the command prompt.

The commands that can be used with ReconICS are:

- Modbus: Will run the Recon Modbus Functions Tools
- Icssploit: Will run the Icssploit tool.
- Testauth: Will perform a scan and a brute force attack
- Help: Will show information about the usage of the tool

To install, the user will need to go to the GitHub page from the project in the URL: https://github.com/nsoare2/ReconICS and follow the steps from the Readme file. Some prerequisites will be needed since the tool needs the Go and Python commands to run.



```
┌──(root㉿kali)-[/home/kali/Desktop/ReconICS]
└─# go run reconics.go

                mmmmm                               mmmmm    mmm   mmmm
                #   "#  mmm    mmm    mmm   m mm      #    m"   " #"   "
                #mmmm" #"  #  #"  "  #" "#  #"  #     #    #      "#mmm
                #   "m #""""  #      #   #  #   #     #    #          "#
                #    " "#mm"  "#mm"  "#m#"  #   #   mm#mm   "mmm" "mmm#"
           Available Tools:
  - modbus
  - icssploit
  - testauth
Enter the tool name (or 'help' for help, 'exit' to exit): help
Available Commands:
  - help: Displays the list of available commands.
  - testauth: Test authentication with reference response.
  - modbus: Run the Python script with an IP address argument in recon_modbus_functions directory.
  - icssploit: Run the icssploit.py script in icssploit directory.
  - exit: Exit ReconICS.
Available Tools:
  - modbus
  - icssploit
  - testauth
Enter the tool name (or 'help' for help, 'exit' to exit): █
```

Figure 5.16. ReconICS - Commands Options

Chapter 6. Conclusion

Although reconnaissance is a well-established concept within the cybersecurity community, its application to critical infrastructure still needs to be explored. This observation is exemplified in Williams' research titled ”Distinguishing Internet-facing ICS devices using PLC programming information” [19], which underscores the discrepancy between the prevailing NIST guidelines advocating against internet connectivity for Industrial Control Systems (ICS) devices and the actual practices within the industry. This incongruity has engendered a false sense of security among professionals who previously assumed their ICS devices were isolated from external networks. This has resulted in a reduced inclination for initial cybersecurity assessments while constructing critical infrastructure. While there has been some perceptible change in this mentality over time, it has occurred gradually, primarily driven by the apprehension of potential losses.

An analysis of the data derived from utilizing the tools discussed in this study has afforded us valuable insights into the characteristics of the environment. It has illuminated vulnerabilities while providing crucial information about the target, as elucidated in Section 5, wherein a comprehensive understanding of the PLC's functionalities was achieved. Importantly, it is noteworthy that the application of these tools did not inflict any harm upon the testbed environment. Nevertheless, it is imperative to subject them to further testing within diverse environments to ascertain their safety and effectiveness.

The significance of this research manifests in the prospect of conducting precise and minimally invasive scans, thereby augmenting the level of security assessment through a versatile tool that can be customized to meet the unique requirements of organizations seeking to enhance their knowledge of their operational landscape. Additionally, it underscores the pivotal role of security measures, particularly within critical infrastructure, where the emphasis on availability often supersedes confidentiality. Notably, the research highlights the frequent online exposure of PLC devices, as demonstrated through tools such as Shodan, reinforcing the imperative need for heightened security measures.

Appendix A. Modbus Functions

Function type	Function name	Function code
Data Access	Read Discrete Inputs	2
	Read Coils	1
	Write Single Coil	5
	Write Multiple Coils	15
	Read Input Registers	4
	Read Multiple Holding Registers	3
	Write Single Holding Register	6
	Write Multiple Holding Registers	16
	Read/Write Multiple Registers	23
	Mask Write Register	22
	Read FIFO Queue	24
	Read File Record	20
	Write File Record	21
Diagnostics	Read Exception Status	7
	Diagnostic	8
	Get Com Event Counter	11
	Get Com Event Log	12
	Report Slave ID	17
	Read Device Identification	43
Other	Encapsulated Interface Transport	43

Table A.1. Modbus Functions Table

Bibliography

[1] Alves, Thiago, and Morris, Thomas, *OpenPLC: An IEC 61,131–3 compliant open source industrial controller for cyber security research*, Computers and Security, Volume 78, 2018.

[2] 1N3, *BruteForce X*, 2018, https://github.com/1N3/BruteX. (2023)

[3] Ackerman, Pascal, *Industrial Cybersecurity*, Second Edition, Packt Publishing, 2021.

[4] Alves, Thiago R., *OpenPLC v2*, 2016, https://github.com/thiagoralves/OpenPLC_ v2. (2023)

[5] Army, Industrial, *Recon Modbus Functions*, 2020, https://github.com/industrialarmy/recon_modbus_functions. (2023)

[6] America‘s Cyber Defense Agency, *Cyber-Attack Against Ukrainian Critical Infrastructure*, America‘s Cyber Defense Agency, 2021.

[7] Assante, Michael J, Lee, Robert M., *The Industrial Control System Cyber Kill Chain*, Sans Institute, 2015.

[8] Cabana, O., Youssef, A.M., Debbabi, M., Lebel, B., Kassouf, M., Agba, B.L., *Detecting, Fingerprinting and Tracking Reconnaissance Campaigns Targeting Industrial Control Systems*, Springer, Cham, 2019.

[9] Formby, David, Rad, Milad and, Beyah, Raheem, *Lowering the Barriers to Industrial Control System Security with GRFICS*, 2018 USENIX Workshop on Advances in Security Education (ASE 18), 2018.

[10] Green, Benjamin and Krotofil, Marina and Abbasi, Ali, *On the Significance of Process Comprehension for Conducting Targeted ICS Attacks*, Association for Computing Machinery, 2017.

[11] Lemos, Robert, *Early Discovery of Pipedream Malware a Success Story for Industrial Security*, DarkReading, 2022.

[12] Lenthang, Marlene and, Margolin, Josh, *Ransomware cyberattack shuts down major US pipeline, company says*, ABC News, 2021.

[13] Modbus, Simply, *Exception Responses*, 2020, https://www.simplymodbus.ca/exceptions.htm. (2023)

[14] Ozeki, *Modbus Function Codes*, https://ozeki.hu/p_5873-modbus-function-codes.html. (2023)

[15] Radiflow, *Hack the Modbus*, 2019, https://www.radiflow.com/blog/hack-the-modbus/. (2023)

[16] Six2dez, *ReconFTW*, 2021, https://github.com/six2dez/reconftw. (2023)

[17] Tijldeneut, *Icssploit*, 2017, https://github.com/tijldeneut/icssploit. (2023)

[18] Wakchaure, Mahesh, Sarwade, Satish and, Siddavatam, Irfan, *Reconnaissance of Industrial Control System By Deep Packet Inspection*, 2nd IEEE International Conference on Engineering and Technology (ICETECH), 2016.

[19] Williams, Paul M., *Distinguishing Internet-facing ICS devices using PLC programming information programming information*, Air Force Institute of Technology, 2014.

[20] National Institute of Standards and Technology, *Guide to Industrial Control Systems (ICS) Security*, 2015.

[21] Schneider Electric, *Modbus Master-Slave Principle*, https://product-help.schneider-electric.com. (2023)

[22] IPcomm, *IEC 60870-5-104*, https://www.ipcomm.de/protocol/IEC104/.(2023)

[23] Zeek, *The Zeek Project*, 2020, https://zeek.org/. (2023)

Vita

Nathalia de Sa Soares, a native of Rio de Janeiro, Brazil, earned her bachelor's degree from Universidade Federal Fluminense in December 2017. Since January 2022, she has been a member of the LSU Applied Cybersecurity Lab (ACL), where she had the opportunity to delve into her interests in the field of cybersecurity and has actively contributed to research projects. Before this, Nathalia spent approximately four years working in cybersecurity in the financial sector in Brazil. Furthermore, she is a member of the Baggili Truth Lab (Bitlab), conducting research in Pentesting and providing assistance to fellow researchers in the lab. Nathalia is on track to complete her master's degree in December 2023 and intends to continue her career in the cybersecurity field.