University of Padova

Department of Mathematics "Tullio Levi-Civita"

Master Thesis in Cybersecurity

VIRTUEPOT: A High-Fidelity and High-Interaction Virtual Honeypot for Industrial Control Systems

Supervisor
Prof. Mauro Conti
University of Padova

Co-supervisor
Dr. Federico Turrin
Spritz Matter Srl
University of Padova Spin-off

Master Candidate
Nikhil Karakuchi Chidananda

Student ID
2005655

Academic Year
2023-2024

"Self-belief and hard work will always earn you success."
— Virat Kohli

Abstract

Industrial Control Systems (ICS) are essential for managing and controlling various industrial activities such as energy production, manufacturing, wastewater management, and transportation. However, as these systems become more interconnected and digitized, they face increasing cybersecurity threats. To address these issues, this research explores the use of honeypots as a proactive cybersecurity tool to protect Industrial Control Systems. A honeypot is an effective tool for studying attacks on ICS and developing defence methods to protect against these attacks. Currently, the ICS industry is facing a growing number of cyber threats, with attackers becoming more sophisticated. As a result, it has become more challenging to create honeypots that can effectively detect and respond to attacks, log interactions, and capture changes in the physical processes of ICS.

Our research aims to gain valuable insights into attack patterns and behaviours using honeypots. By doing so, we can gather crucial information about the latest Tactics, Techniques, and Procedures (TTPs) used by attackers, as well as their technical knowledge and capabilities. In this thesis, we introduce VirtuePot, a honeypot that focuses on the physical interaction and design of ICS honeypots. VirtuePot simulates the behaviour and services of real Programmable Logic Controllers (PLCs) using dynamic service simulations. This includes advanced simulations of industrial processes, communication protocols, and command responses. We deployed VirtuePot both in the cloud (using DigitalOcean) and locally on-premise at the VSIX Internet Exchange Point, and collected data over 61 days.

Our findings show that VirtuePot recorded a significant amount of ICS interactions from around the world. The log analysis revealed that the on-premise deployment at the VSIX Internet Exchange Point attracted more realistic attacks compared to the cloud (DigitalOcean) deployment. This indicates that attackers are actively targeting ICS systems, and the deployment location can impact the nature and realism of the attacks encountered.

Keywords: Cyber-physical system (CPS); Honeypot; Programmable Logic Controller (PLC); Industrial Control Systems (ICS); SCADA;

Contents

Listing of figures

Listing of tables

Listing of acronyms

CPS ..... Cyber-Physical System

ICS ..... Industrial Control Systems.

PLC ..... Programmable Logic Controller.

IXP ..... Internet Exchange Point.

TCP/IP ..... Transmission Control Protocol/Internet Protocol

HMI ..... Human Machine Interface

SCADA ..... Supervisory Control and Data Acquisition

OT ..... Operational Technology

LAN ..... Local Area Network

DMZ ..... Demilitarized Zone

HIL ..... Hardware-in-the-loop

LTS ..... Long-term support

DNP3 ..... Distributed Network Protocol 3

OPC ..... Open Platform Communications

CPCS ..... Continuous Process Control System

PERA ..... Purdue Enterprise Reference Architecture

ERP ..... Enterprise Resource Planning

SIEM ..... Security Information and Event Management

ELK ..... Elasticsearch Logstash and Kiban

1 Introduction

Industrial Control Systems (ICS) are widely used in many industries, including energy, oil, and gas, water and wastewater treatment, chemical, automotive, and many more, these everyday utilities are part of our day-to-day life activities and have become so usual and any attack on utility ICS like the gas stations, for example, A cyberattack disrupted the sale of heavily subsidized gasoline in Iran, and most popular Stuxnet [6], a computer virus commonly thought to have been developed by the United States and Israel, was found in 2010 after being used to target the centrifuges in Iranian nuclear facilities. It was the first publicly reported instance of a virus attacking industrial machinery. So most of the industrial Control Systems were designed and installed almost a decade ago so most of them are operated in an air gaped system, now slowly we are converting them into smart and connected to the Internet and making it remotely controlled and monitored Also diagnose the operation remotely [7]. Therefore industrial control systems are increasingly exposed to attackers due to a lack of security measures in widely used communication protocols like Modbus, and S7 [8], Industrial Control Systems are increasingly vulnerable to new types of cyber threats, and Intrusion Detection Systems are critical for detecting potential attacks and malicious activities, making earlier studies valuable for further research in the industrial safety context [9]. as per the recent IXP network traffic analysis for the ICS protocol, the study shows that 75% of the Industrial Control systems are still communicating unencrypted, without integrity protection [10].

The recent study shows that attacks on ICS especially PLCs, there are many honeypots developed in the past [11] [12] [13] [14] [15]. Researchers can utilize honeypots to learn about new

hacker strategies and virus behaviour [16]. However, existing ICS honeypot solutions lack the critical functionality to collect data on some of the most recent and sophisticated attack tactics. A further drawback of the present research is the absence of PLC and network connectivity expansion options. Solving this issue is critical since ICS settings are quite diverse in terms of device categories and network protocols. Another disadvantage is that the majority of present solutions have little interactivity. This interaction level severely limits the usefulness of data acquired from attacker interactions.

Thesis Statement: VIRTUEPOT

We have developed an innovative High-interaction honeypot for Programmable Logic Controllers (PLCs) by integrating various established technologies such as Modbus and S7 etc., This honeypot addresses the limitations found in existing literature on honeypots and aims to enhance our understanding and response to emerging threats in the field of Industrial Control Systems (ICS).

In order to describe the thesis statement we extant VIRTUEPOT: A HighFidelity and High-Interaction Virtual Honeypot for Industrial Control Systems. That includes the TCP/IP Stack Simulation, Modbus, HMI, SCADA, HTTP Website, OpenPLC Server, Zeek for data logging, ELK Stack for data visualization, advanced simulation, and physical interaction. These programs can then be analyzed to uncover novel attack methods. This functionality is also unique to our approach.

1.1 Contributions

We provide the summary and the limitations of the current existing honeypot and using Virtuepot, the honeypot architecture can create multiple decoy PLCs.

This is the extended work of the ICSpot [15] by Federico Turrin and Francesco Trolese, where a honeypot implementation examined the first ICS honeypot that addresses the current state-of-the-art limitations by integrating a physical process interaction.

In this state-of-the-art thesis, we are going to address the shortcomings of current ICS honeypots. the primary limited physical interaction. Even though ICSs are defined by physical processes, their absence can result in an incomplete emulation of an industrial system. The source code of VirtuePot is available on GitHub [17] and logs of the honeypot data experiment are available on the archive [18]. Overall, the major contributions of this work are summarized as follows:

- We present two forms of ICS Honeypot deployment (On-prime and Cloud) and expose them to the internet.
- We introduce an ICS honeypot with an interactive physical process ability to mimic real-world ICS environments.
- We effectively engage and trick advanced network reconnaissance tools, achieving results that are close to real ICS device.
- VirtuePot integrates with SIEM systems for comprehensive reporting features, making it accessible for monitoring and analysing potential security incidents efficiently.
- Conduct a comprehensive data collection deployed at multiple locations worldwide.

1.2 Outline

In Chapter 2 we described the background and context of the research, chapter 3 describes the related research, in Chapter 4 explains the issues with the current existing state-of-the-art limitations, Chapter 5 we describe the architecture development of our honeypot, Chapter 6 explains the implantation methods used to evaluate the honeypot. Chapter 7 presents the evaluation findings and results. Finally, Chapter 8 provides a conclusion, findings, and avenues for further research.

2

Background and Theory

2.1 Industrial Control Systems

Industrial Control Systems (ICS) term refers to the wide range of control systems that including the Programmable Logic Controller(PLC), Supervisory Control and Data Acquisition (SCADA), Remote Terminal Units (RTUs), and Distributed Control Systems (DCS) these are the commonly found in the manufacturing industries and critical infrastructures Figure **??** shows the relative positions of ICS, SCADA, DCS, and PLCs in the context of Cyber-Physical Systems (CPS) and Operational Technology (OT), An ICS made up of the components such as electrical, mechanical, compressor, hydraulic etc. work together to accomplish the industrial tasks such as power grid, natural gas, water management, manufacturing plant, nuclear power plant etc. Most of the time the ICS runs in the continuous process control system (CPCS), which means that it is commonly handled by the PLC, and the control here will be a fully automated process, or sometimes human control is also required in the system can be configured to operate in the open, closed, and manual mode. in the open mode, the output of the system is controlled by the pre-made settings or configurations, In the closed mode the control system output will depend on the input so it has a desired objective. In manual mode, the system is entirely operated by human beings.

Process Control Systems (PCS) are a particular kind of Industrial Control Systems that are responsible for managing and controlling continuous or batch operations, such as chemical

plants or water utilities, etc to guarantee the outcomes that are needed.

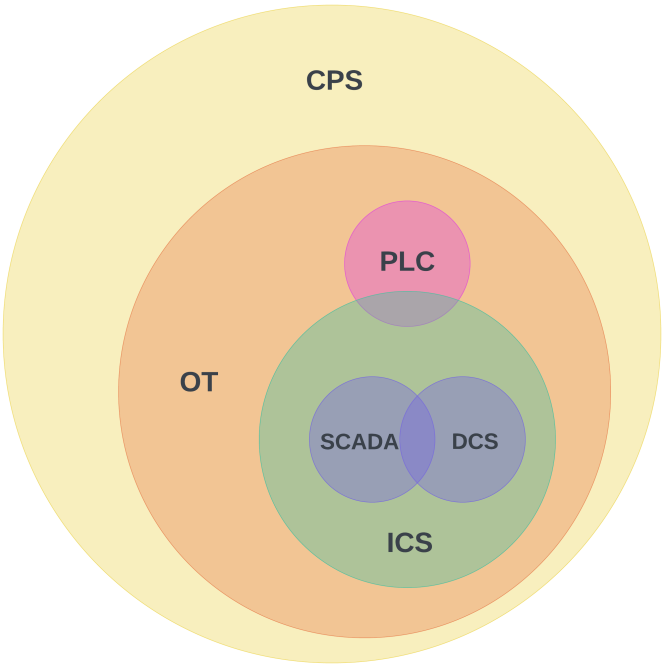

Figure 2.1: The Position of ICS with PLCs in the Overall View [1].

2.1.1 The Components of an OT System.

The basic operations of the operation technology are shown in Figure 2.2 The Key components included the following:

- **Sensors:** Sensors are embedded in ICS to provide data for process control, safety monitoring, quality assurance, and predictive maintenance. To automate and optimize industrial operations, data received from these sensors is frequently processed by programmable logic controllers (PLCs) or distributed control systems (DCS).

- **Actuators:** Actuators are critical components of control systems and automation, receiving information from sensors or controllers and performing specific operations or adjustments to maintain or alter the status of a system.

- **Controller:** The hardware or software that automatically adjusts a controlled variable in a control system is referred to as a "controller." Controllers are in charge of monitoring process variables, comparing them to setpoints or reference values, and making control

choices to keep the system in the intended condition. PLCs (Programmable Logic Controllers) are a form of controller that is widely utilized in industrial automation control systems.

- **Remote Diagnostics and Maintenance:** These utilities use technology, communication networks, and data analysis to provide proactive asset monitoring, troubleshooting, and maintenance from a remote location. Manufacturing, utilities (such as power production and water treatment), healthcare (for medical equipment), transportation (for fleets and infrastructure), and many other industries benefit exponentially from remote diagnosis and maintenance services.

- **Human-Machine Interface:** HMIs are critical in many sectors, including manufacturing, automation, and industrial control, since they provide a user-friendly and straightforward way of monitoring and operating challenging equipment and processes.

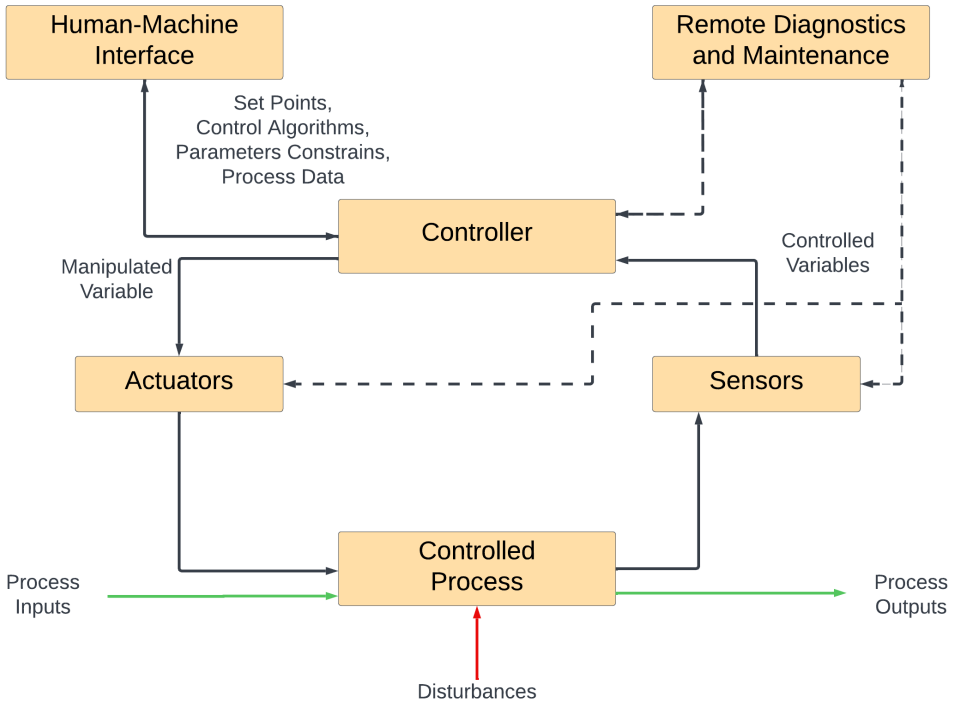

Figure 2.2: OT System Components [2]

2.1.2 Programmable Logic Controller(PLC)

A Programmable Logic Controller (PLC) [19] is a small industrial computer that executes logic tasks based on input signals from electrical hardware such as pumps, relays, timers, switches, and other devices. As a result, PLCs can regulate and automate complicated industrial processes, they are an essential component of Industrial Control Systems (ICS) and Supervisory Control and Data Acquisition (SCADA) [20] settings. In certain cases, PLCs are used as field devices, similar to Remote Terminal Units (RTUs), which are specialized control units designed for remote stations. PLCs are referred to as RTUs in such circumstances.

PLCs include programmable memory blocks that contain instructions for implementing various control system operations. These functions include, among other things, input and output control, counting, logic operations, communication, and arithmetic computations. PLC memory blocks are used to store data as well as program code.

PLC code can be written in a variety of programming languages, including, Ladder logic (LAD) [21] and function block diagram (FBD), which are both graphical. Text languages include structured text (ST) and instruction lists (IL). Sequential function charts (SFCs) are a graphical approach to organizing sequential or parallel processing programs. After the code is written, it is compiled, which results in the creation of assembly code.

2.1.3 Supervisory Control and Data Acquisition (SCADA)

SCADA systems [20] are essential in the automation and operations of industrial processes because they provide operators with a user-friendly interface for monitoring, controlling, and optimizing systems. They improve efficiency, safety, and dependability in many industries, including manufacturing, energy, water treatment, and transportation. Furthermore, SCADA systems contribute to data-driven decision-making and the ICS's operational integrity.

A crucial element of a SCADA system, the HMI [22] provides operators with a graphical interface via which they may interact with and monitor industrial processes and equipment. Typically, the HMI shows real-time data such as system status, process variables, alerts, and trends. The HMI allows operators to see the whole process and make knowledgeable choices.

SCADA systems gather data continually from numerous sensors, instruments, and control devices situated throughout the industrial plant. This information comprises temperature, pressure, flow rates, and other process factors. The HMI then displays the real-time data to the operators.

Control Functions: In addition to monitoring, SCADA systems often have control capabilities, allowing operators to make process changes. Depending on the system architecture, operators may use the HMI to start, stop, and modify equipment, set setpoints, and execute control methods.

Alarm and Event Handling: SCADA systems have alarm management capabilities. The SCADA system creates alarms when unexpected or out-of-spec events occur in the industrial process. These alerts are presented on the HMI, and operators can recognize them and take corrective action.

2.1.4 Industrial Control Systems architecture for process automation

In the early days, process control systems relied on pneumatic mechanisms and manually operated relay techniques. Later, analog electronic systems appeared, automating labour-intensive tasks. The advent of computerized control systems in the 1970s sketched a new era, leading to the field of computer-aided manufacturing. Over time, these techniques evolved alongside general-purpose computing technologies, creating a sophisticated distributed ecosystem of software applications and hardware. Termed 'automation systems,' they made it easy for computers to streamline manual processes.

Historically, control systems remained isolated, executing process management locally at respective production sites. However, as data availability increased and equipment complexity increased, the need emerged for interconnecting process control with enterprise (IT) networks and enabling third-party remote access. In the 1990s, T. J. Williams [23], part of the Purdue University Consortium for Computer Integrated Manufacturing, introduced the Purdue Enterprise Reference Architecture (PERA) shown in Figure 2.4. This model guided the interface design between process control functions and enterprise functions. Building upon PERA, the International Society of Automation (ISA) developed the ISA-95 international standard—a layered network model that facilitates vendor-independent information flows. It quickly gained prominence among OT professionals, shaping the implementation of industrial control systems within OT environments.

Level 0: Physical Process, Sensors and Actuators

This is the lower level, where physical processes appear. It includes sensors (such as temperature sensors, pressure sensors, etc.), actuators (such as motors, valves, etc.), and other machin-

ery. These devices directly interact with the physical environment (e.g., assembly lines, pumps, conveyor belts). Communication within this level is typically local and wired (e.g., field buses, Ethernet). Modern sensors may also communicate with cloud-based monitoring systems via cellular networks.

Level 1: Regulatory Control

This level contains devices that send control commands to Level 0. Key parts include: Programmable Logic Controllers (PLCs): These are specialized computers that monitor inputs (from sensors or manual input) and execute control logic. They adjust outputs (actuators) based on predefined rules. Remote Terminal Units (RTUs): RTUs connect Level 0 hardware to higher-level systems (Level 2). They gather data from sensors and send it to supervisory systems. Communication within Level 1 can be both local (wired) and remote (e.g., over the internet).

Level 2: Supervisory control

This zone supervises, monitors, and manages physical processes. Key components include:

Supervisory Control and Data Acquisition (SCADA) Software: SCADA oversees physical processes, whether locally or remotely. It aggregates data from sensors, PLCs, and other devices and sends it to data historians.

Distributed Control Systems (DCS): DCS performs similar functions to SCADA but is typically deployed locally within a specific area (e.g., a manufacturing plant). It ensures real-time control and coordination.

Human-Machine Interfaces (HMIs): HMIs connect to DCS and PLCs, allowing operators to interact with the system. They provide basic controls, visualizations, and alarms. Effective communication within this zone ensures smooth operation and timely responses to process changes.

Level 3: Operations Management

This zone focuses on managing production workflows. Here are the components:

Manufacturing Operations Management (MOM) Systems: These systems handle production operations, including scheduling, resource distribution, and tracking work orders. They ensure efficient utilization of resources.

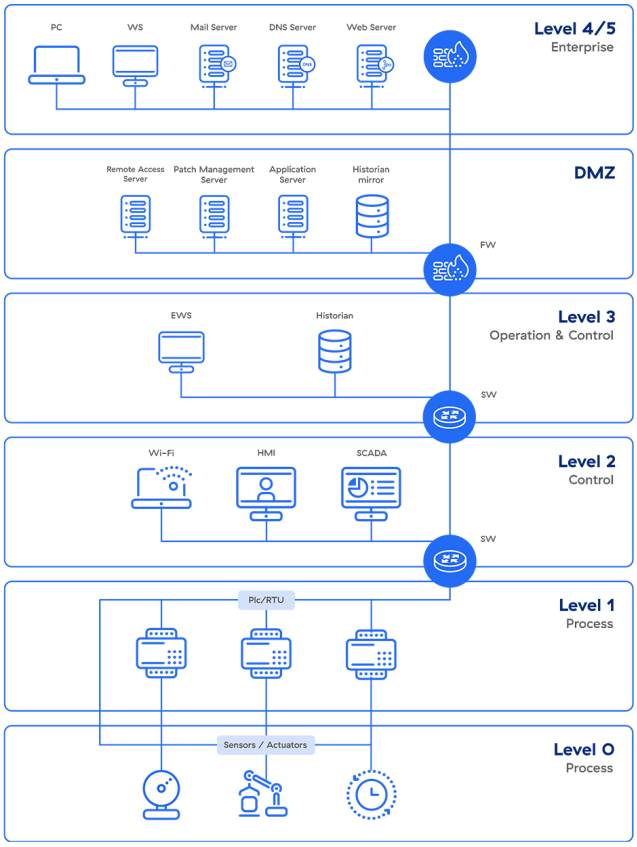

Figure 2.3: A basic example of the Purdue model [3]

Manufacturing Execution Systems (MES): MES gather real-time data from different sources (such as sensors, PLCs, and operators) to optimize production processes. They provide visibility into production status, quality, and performance.

Data Historians: These systems store historical process data. In modern solutions, they also perform contextual analysis, identifying patterns and anomalies. This data is valuable for process optimization and troubleshooting.

Disruptions in this zone can have significant consequences, including economic losses, safety risks, and operational downtime.

Level 3.5: Demilitarized Zone (DMZ)

The DMZ [24] is a buffer between the IT (Information Technology) and OT networks. Key features include:

Security Systems: Firewalls and proxies are deployed to prevent lateral movement of threats between IT and OT systems.

Bidirectional Data Flows: As automation increases, bidirectional data exchange between IT and OT becomes essential. The DMZ facilitates controlled communication.

Risk Considerations: While IT-OT convergence offers advantages, organizations must manage cyber risks effectively.

Level 4/5: Enterprise Zone

In this zone, we move into the standard IT network where critical business functions occur. Here's what you'll find:

Enterprise Resource Planning (ERP) Systems: These systems manage various aspects of the business, including production schedules, material usage, shipping, and product levels. That helps optimize resource allocation and streamline operations.

Business Orchestration: The IT network coordinates manufacturing operations, ensuring alignment with overall business goals.

Potential Consequences: Disturbances here can lead to lengthy downtime, economic losses, and risks to critical infrastructure and revenue.

Level 5 or Cloud level does not officially exist in the Purdue or IEC 62443 [25] reference architecture. However, we added it to illustrate two ongoing trends

2.2 Cyber Attacks On ICS

Manufacturing plants use the ICS for their operation and use an automated system for production output. These methods enable consistent manufacturing, which leads to higher product quality. They can save costs by lowering energy use, reducing manufacturing carbon footprints, and reducing manpower requirements

ICS have traditionally prioritized reliability, durability, economic efficiency, and safety. Regardless, the pervasive computerization and automation of these systems have led to increased integration and interdependencies, presenting unforeseen disturbances. This includes complex control loops, cascading failures, and malware propagation as a consequence of enhanced efficiency through technology adoption. Over recent decades, industrial plants have undergone significant modernization, shifting from relay panels to embedded computers and from analog sensors to IP-enabled smart transmitters with extensive communication capabilities, con-

figurations, and even web server functionalities for remote maintenance access. While security measures aim to limit vulnerabilities, continual innovation by vendors expands the scope for remote exploitation of physical processes and equipment.

The widespread "cyberfication" of industrial systems has raised concerns about vulnerabilities to both random cyber failures and deliberate security attacks. Embedded computers now facilitate precise control over physical applications to achieve specific outcomes, yet they also enable malicious instructions that can cause unintended actions. This phenomenon underscores the concept of cyber-physical attacks, where software instructions, devoid of physical force, can manipulate physical systems to malfunction or cause damage. The distinguishing characteristic of cyber-physical attacks lies in their potential to cause tangible physical harm, marking a departure from conventional cyber attacks.

These technologies, however, constitute a substantial attacking risk. Because they are automated, they do not require continual human involvement. While this increases the system's efficiency, it also introduces the possibility of damage.

Skilled attackers may compromise ICS systems, risking human safety and causing substantial disruption in society. When an attacker issues a command, the physical processes they control might be disrupted, resulting in disturbance and even injury to someone.

Companies that rely on these technologies worry about both data theft and financial failures.

As improved algorithms and device communication improve these systems, increased network interconnectivity equals increased cybersecurity concerns.

Over the past ten years, the number of cyberattacks against ICS systems has increased, followed by an increase in ransomware attacks.

Meanwhile, Kaspersky's analysis shows that more than 40% operational technology (OT) devices will be targeted by malicious cyber activities by 2022 [26].

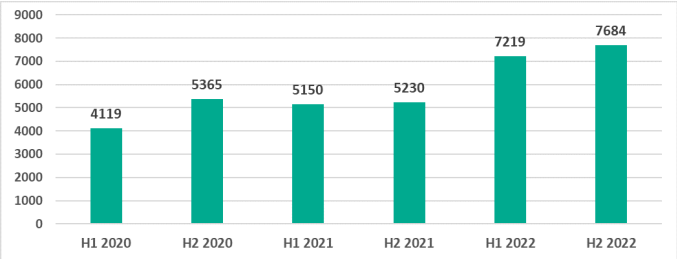

Figure 2.4: Number of malware families blocked on ICS computers

Source: Kaspersky.

2.2.1 Industrial Control Systems Threat Landscape

Within the context of information technology, attackers typically look for data, such as intellectual property or specific records, as their ultimate goal. Malicious activities in the OT (Operational Technology) area are particularly concerning because the objective of the attackers is to create a tangible effect in the physical world.

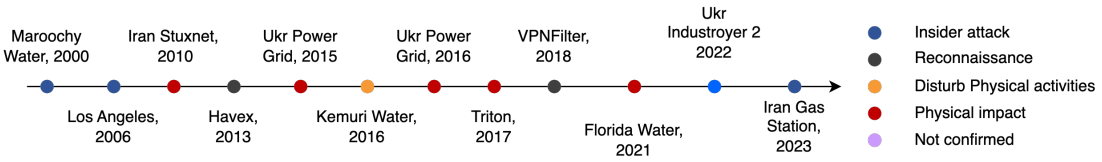

Figure 2.5: Historical time of cyber-physical attacks

The problems are made worse by the fact that the majority of control equipment, systems, and protocols either lack essential security measures or are not adequately set with them. As a precautionary security measure, it is feasible to separate OT equipment from high-risk networks and the Internet via the OT DMZ. Nevertheless, many industrial settings lack a network layer or have a network layer that is severely constrained in its functionality. Consequently, the vulnerability of the OT zone to malware penetration or attacks enabled by humans remains significant. Figure 2.5 depicts the chronological sequence of publicly documented attacks that resulted in concrete effects.

2.2.2 Remote Access Attacks

A remote access attack takes place when a hacker accesses an ICS remotely [27]. SCADA systems are one of the most standard techniques for hackers to get remote access to ICS systems. In this scenario, the attacker gains access to the SCADA system's internal network (LAN) and then accesses the SCADA through the LAN. This may be accomplished by using a machine running specialized software called remote access software. This program enables remote access to the LAN and SCADA system.

2.2.3 Wireless Network Attacks

Wireless networking has grown in popularity since it eliminates network connection difficulties. Wireless networks enable trouble-free connectivity. Because connected devices in open areas are vulnerable to reproduction and physical assaults, security has become a top priority

these days. Lightweight security systems are necessary to authenticate connected devices and secure industrial data due to resource restrictions. Wireless network attacks are cyber threats that target wireless communication technologies [28] including Wi-Fi, Bluetooth, and other wireless protocols. These assaults may compromise wireless network security and privacy, possibly resulting in unauthorized access, data theft, or network interruptions.

2.2.4 Malware and Virus Attacks

Another method for hackers to get access to ICS systems is through virus and malware-based assaults, such as the Stuxnet Malware [6, 29] or the Triton malware [30]. This tactic is similar to a computer virus or malware assault in that hackers often attempt to mislead someone into installing an infected file on a computer linked to the ICS. Once downloaded, the malware can spread to other systems linked to the ICS via network shares, portable drives, and other methods computers use to transfer data.

2.3 ICS Attack Lifecycle

An attacker aiming at a remote process may not initially have full knowledge of the procedure and the methods to control it. A malicious attack may need to progress through multiple phases [31] before the malicious goals can be shown in Figure 2.6. Perfect understanding is not gathered and the attacker may need to turn back to earlier levels or recursively repeat their activities at the same level.

2.3.1 Access

Access [32] is the phase that closely matches classical IT hacking. Typically, the attacker requires functional code within the target's network to affect the process, and so has to find a way of entry.

A process network usually interacts with the network of the company and a field network, as well as multiple regulatory connections that are relevant to the handling of any potentially hazardous substances utilized. In addition to the continuous flow of data from the control network to corporate and third-party systems, process control systems share many of the same requirements as IT systems. The network must receive patches and anti-virus updates. Control rules must also be transmitted from the network to field devices. Regulatory data needs to be transmitted to many entities. Sometimes, it is necessary to transmit the data instantaneously.

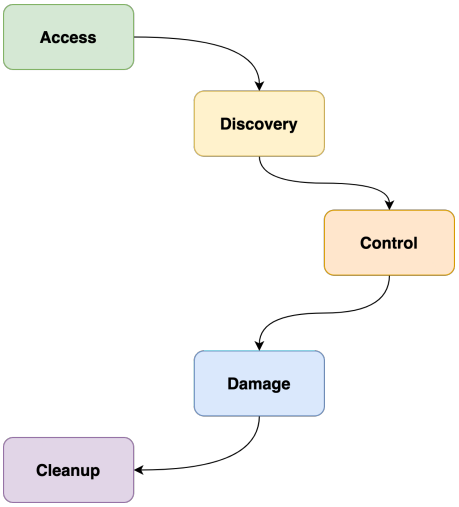

Figure 2.6: Stages of cyber-physical attacks

These data flows have the ability to serve as access points into the process network. This stage is very similar to the process of entering any other network.

2.3.2 Discovery

It concerns discovering [33] knowledge about a manufacturing plant via documentation. Without precise knowledge, it is uncertain that an attacker can achieve more than a distraction. Blindly trying to destroy a process by overheating industrial components, for example, will usually just result in exercising the emergency shutdown logic and the pressure relief valves.

The attacker must so recreate the design of the industrial plant and how it performs its functions. This is the most difficult and time-intensive of the steps. There are various data sources that describe the procedure. The attacker may first investigate general information on the physics, kinetics, and thermodynamics of the physical processes of interest. This can be done by conferring public information as well as confidential material of process design enterprises.

Operator interfaces are supposed to be human-readable. Regulatory filings will describe the inner workings of safety or environmental-related subsystems. Engineering diagrams may be kept in change management systems electronically so changes in the physical process can be matched to changes in the control logic. This element of the discovery step may potentially involve reconnaissance.

2.3.3 Command and Control

The process is not intended for the attacker. Every component of the process has a normal range and a possible range. Adjusting one aspect of the process for malicious reasons may have negative consequences on other sections of the process. The control [34] stage investigates what each actuator does and what side effects are feasible. It may be easy to turn off a component for example a pump, but the side consequence is that pressure builds up quickly in an upstream pipe. Not every action can be taken at every step of the process. Instructing a breaker to close while a line is charged may be prevented via an interlock. The attacker may need to hack an embedded controller to circumvent that interlock. The control step also incorporates the examination of timings. If the harm occurs in seconds, a safety shutdown minutes later will not stop the attacker.

Some parts of control can be researched statically, but other sections must be investigated dynamically on the process (process reconnaissance). No graphic will ever be comprehensive enough to accurately predict the transit time of a disturbance down a pipe to the accuracy needed to set up a resonance between two pumps. Since that data must be retrieved from the live process, this is a perfect chance for the defenders to notice the attackers.

2.3.4 Damage

After learning the process and gaining control over it, the attacker needs to choose the exact techniques needed to accomplish their goals. Multiple conflicting scenarios could exist. The attacker must come up with an objective (a measurable standard) for choosing among them. Rebounding a few components off the floor until they break the target facility may seem like an appropriate choice, but it is important to consider that the economic consequences of this action may be significantly less severe. Some of the damage [35] scenarios are straightforward such as operating rotating equipment at its natural frequency at which a mechanical resonance and equipment vibrations may occur. Vibrations significantly reduce the expected equipment life span and lead to equipment breakage.

2.3.5 Cleanup

The cleanup [36] phase involves changing procedures and files to create a forensic trace that leads analysts to wrong assumptions. The idea is to blame the attack on operator error or equipment failure, rather than a cyber incident. An example of a cleanup phase would be to show

the operator a process out of control, making her take a particular action. When investigators asked the operator if she was manually manipulating the process when it malfunctioned.

2.4 ICS Honeypots

2.4.1 Honeypot Characteristics

A Honeypot is a security technique or device used to attract and identify possible attackers, malware, or unauthorized network activity. It acts as a decoy or trap, mimicking vulnerable systems, services, or applications to divert the attention of hackers. Honeypots' major goal is to gain information about the tactics, methods, techniques, and procedures used by attackers, as well as to monitor and mitigate network risks. Authenticity, targetability, cost, and risk are all important considerations. A real honeypot looks like the features of a real computer system. More realistic characteristics result in a more complicated honeypot that gathers more comprehensive data. If the attacker avoids honeypots, they are ineffective. As a result, a honeypot must be targetable, with a significant enough presence to draw the attacker's notice. The initial development, deployment, maintenance, operation, data storage, and analysis of the data generated are all important factors.

2.4.2 Level of interaction

Low-Interaction Honeypots: These are intended to mimic the behaviour of vulnerable systems without revealing real services. They are simple to set up and provide no threat, but they only give limited interaction data. Because of their limited capabilities, low-interaction honeypots might not provide attackers with a comprehensive environment in which they can execute their attacks. Attackers may not be able to complete all of the stages of their attack or may be unaware that they are attacking a fake system. The main advantage of the low interactive honeypots is they can not fully compromise the real system and this decreases the possibility of real honeypot and network damage. Gaspot [37], which was supplied as an example of a low-interaction honeypot, is the finest example. It is written in Python and has few configuration choices, making it a lightweight and cost-effective solution for security monitoring.
High-Interaction Honeypots: These types of honeypots mimic real systems and services, making them appear stronger to attackers. While they provide useful interaction data, the installation is risky since they run real, possibly insecure software. High-interaction honeypots

Interaction	Uses	Advantages	Disadvantages
Low	Production	Real-Time Threat Detection Large presence potential	Security Risks Resource Intensive Complexity
High	Research	Isolation Minimal Impact Learning and Skill Development	Higher cost Risk of Failure Deployment Complexity

Table 2.1: Advantages and disadvantages of Production vs Research Honeypots [5]

offer attackers a very realistic environment in which to execute a wide range of attacks. This provides critical information about an attacker's methods, strategies, and targets. this type of honeypot offers valuable insight into the attacker's tactics, methods, techniques, and intentions but it comes with a high level of risk, as the fully compromised high-interaction honeypot can be used to launch the leveraged attacks within the network. A good example of a high-interaction honeypot is the ICSpot [15]. Organizations that opt to deploy high-interaction honeypots should do so after carefully considering the risks and putting good protections in place to avoid key systems from being compromised.

2.4.3 Types of Honeypots

Research Honeypots: Honeypots for research and data collection: These are commonly employed in academic or industrial research environments.
Production Honeypots: Production honeypots are often employed in operational contexts to detect and respond to real-time threats.

3

Related Research

3.0.1 Gaspot

Gaspot [38] was developed at TrendMicro and presented at Blackhat 2015. The minimal interaction honeypot was inspired by assaults on gas station control devices. It simulates basic services provided by these devices and logs all interactions. The honeypot is straightforward, responding to queries with randomized values within reasonable limits. Attacker interactions and origins were studied after deployment in several nations.

3.0.2 Conpot

Conpot [39] is a server-side Industrial Control Systems honeypot that is continuously updated and easy to deploy, change, and expand. Conpot is not designed to simulate processes or devices, despite its extensibility. The system is defined in XML files, and protocol emulation is done in Python. The Modbus_tk library is used in the examples to emulate device memory. Conpot's functionality has been enhanced to simulate a smart meter. Modbus, SNMP, and a static HTTP HMI are all available on the spoof smart meter. Because the honeypot's primary purpose is to provide intelligence to a production system, any interaction is viewed as a sign of compromise. As a result, minimal effort is put into process or device emulation.

3.0.3 GridPot

In the GridPot [40] project, Conpot has been developed to replicate electrical grid components. GridPot uses GridLAB-D to create a model of an IEEE power distribution test case, which is then used to create a process model.

3.0.4 CryPLH

CryPLH [13], or the Crysys PLC Honeypot, is a low-interaction honeypot that is actively being developed to replicate a Siemens Simatic 300 PLC. It employs a central configuration file to simplify and end users configuration by simulating the exposed HTTP, HTTPS, SNMP, and Siemens SIMATIC STEP7 (carried out using the ISOTSAP protocol) configuration interfaces on a simple Ubuntu Linux VM. Both the HTTP/S and ISOTSAP interfaces have logins that will not accept any username/password combinations, and the viewable web portal will not alter to reflect the PLC's environment.

3.0.5 SCADA HoneyNet

Cisco Systems published the Supervisory Control and Data Acquisition (SCADA) HoneyNet [41] Project in March 2004 as the first low-interaction CPS-focused honeypot. HoneyD, Arpd, Snort, and Tripwire were used to simulate multiple hosts on a network. It was designed to emulate FTP, HTTP, Telnet, and Modbus for Schneider PLCs, as well as FTP, HTTP, SNMP, and S7comm for Siemens PLCs. The honeypot makes no attempt to mimic the behavior of a process. The project has been abandoned.

3.0.6 Virtual ICS Honeypots in a Box

Virtual ICS Honeypots in a Box [42] is a honeypot that is based on the MiniCPS framework capable of emulating the software-defined network. The system supports process and device simulation, similar to this work, however, the device simulation is limited to emulating device services and logic and does not include actuation fingerprints.

3.0.7 HoneyVP

HoneyVP [43] architecture identifies three independent and basic components: virtual, physical, and coordinator. Finally, a local-remote collaborative ICS honeypot system is tested for

practicality and effectiveness. HoneyVP offers a cost-effective option for ICS security researchers, making honeypots more appealing and capturing physical interactions.

3.0.8 Snap7

Snap7 [44] is an open-source communication library that facilitates communication with Siemens S7 PLCs. It provides a robust and efficient way to interact with Siemens S7 series PLCs using the Siemens S7 protocol, which is a proprietary communication protocol designed for Siemens automation systems. Snap7 is highly regarded for its ease of use, cross-platform capabilities, and comprehensive functionality, making it an attractive option for developers and engineers working in industrial automation.

One of the key features of Snap7 is its ability to perform a wide range of operations with Siemens S7 PLCs, including reading and writing data to and from PLC memory, monitoring and controlling PLC status and managing PLC blocks. This versatility makes Snap7 an invaluable tool for applications requiring real-time data acquisition, remote monitoring, and control of industrial processes. The library supports various data types and structures, allowing for seamless integration with complex automation systems.

3.0.9 HoneyPLC

HoneyPLC [45] is a malware-collecting honeypot with a high level of interaction. It supports a wide range of PLC models and suppliers. HoneyPLC demonstrates a high level of stealth, as made evident that it is correctly identified as actual devices by numerous widely used reconnaissance tools, including Nmap, Shodan's Honeyscore, the Siemens Step7 Manager, PLCinject, and PLCScan. HoneyPLC was deployed on Amazon AWS and recorded a huge number of unique interactions over the Internet, demonstrating not only that attackers are targeting ICS systems, but also that HoneyPLC can effectively engage and fool them while collecting data samples for further study.

3.0.10 ICSPot

The ICSPot [15] identifies a significant flaw in current ICS honeypots: a lack of reliable physical process modelling. ICSpot seeks to overcome this shortcoming by presenting a more accurate representation of these processes within the honeypot. This means that ICSpot was not created from the bottom up, but rather includes existing implements and technologies to deliver a

more complete and realistic ICS honeypot solution. This is a practical strategy since it makes use of existing knowledge and tools in the ICS security sector. also, the author describes the validation of the honeypot in both a local Internet Exchange Point (IXP) and an Amazon Web Services (AWS) server, this approach to deploying honeypots in real-world scenarios, including on-premises and cloud environments then actively collecting interaction data for 30 days.

Honeypot	Open Source	Physical Interaction	Network Simulation	HMI	PLC registers	ICS protocols	Entry Point
Gaspot [38]	✓	✗	✗	✗	✗	—	Internet
Conpot [39]	✓	✗	◐	✗	✗	Ethernet/IP, S7,Modbus, BACnet	Internet
GridPot [40]	✓	✓	✗	✗	✗	Modbus,S7	Internet
SCADA HoneyNet [41]	✓	✗	◐	✗	✗	Modbus	Internet
Virtual ICS Honeypots in a Box [42]	✗	✓	✓	✓	✓	Modbus,DNP3, EtherNet/IP	VPN, Internet
HoneyVP [43]	✗	◐	✓	✗	✓	S7	Internet
HoneyPLC [45]	✓	✗	◐	✗	✗	S7	Internet
ICSPot [15]	✓	✓	✓	✓	✓	S7,Modbus	Internet

✗= Not supported ◐= Partially supported ✓= Fully supported

Table 3.1: Literature comparison of honeypots

4

Problem Statement And Limitations

The primary purpose of traditional network-focused honeypots, as well as current CPS honeypots was to mimic the types of protocol peculiarities that are fingerprinting applications like Nmap. On the other writing, ICS honeypots should supply supplementary data derived from the associated physical system. This auxiliary data includes the capacity to compare the CPS's state from one moment to the next for consistency (i.e., using the physics of the process and sensors), as well as the capability to look for unusual actuation fingerprints on particularly connected devices. Attackers can easily tell if they are in a honeypot if the process physics or device actuation times are unrealistic. However, existing ICS honeypot performances have significant limitations in capturing data on the latest and most sophisticated attack techniques. Specifically, we have identified the following limitations:

4.1 Problems And Limitations of Current ICS Honeypots:

Ensuring ICS is vital, and honeypots play a unique role in identifying and learning possible risks. However, implementing honeypots in ICS contexts poses specific challenges:

1) Emulation of Industrial Protocols
2) Components of Imitating Control Systems (ICS)
3) Behavioral Accuracy
4) Inclusion of ICS-specific Artifacts
5) Network Topology
6) Simulation of Physical Processes

Table 4.1: Limited Realism

4.1.1 Limited Realism

The current ICS honeypots don't have the realistic emulation of the industrial process, they may not correctly mimic the complex architecture and large set of devices and protocols used in the real ICS eco-system [15] which makes them less effective in getting the attackers. also response to attackers like a real honeypot should respond to attacks in the same manner that a real system would. This includes activities such as issuing alerts, recording incidents, and, if necessary, altering management processes.

Realism in the context of ICS honeypots describes how closely the honeypot environment resembles a real-world ICS ecosystem. 1 Since realism impacts an ICS honeypot's potential to draw in and spot real threats, establishing it is vital to its usefulness. Key components of ICS honeypot realism include the table 4.1:

Emulation of Industrial Protocols:

Industrial communication protocols used in operational environments should be imitated by realistic ICS honeypots [46]. This covers numerous protocols, including DNP3, OPC, and Modbus. Because accurate emulation guarantees that the honeypot runs like a real ICS system, potential attackers will find it more enticing [46].

Components of Imitating Control Systems (ICS)

Programmable Logic Controller (PLC) Emulation:

1. Software Emulation: To replicate the operations of real PLCs, virtualized or emulated PLCs are frequently deployed. The operation and logic properties of these simulated PLCs are identical to those of the actual ones. [3]

2. I/O Signal Simulation: To model how sensors, actuators, and extra components communicate in an ICS system, modelled PLCs provide simulated inputs as well as outputs signals.

3. Human–Machine Interfaces (HMIs):

    - User Interface with Graphics (GUI) Emulation: Software that mimics the appearance and feel of authentic HMIs is used to imitate the HMI components. It involves simulated controls, alerts, and process visualization.
    - Emulation of Processes: Through interaction between the HMI and the replicated processes, intruders might assume they are in charge of actual industrial systems.
    - Systems Simulation: Dynamic System Modelling: Certain Industrial Cybersecurity (ICS) honeypots don't imitate individual pieces; they also replicate full industrial processes. This requires dynamic modelling processes to replicate the activities of real processes that are governed by ICS systems.

Behavioural Accuracy

While attaining behavioural precision in Industrial Control Systems (ICS) honeypots is vital for efficiently simulating real-world scenarios and recognizing probable risks, this method has several limitations and difficulties as follows.

1. Industrial Process Complexity
2. Energetic Character of ICS Environments
3. Variation in ICS Protocols
4. Tailoring of Attacks
5. Restriction of Resources
6. Adjusting to Changing Dangers
7. A Legislative and Ethical Perspective
8. Insufficient Standardization

ICS honeypot solutions are continually being studied and improved to overcome these restrictions. To maximize the realism and effectiveness of ICS honeypots while minimizing their drawbacks, a multidisciplinary strategy encompassing cooperation between cybersecurity specialists, ICS professionals, and service vendors is needed.

4.1.2 Protocol Complexity

The complicated structure of industrial protocols within the framework of Industrial Control Systems (ICS) honeypots provides various limits and difficulties:

1. Diverse Protocols
2. Customization of Protocols
3. Standards for Dynamic Protocols
4. Practical Issues with Message Traffic Compatibility
5. Insufficient Standardization
6. Protection via obscurity
7. Cryptological Difficulties

Accurate reproduction of industrial protocols in ICS honeypots is difficult due to their complexity. It is challenging to correctly emulate the complexities of a large array of data transfer protocols, including DNP3 or Modbus. Customized or proprietary protocols restrict coverage by making emulation considerably more difficult. Keeping honeypots up to date is tough due to the dynamic nature of protocol standards and the continual growth in technology specifications. Other challenges include coping with inconsistent protocols, overcoming interoperability difficulties across varied systems, and attaining appropriate message flow patterns. Complicating issues are the dependence on security via obfuscation and cryptography problems. emulation is made more difficult by the range of ICS assaults and the absence of standardized attack strategies for protocols. To get over these limits, further research and cooperation are needed to make the honeypot framework more adaptable so it can adjust to the evolving ICS protocol environment and offer plausible threat scenarios.

4.1.3 Data Privacy Concerns

ICS suffer a variety of critical issues connected to data privacy, mostly because operating data is vital and security compromises might have serious effects. One problem is that to efficiently monitor and track industrial operations, ICS systems must gather and understand sensitive data. Putting strong security procedures in place to safeguard this data while nevertheless delivering the essential functionality is the tricky part. It can be challenging to find a balance between privacy of information and operational efficacy, particularly in light of the possibility that key infrastructure may be subject to cyberattacks [47]. Addressing these problems needs a coordinated effort between cybersecurity experts, ICS professionals, and solution suppliers to continuously improve the effectiveness of ICS honeypots in detecting and mitigating threats.

The absence of standardized safety frameworks created especially for ICS systems is another drawback. In contrast with existing data protection legislation in the larger IT industries, industry-specific standards for ICS could not adequately encompass the intricacies of data privacy. This weakness can make it more difficult to design standardized protocols that ICS controllers can adhere to, which makes it difficult to provide reliable and efficient data privacy protections in a range of industrial environments. To overcome these challenges, industry-specific privacy standards must be defined cooperatively, and cutting-edge security technology must be integrated to preserve private data without interfering with important industrial activities.

4.1.4 Accuracy of Gathered Information

One key problem in ICS is to maintain the integrity of obtained data because of the possibility of data breach and tampering. One challenge in ICS systems is the multitude of data sources, such as sensors, actuators, and control systems. It is difficult to verify the validity and correctness of data from these many sources, and attackers may utilize flaws to modify the data [48].

The fact that industrial processes are dynamic gives rise to yet another limitation. Variations in trends in data could be produced by sudden occurrences or by abrupt changes in operating parameters. It becomes difficult to discern between purposeful tampering and genuine adjustments, which might lead to false alerts or miss true security risks. The problem is further enhanced by the absence of specified procedures for information checking and validation in ICS settings. In contrast to typical IT systems, which have established cryptographic techniques and integrity checks, ICS could not have standard operating procedures to assure the reliability of the data that is gathered. This restriction may make it more difficult to design reliable systems for spotting and quickly handling information security breaches [48].

The incorporation of sophisticated anomaly detection algorithms, safe data transfer protocols, and defined integrity testing procedures suited to the special needs of ICS is essential to overcome these limits. Establishing standard methods and rules that increase the accuracy of data gathered in important industrial processes needs cooperation between cybersecurity specialists, ICS specialists, and regulatory bodies.

4.1.5 Limitations in Incident Response

Problems with incident response in ICS bring considerable constraints, mainly because ICS contexts are diverse. The multidisciplinary aspect of dealing with incidents [49] in ICS, which needs coordination between both information technology (IT) and OT (Operational Technology) teams, is just one of its constraints. It may be difficult to coordinate responses to security issues when two traditionally different locations are bridged, which can cause delays.

The relevance underpinning real-time operations in ICS is the source of another constraint. Operations in factories may be impacted by incident response procedures including fixing security flaws or disconnecting affected systems. It may be difficult to strike a balance between the necessity of responding promptly and the desire to cause as little disturbance as possible, especially when dealing with complicated and dynamic cyber-attacks. A consistent and effective strategy is made increasingly tougher by the absence of specified processes for responding to incidents designed especially for ICS systems. Specialized incident response frameworks that take into account both cyber and physical parts are crucial since ICS accidents typically involve unique concerns, such as the risk of injury to people.

Also, incident identification and response are complicated by the limited access to ICS systems. It's possible that traditional security solutions developed for IT environments don't offer an in-depth understanding of the

specific equipment and communication methods utilized in ICS. This restriction may make it more difficult to swiftly identify and prevent security risks.

Defined incident response mechanisms for ICS must be created and used by the industry as a whole to meet these limits. It is necessary to incorporate specific technology for ICS detection and reaction to problems and to offer educational courses that strengthen the competencies of cybersecurity specialists working in the industry. To tackle the particular problems presented by events in industrial contexts, joint initiatives including cybersecurity specialists, ICS vendors, and regulatory bodies can aid in accelerating the establishment of efficient incident management systems.

4.1.6 Scalability

Because industrial settings are large and varied, scalability is a major difficulty in the wider environment of ICS. The wide diversity of systems and devices found in ICS connections, from PLCs to actuators and sensor technology, presents one issue. The sheer quantity and variety of these variables can impose stress on conventional ICS systems as industrial sites increase, making it tough to scale security measures efficiently.

Scalability challenges are also brought on by the heterogeneity of ICS components and the range of communication protocols deployed in distinct industries. It is difficult to integrate security solutions that can react to the unique needs of every device and protocol, and as the number of devices rises, scalability becomes an increasingly critical consideration.

The resource limits that are commonly present in ICS settings make scaling concerns worse. Industrial systems may have restrictive memory, bandwidth, or processing capability, which makes it difficult to apply scalable safety measures while interfering with the functioning of vital activities.

Furthermore, scalability challenges are posed by the interconnection between IT and OT (Operational Technology) platforms. It becomes more difficult to ensure that safety controls scale smoothly across these related areas as organizations mix IT and OT systems to increase efficiency.

The establishment of adaptable security designs, the effective deployment of safeguarding solutions, and the incorporation of advancements that can adjust to the growing and rising complexity of industrial settings are all important for tackling the scalability issue in ICS. ICS specialists, cybersecurity experts, and solution suppliers must work together to build flexible safety measures that can successfully protect critical systems from evolving threats.

5 Architecture and Design

In this chapter, we will illustrate the components of the VIRTUEPOT and the design features and address the limitations we found earlier. These include true OT software and monitoring tools enabling us to record real-time incidences within the honeypot. We also show the steps we implemented to convince possible threat actors to break into our system. Figure 5.1 shows the components of VIRTUEPOT, which is developed as a High-interaction honeypot.

5.1 The VIRTUEPOT architecture is composed of the following components, namely

5.1.1 HoneyD Framework

Honeyd [50], a tool for simulating computer systems on the network layer, can successfully simulate large-scale network and system services. also has a system logging module that can log the iteration with the system, the HoneyD's personality engine allows us to simulate the TCP/IP Stack so that we can use this tool to fool the attackers with reconnaissance tools like Nmap, example when Nmap try to read the fingerprint of a system the HoneyD will repose with the fake spoofed fingerprint information so it will help us to hide the original information of the system and keep attackers engaged with the honeypot. Honeyd's Subsystem virtualization enables Virtuepot to provide network service extension by switching network traffic to

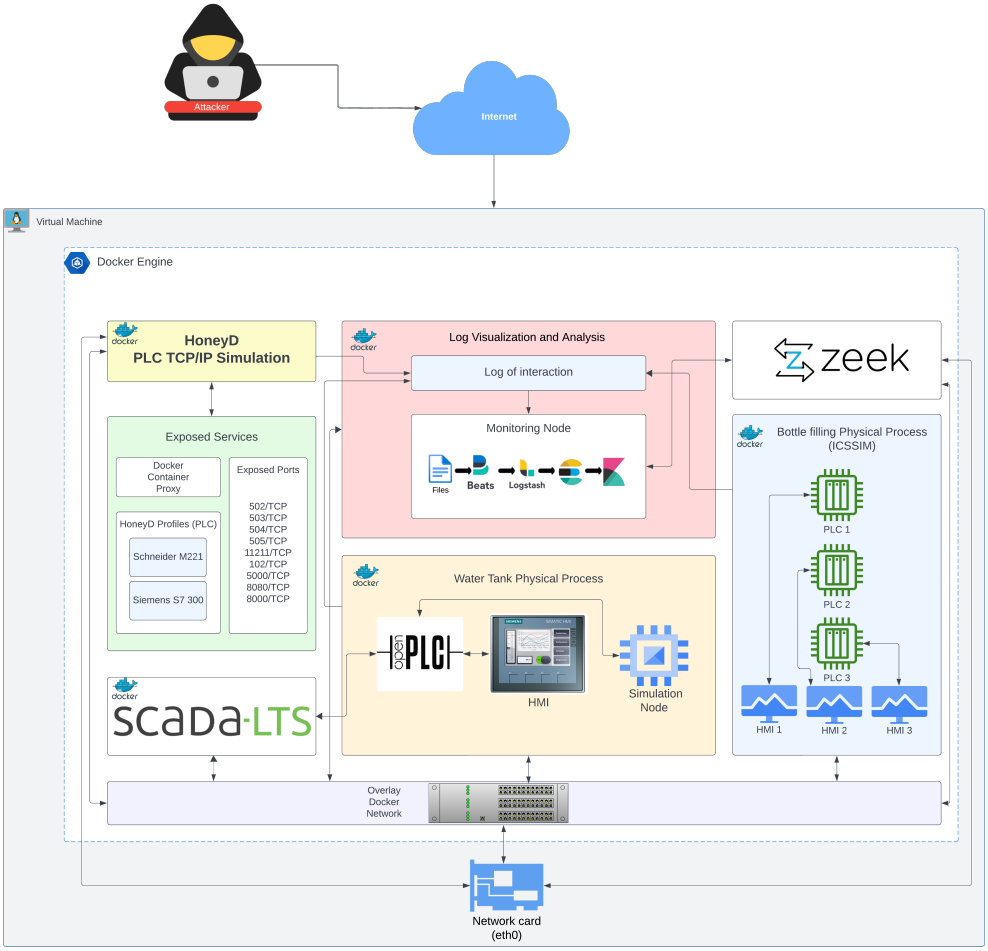

Figure 5.1: Virtuepot architecture.

a suitable simulation, such as an OpenPLC server. To forward TCP and UDP requests, the Subsystem Virtualization component interfaces with the Network Services. component.

- **PLC Profiles** The PLC profiles are configured using the *honeyd.conf* file it allows us to emulate various services and operating systems to attract the attackers and able to monitor and log their activities. using the we can emulate the hosts, network configurations, services, and personality including the os fingerprint, the personality engine modifies the protocol headers of every outgoing packet to correspond with the characteristics of the specified operating system, causing the honeypots network layer to behave as defined by the personality. The framework refers to Nmap's [51] fingerprinting data for a personality's TCP and UCP behaviour, and Xprobe's [52] fingerprinting database for a personality's ICMP behavior.

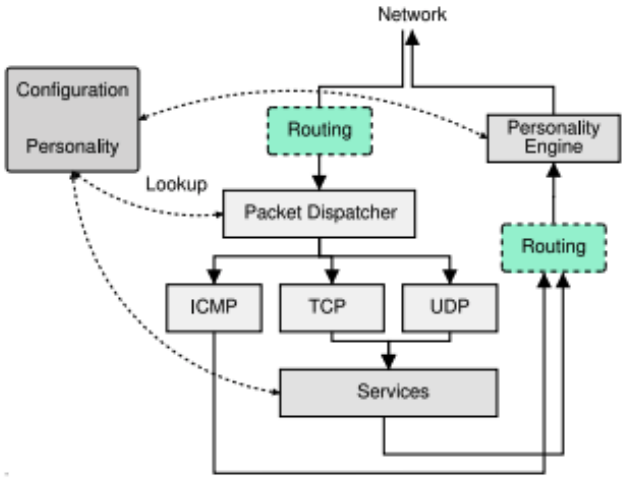

Figure 5.2: Honeyd's architecture [4].

The 5.2 Shows the overview of the architecture, the incoming packets will come to the protocol handler and it replies with configuration, and packets are routed to the relevant protocol handler. The particular services for TCP and UDP accept new data and, if necessary, give responses. The personality engine changes all outgoing packets to mimic the behavior of the configured network stack. Honeyd only uses the routing component for simulating network topologies.

5.1.2 SCADA-LTS

SCADA-LTS [53] is a GUI-based multi-platform open-source software that enables the creation of SCADA system applications Scada-LTS comes with everything you need to get started quickly: communication protocols, a data collecting engine, alarms and events, an HMI builder, and much more. The software design is written in Java, and the server may run on any platform (PC/Mac/Linux) Figure 5.3 shows the graphic view of the SCADA-LTS. The user interface is accessible via a typical web browser, and no client installation is required. and supports protocols, including Modbus TCP/IP, OPC DA 2.0 ASCII Serial, IEC 101, DNP3 and File readers, which are supported by the program.

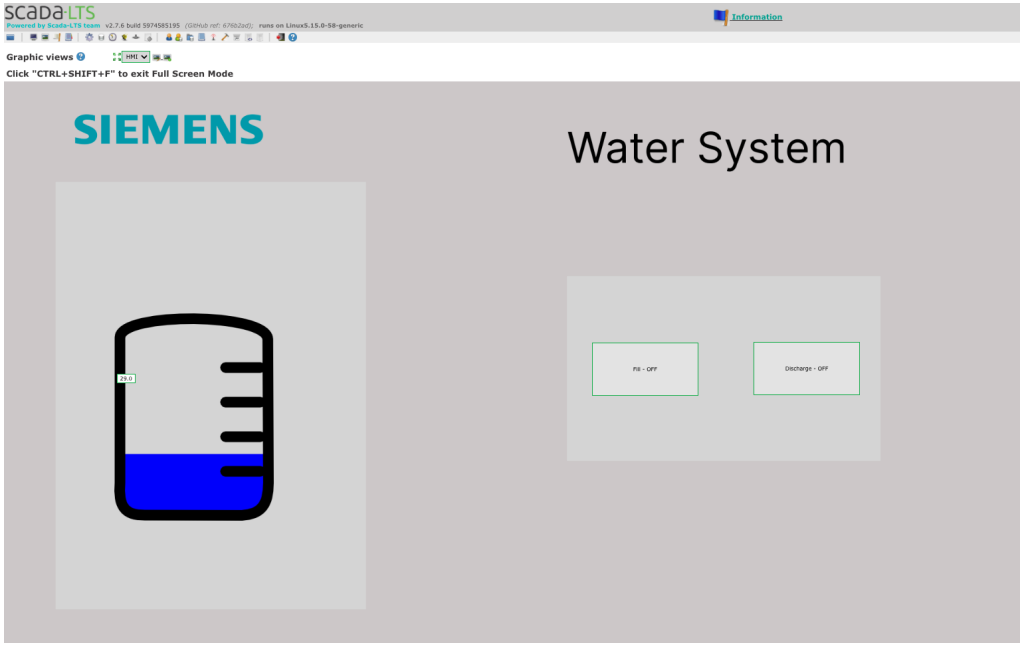

Figure 5.3: SCADA-LTS GUI.

5.1.3 ICSSIM Framework

In the proposed system we are using the ICSSIM [54] Framework to build the virtual ICS environment, this framework is capable of simulating the control system components and the communications between them and the simulated components are deployed on the docker-machine, this framework reduced the developing time and this testbed is expandable, adaptable, repeatable, low-cost, and comprehensive. This framework uses the Purdue Enterprise Reference Architecture [55] as shown in Figure5.4 and the architecture includes the five-layer tiers like sensors and actuators in Tier 1, Basic controls in Tier 2 this includes the PLCs, Supervisory Control in Tier 3 this includes Engineering Workstations, Historian, and HMI's, In Tier 4 we have the Operating Demilitarised Zone (DMZ), and Tier 5 is the enterprise zone, which serves as a host for non-ICS devices and servers as well as a data supervisor.

5.1.4 OpenPLC

OpenPLC [56] is an open-source Programmable Logic Controller(PLC) software that supports various protocols such as Modbus, Profibus, Ethernet/IP, and more. Using a graphical or text-based programming interface, OpenPLC programmers can create logic programs to

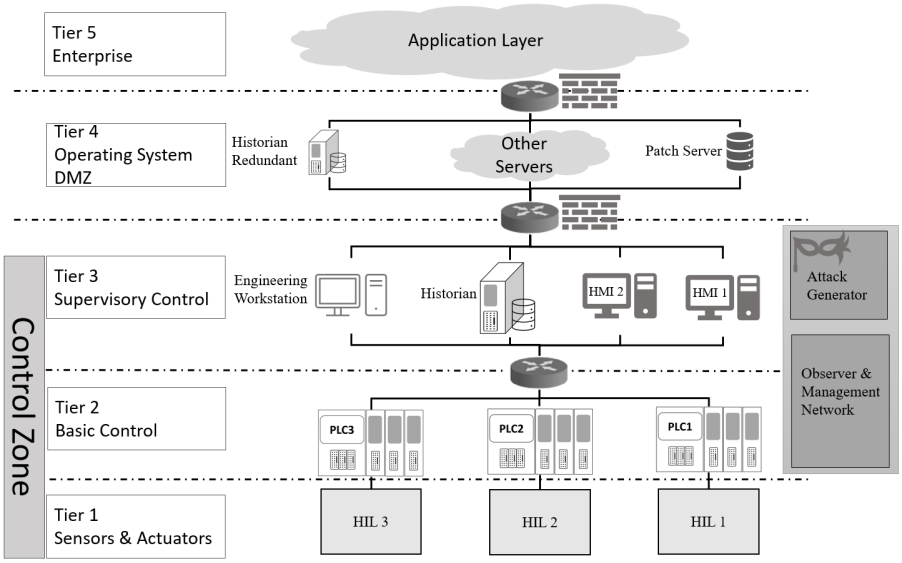

Figure 5.4: ICSSIM architecture.

Source: ICSSIM.

manage industrial operations. Standard programming languages such as Ladder Logic, Structured Text, and Sequential Function Chart (SFC), Function Block Diagram (FBD)are typically supported.

This compatibility ensures that users familiar with traditional PLC programming can easily transition to OpenPLC without a steep learning curve. Additionally, the platform is designed to run on a variety of hardware, from low-cost microcontrollers like Arduino and Raspberry Pi to more powerful industrial computers. This hardware flexibility makes OpenPLC an attractive option for both educational purposes and professional applications.

5.1.5 Zeek

Zeek [57], originally known as Bro, is an open-source network security monitoring and traffic analysis application. Lawrence Berkeley National Laboratory researchers created it to support network administrators and security professionals in analyzing network traffic and detecting possible security risks. Zeek can able to continuously collect and analyze network data, offering insights into network activities, traffic patterns, and potential security risks. It interprets and records a wide range of network protocols, including HTTP, DNS, FTP, SSH, and many more. This data can be extremely useful for network troubleshooting and security investigations. Zeek creates wide log files that may be analyzed further and integrated with other secu-

rity products and SIEM (Security Information and Event Management) systems Like Elastic Stack.

5.1.6 ELK Stack

The ELK Stack [58] is a collection of open-source tools for log and data analysis. "ELK" refers to three essential components of this stack Elasticsearch, Logstash, and Kibana. In addition, we are using Beats also.

- Elasticsearch is a search and analytics engine that is distributed and RESTful. It is intended for storing, searching, and analyzing enormous amounts of data in real time. Elasticsearch is at the heart of the ELK Stack, acting as a log data storage and search engine.
- Logstash is a data processing pipeline that collects, analyses and enhances data from a variety of sources. It can gather and analyze log data from diverse sources, convert it to a standard format, and then transmit it to Elasticsearch for indexing and storage. Logstash also supports several plugins for interacting with a variety of data sources.
- Kibana is a data visualization and exploration tool that offers a web-based user interface for querying and visualizing Elasticsearch data. It enables users to obtain insights into their data by creating custom dashboards, doing ad-hoc searches, and generating charts and graphs shown in Figure **??**.
- Elastic created the Beats lightweight data shippers. They can gather and transfer many data kinds to Elasticsearch and Logstash. Filebeat (for log files), Metricbeat (for system metrics), and Packetbeat (for network traffic analysis) are a few examples.

Because the ELK Stack is commonly used for log and event data analysis, it is well-suited for log management, security information and event management (SIEM), and other data analytics and visualization activities. It's used in a variety of sectors to centralize logs, obtain insights into system and application performance, diagnose problems, and monitor network security. The stack's open-source transparency and adaptability make it a popular choice for organizations of all sizes, while Elastic's commercial products give extra enterprise-level capabilities and support.

5.1.7 HMI

Virtuepot has a custom-developed Django-based HMI interface that shows the details of the physical process running on the PLC, it shows the water tank level the flow of the water line

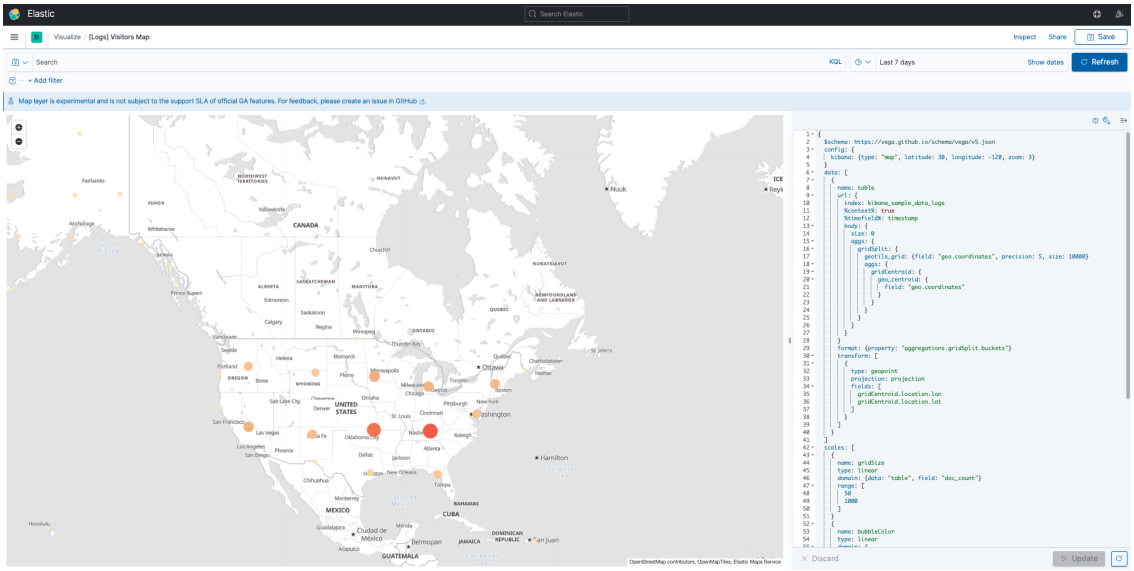

Figure 5.5: ELK Stack: Kibana

ongoing water, and outgoing water volume, which is exposed on port 5000, refer the Figure 5.6.

5.1.8 Simulation Server

The simulation server is the module used to simulate the physical process controlled by the PLC and it builds a simulated environment so that real process data is simulated to mimic the process of the real system.

5.1.9 Docker

Docker and Docker overlay network are important components of the VirtuePot system design, allowing for the quick installation and communication of various components inside the honeypot environment. These tools serve as the foundation for honeypot deployment, isolation, and communication. Their contribution is critical to the development of an efficient platform for mimicking operational technological systems and attracting possible threat actors.

With its combination of HoneyD, SCADA-LTS, ICSSIM, OpenPLC, Zeek, ELK Stack, HMI, Docker, and Simulation Server, VirtuePot's architecture provides a complete and realistic environment for mimicking industrial control systems. This interactive honeypot aims to

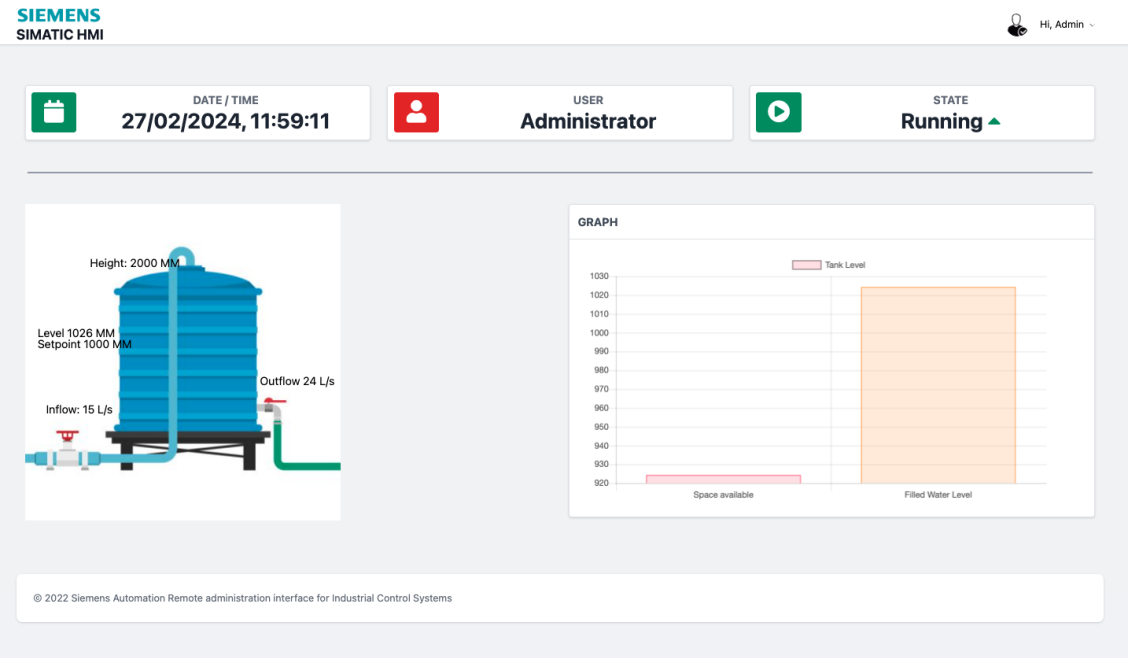

Figure 5.6: HMI

attract potential threat actors, monitor their activity in real-time, and give useful insights into attacker techniques and behaviours, each component is containerized using Docker technology. In our honeypot, there are two simulations, one simulation on the OpenPLC which is running the water tank control process and consists of the PLC process, HMI for controlling the process, and also SCADA system is connected to manage the physical process. Another is simulating the bottle-filling factory control process, The control process is divided into two primary hardware zones, each operated by a separate PLC, PLC-1, and PLC-2. PLC-1 controls the water tank and valves. PLC-2 controls the conveyor belts, which change out filled bottles for empty ones, and it has 3 HMI's for controlling the tank input output valve, tank level, conveyor belt engine mode, and bottle level. Then we integrate all the PLCs with HoneyD for the appropriate fingerprint to reply to the Nmap scan. at this point the Nmap will confirm to the attacker that he is dealing with the real PLC, not the honeypot also able to initiate the connection with the PLC memory blocks, first, the connection is handled by the Honeyd and forwarded to the relevant PLC Server, parallel the Honeyd is logging all interactions, including source IP addresses and memory block requests, Finally, the attacker can able to inject malicious PLC ladder logic program and Modbus packets to interrupt communication. and able to inject some random packets into the network to interrupt or control the operation of the

PLC. Then Zeek is running in the same network interface so that we can get detailed insights into network traffic and security events.

Network The following diagram 5.7 illustrates the honeypots shown network output.

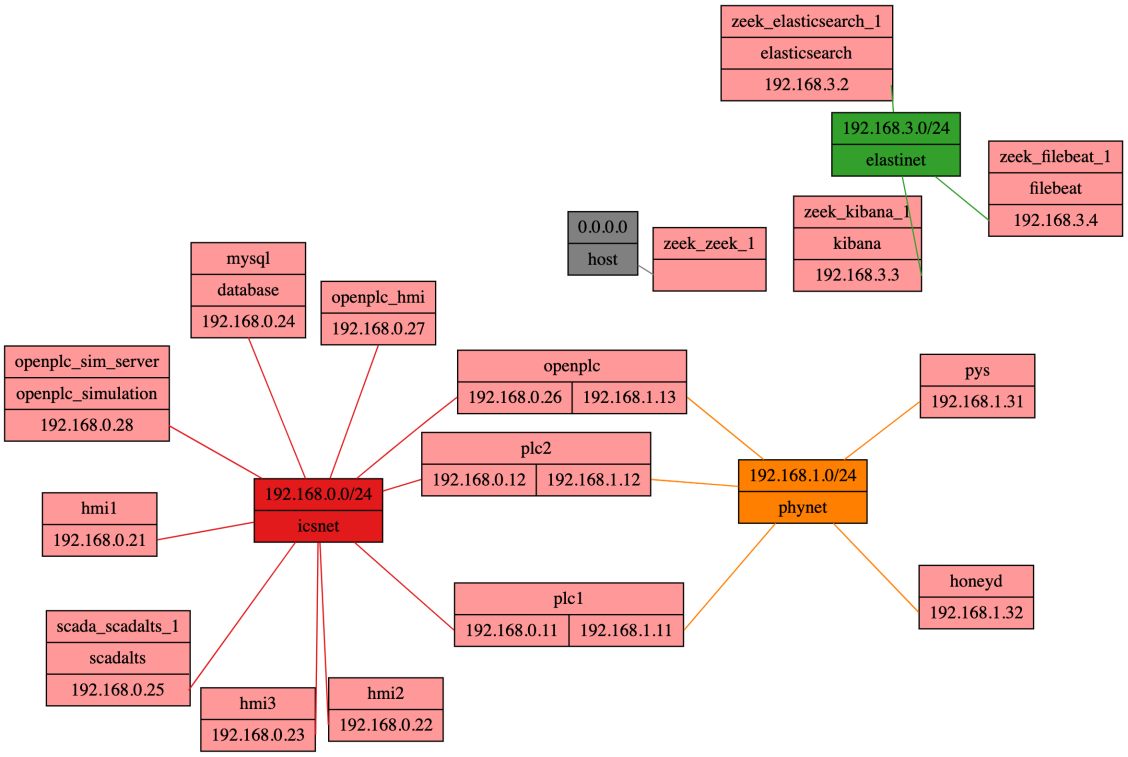

Figure 5.7: Honeypot Network

As we use Zeek with the ICSNPP [59] package, it can help in the detection of suspicious or unauthorized activity inside your ICS network. This includes unusual protocol use, data exfiltration, and other possibly harmful behaviour. Modbus, DNP3, S7, and other industrial protocols are supported also by ICS-specific network traffic. This provides you with information on the communication patterns and data flows in your ICS environment. To centralize monitoring and analysis of ICS network data, Zeek's logs can be combined with SIEM (Security Information and Event Management) solutions, to the ELK Stack discussed before.

6

Implementation

We planned to implement the VirtuePot based on the Modbus [60], S7comm, DNP3, and other protocols. Our honeypot is based on the OpenPLC [56], Honeyd, and ICSSIM [54] framework as described in the chapter 5 this will help in the implementing in the virtual instead of the physical ICS hardware and all the components of our model will be running on the Docker container so it will help to isolation each honeypot service could be contained within its own Docker container, making it impossible for an attacker to pivot to other services or the host computer if one honeypot is compromised. for the PLC simulation, we planned to use a low-interactive honeypot like Honeyd to provide the personality engine for the simulation of the TCP/IP stack for the target devices like PLC, and HMI, proxy the traffic to the physics-aware high-interaction framework like OpenPLC so this allows us to maintain the connection and establish a physical interaction. network scanning tools traffic like Nmap will be handled by the Honeyd and using subsystem virtualization feature it will be integrated with OpenPLC. The HMI will be implemented using the Django framework so it will be connected to the OpenPLC for the interactions and physical process controls, it will be listening on the standard TCP port 5000 and the Scada-LTS interface will be connected as well for the physical process controls. The discovery of over a thousand exposed PLCs on the internet, as shown in the figure6.1, raises important security concerns. PLCs are critical components in industrial and manufacturing settings, controlling various processes. When they are accessible from the internet, it poses a significant security risk.

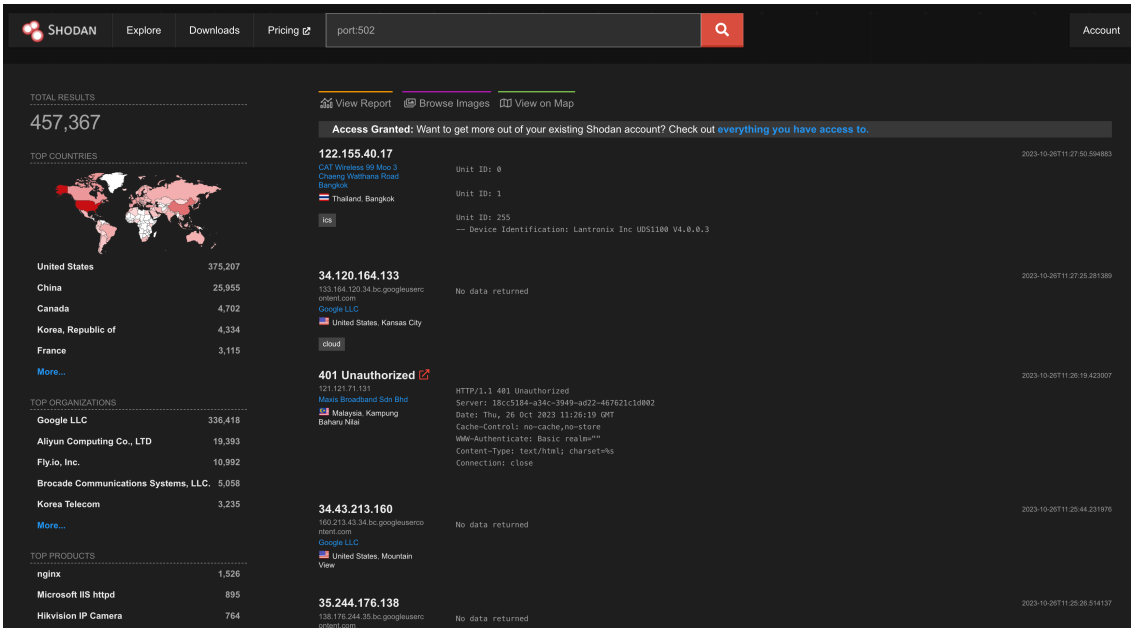

Figure 6.1: Discover PLCs exposed on the internet using Shodan

6.0.1 Honeyd Framework and Configuration

To Integrate the Honeyd framework with sophisticated service simulations we need to use the various capabilities of Honeyd These capabilities include its personality engine, configuration files, and subsystem virtualization.

Before Honeyd can start its processes, a configuration file is a requirement. This configuration file is a straightforward text document that sticks to context-free grammar, describing command options and syntax. The significance of these configuration commands lies in their capacity to customize Honeyd's behaviour.

Here are some of the important commands:

Personality setting This code assigns to change the personality of a honeypot, letting it emulate various systems and services.

Network Space With this code, can select the network space, including IP address ranges, subnets, and routing details.

Port Behavior The configuration file specify how ports behave, which ports are open, and how the honeypot replies to incoming traffic.

```
create schneider_m221
set schneider_m221 personality "Schneider Electric TSX ETY programmable logic
                                        controller"
set schneider_m221 default tcp action reset
add schneider_m221 tcp port 502 proxy 0.0.0.0:502
set schneider_m221 default icmp action open
set schneider_m221 ethernet "28:29:86:F9:7C:6E"
bind 192.168.1.168 schneider_m221

create siemens_s7_300
set siemens_s7_300 personality "Siemens Simatic 300 programmable logic controller"
set siemens_s7_300 default tcp action reset
add siemens_s7_300 subsystem "/usr/share/honeyd/s7commServer" shared restart
set siemens_s7_300 default icmp action open
set siemens_s7_300 ethernet "00:1C:06:0C:2E:C6"
bind 192.168.1.169 siemens_s7_300

create allen_bardley_plc5
set allen_bardley_plc5 personality "Allen-Bradley PLC-5 programmable logic
                                        controller"
add allen_bardley_plc5 tcp port 503 proxy 0.0.0.0:503
set allen_bardley_plc5 default icmp action open
set allen_bardley_plc5 ethernet "00:00:BC:18:51:A2"
bind 192.168.1.170 allen_bardley_plc5

create simens_s7_1200
set simens_s7_1200 personality "Siemens Simatic 1200 programmable logic controller"
add simens_s7_1200 tcp port 504 proxy 0.0.0.0:504
set simens_s7_1200 default icmp action open
set simens_s7_1200 ethernet "8C:F3:19:D9:A6:11"
bind 192.168.1.171 simens_s7_1200
```

Honeyd makes effective use of these commands to customize the honeypot's personality, subsystem virtualization, port behaviour, and network space. The above example in the code shows the format of a configuration file. It begins by creating a base subsystem and adding the necessary subsystem. Then, a clone is made to create the precise virtual honeypot example siemens_s7_300. The personality is set to emulate the PLC, the honeypot is attached to an IP address and the manufacturer's MAC address is used to mimic the defined PLC.

Fingerprint Database Integration

To keep multiple PLC prototypes, we've taken the PLC profiles provided by the HoneyPLC [11] project as an example and created a new PLC profile as shown in the code 6.0.1. These profiles enable Honeypot to answer fingerprint requests from various PLC models, including Schneider M221, Allen Bradley PLC5, Siemens S7-300 and S7-1200. These fingerprints are combined into Honeyd's fingerprint database, providing a wide range of replies to potential attackers.

6.0.2 Integrating OpenPLC with Honeyd

A key challenge in implementing a VIRTUEPOT is routing network requests appropriately. We use Honeyd to handle requests from monitoring tools, while those acquired via ICS network protocols, such as register manipulation, are controlled by OpenPLC. To achieve this, we've integrated OpenPLC with Honeyd using Honeyd's subsystem virtualization component. This integration allows for a seamless flow of network requests, ensuring that attackers encounter a realistic environment.

6.0.3 Enhancing Interaction with HMI

In complement to PLC containers, our honeypot has HMI containers. These containers give the user an interface for interacting with the honeypot. In our honeypot, Django-based HMI is running on port 5000, and Scada-LTS, an open-source SCADA interface, is installed on these containers. It can interact with various PLCs, making it a versatile and powerful tool for managing and monitoring the honeypot. like software PLCs, HMI interfaces reach with built-in communication support and do not require a dedicated broker to handle communications. Scada-LTS's web interface is accessible on port 8000 and can be accessed through a web browser. It's worth noting that one version of Scada-LTS is exposed to known vulnerabilities (CVE-2021-26828, CVE-2022-41976) that allow an authenticated attacker to execute arbitrary code and privilege escalation.

6.0.4 Log and Monitoring Dashboard

This dashboard offers a comprehensive view of the honeypot and devices within the network, and it provides real-time visualizations of attackers' activities, offering insights into attack patterns, the origin of attacks, and the type of malware used. so we use Elasticsearch ELK Stack

and as mentioned in the chapter5 (Elasticsearch, Logstash, and Kibana) is a perfect solution for a tracking and analytics platform on honeypot data. all the components of the honeypot are integrated with the logging functionalities and we configured the honey to collect all the integration logs all the information gathered by the Zeek the honeyd is stored in the file beats integrated with the Elastichsearch for further processing and embedded with the Kibana for visualization.

6.0.5 Simulating the physical processes

Python Script is used to model, simulate, and analyze cyber-physical systems. It's extensively accepted in both industry and research for its power to imitate the behaviour of physical systems. By utilizing pyModbusTCP we develop models that simulate the behavior of the plant, enabling investigation and understanding of its physical processes.

6.0.6 Experiential approach

The main goal of our honeypot is to have attacker interaction should be a similar experience to a real ICS environment, the small details of the honeypot are very important because the Honeypot interfaces are essential for spoofing the attacker or public scanners like Shodan and censys.

Reconnaissance programs like Shodan use a "HoneyScore" evaluation algorithm to analyze whether an internet-exposed machine is a honeypot described in the chapter 7.

The honeypot implemented for this experiment delivers high interaction, which is more deceiving than low interaction. It also permits cost-efficient deployment on an on-prime without the extra complexity of a high-interaction honeypot.

7 Evaluation and Results

7.1 Evaluation of VIRTUEPOT Covertness against Nmap and Shodan

Experimental Focus

Covertness: The core analysis centres on how well VIRTUEPOT maintains its disguise as a legitimate system, avoiding detection by Nmap scanning and the Shodan Honeyscore algorithm.
Deployment Comparison: Data from VIRTUEPOT instances deployed in a VSIX IXP (on-premise) is compared with a cloud deployment to identify any differences in attacker behaviour.

The experiment described includes measuring the covertness of the VIRTUEPOT by testing it against two well-known tools Nmap and Shodan's Honeyscore. The purpose is to analyze how well VIRTUEPOT retains undetected as a honeypot under different scanning tools.

Nmap: This utility is often used for network discovery and OS fingerprinting. It's frequently used by security professionals and attackers alike to probe networks, detect open ports, and identify the OS systems running on target workstations. By putting the VIRTUEPOT to Nmap scans, you're assessing how well it can avoid discovery as a honeypot by analyzing how it responds to Nmap's probing tactics.

Shodan's Honeyscore: This is a feature within Shodan's API used to identify potential honeypots by awarding a score between 0.0 and 1.0 to an IP address. A high score close to

1.0 suggests a high possibility of the host being a honeypot. The criteria employed by the Honeyscore algorithm include assessing elements such as the number of open ports, services operating (particularly if they don't match the typical environment, such as ICS running on Cloud, and default settings that match those of known honeypots. Shodan-identified honeypots commonly have Shodan-attached tags like "cloud" or "hosting" and "honeypot" combined. Shodan utilizes a proprietary technique to produce a value from 0.0 to 1.0 to determine honeypots (honeyscore is near 1.0) from legitimate systems (honey score is close to 0.0), in our approach, Shodan does not scan our IP addresses for honeyscore so the information is not available.

The objective of this experiment is to discover how efficiently VIRTUEPOT maintains its covertness when facing inspection from these tools. To pass this test, VIRTUEPOT should preferably appear as a legitimate system rather than a honeypot. If Nmap and Shodan's Honeyscore fail to decisively identify it as a honeypot, it suggests that VIRTUEPOT has a fair level of stealth and can effectively hide its true identity, hence capturing important interaction data without raising suspicions.

7.2 Results

In this experiment, we deployed two identical honeypots at different locations: one at an Internet exchange point (VSIX) and the other in the cloud (DigitalOcean). These honeypots are designed to engage attackers more effectively than low-interaction honeypots while being cost-efficient and simpler to deploy.

The honeypots run on actual machines with Ubuntu 22.04 LTS and Docker. One is set up in the VSIX IXP building in Padua, Italy, and the other in a public cloud in Frankfurt, Germany. This setup allows us to compare the effectiveness of each deployment.

We collected data for 61 days, from November through December 2023, monitoring activity on several ports: 502, 503, 504, 102, 11211, 5000, 8080, and 8000, described in table 7.1. Data collected for these ports helps us evaluate and validate the efficacy of our deployment techniques.

The process of collecting data involves the following steps:

1. **Filtering Raw Traffic:** We start by feeding raw traffic data into Zeek, which filters out irrelevant information, allowing us to focus only on pertinent data.

2. **Identifying Scanners:** We then identify scanners by removing duplicate IP addresses and correlating them with owner names using Greynoise. Greynoise [61] helps categorize IP

Port	Description
502	Modbus (OpenPLC)
503	Modbus (ICSSIM)
504	Modbus (ICSSIM)
102	Siemens S7
11211	Memcached
5000	HMI
8080	OpenPLC Server
8000	Scada-LTS

Table 7.1: Port Description

addresses that scan the internet, reducing the time spent on irrelevant or benign activity and allowing analysts to focus on emerging and targeted threats.

3. **Extracting Scanner IPs:** We manually observe and compile a list of scanner IP addresses.

4. **Filtering Scanner Traffic:** Using the list of scanner IP addresses, we filter out all packets associated with scanners and crawlers.

5. **Analyzing Remaining Traffic:** Finally, we examine the remaining traffic for attack patterns, specifically focusing on ICS protocols including Modbus and S7Comm.

In order to derive insights from the VSIX IXP deployment data, it is essential to compare it with the data obtained from a cloud deployment. Hence, a cloud deployment is implemented as a reference point. The process of gathering and analyzing data in a cloud deployment is not similar to that of an on-premise deployment, which means the VSIX IXP.

7.2.1 Modbus Attack Functions

In this study, attacks are defined as requests for PLC memory to read, stop, start, and write commands for single and multiple registers. Specifically:

- PLC Stop and Start Commands: These are considered attacks because they can disrupt the availability of the PLC.
- PLC Memory Read Multiple Requests: These are typically part of the reconnaissance phase of an attack or a type of Denial of Service (DoS) attack.

Figure 7.1 illustrates the attack functions executed by attackers using the Modbus protocol.

Figure 7.2 shows the locations of the attackers' IP addresses, with red marks representing attacks on the VSIX deployment and blue marks representing attacks on the cloud deployment.

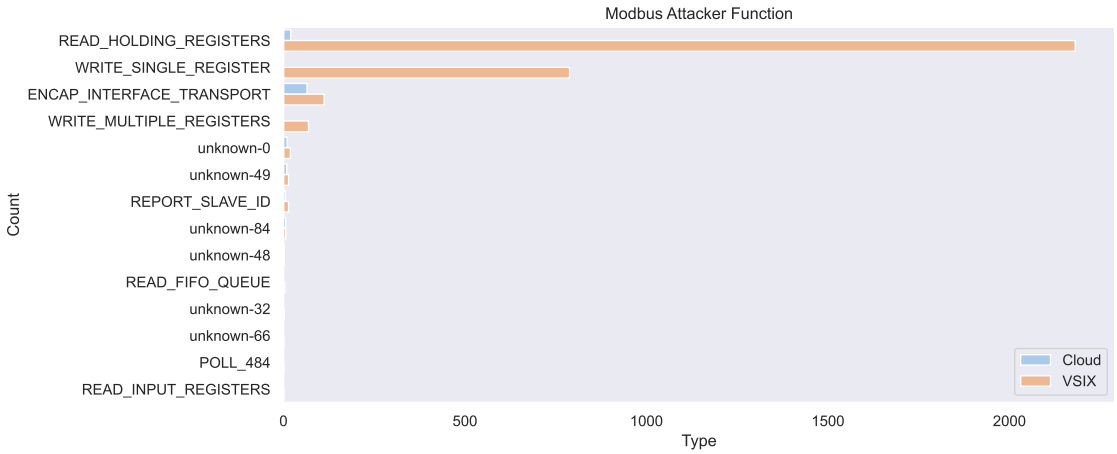

Figure 7.1: Modbus Attack Functions

Figure 7.2: Modbus Attacker locations

7.2.2 Interactions Origin

We recorded the highest number of scans originating from the United States, China, Germany, Brazil, and the Netherlands. However, it's important to understand that these scans might not

actually come from these countries. Attackers often use VPNs to mask their real IP addresses, making it look like their traffic is coming from a different location. Therefore, the true origin of the scans might be hidden behind these VPNs.

Origin: The following diagram 7.3 illustrates the honeypots traffic from different contries.

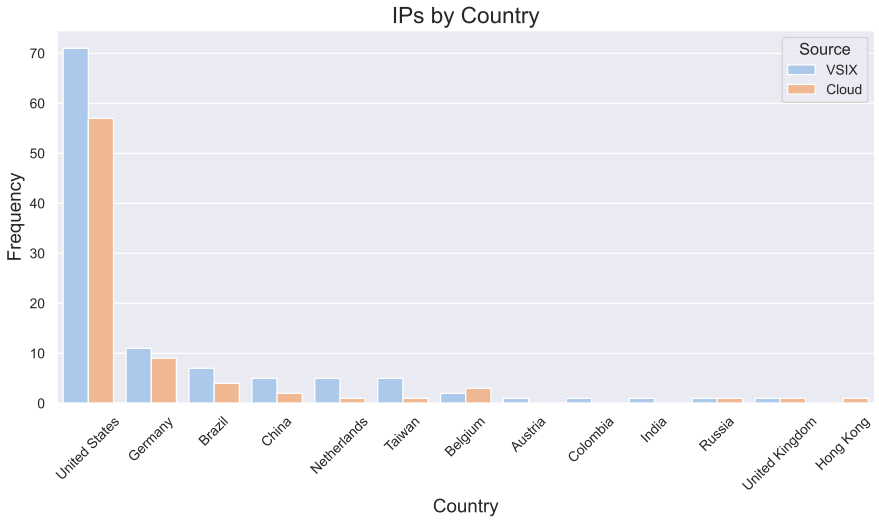

Figure 7.3: Number of unique IPs traffic to Modbus

Netblock holder

Identifying Attacking Hosts: Looking Beyond Netblock Ownership

When we examine the netblock (Autonomous System Number) owner, we can identify organizations that have a large number of attacking hosts. However, it's important to remember that these "attacking" devices are often compromised systems, not malicious actors themselves. This means that the true culprits are exploiting these compromised devices to carry out their attacks. Refer to figure 7.4 for more details.

7.2.3 Interaction Analysis

Figure 7.6 shows the distribution of various accessible services in both the Cloud and VSIX deployments. The most frequently accessed port is 5000, which operates the HMI and is connected to the PLC running the Modbus protocol. The HMI on port 5000 has proven to be

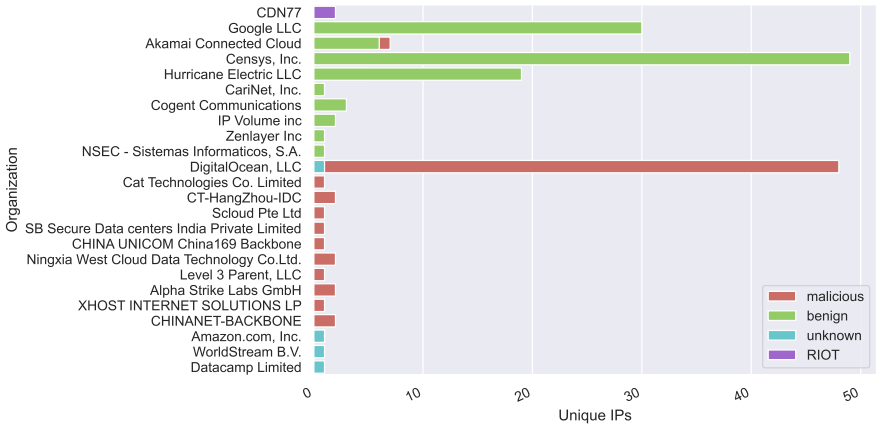

Figure 7.4: Number of unique IPs with traffic type

highly effective in attracting attackers. In the VSIX instance, we recorded interactions from 22,417 different IP addresses, while in the Cloud instance, we recorded interactions from 800 IP addresses.

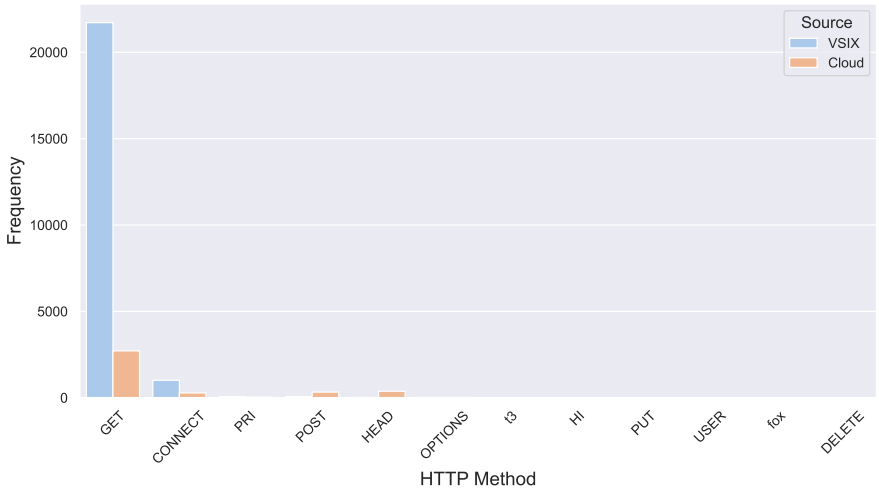

Figure 7.5: HTTP Methods used by the attackers

Conversely, we observed 3,500 unique IPs interacting with port 8080 in the Cloud deployment and approximately 5,000 in the VSIX deployment. While the high activity on port 5000

was anticipated due to its well-known status as a service port, the sheer volume of requests confirms the effectiveness of the HMI in engaging attackers. The HTTP methods used by attackers are shown in Figure 7.5.

Port 8000, which hosts the dashboard for controlling OpenPLC (allowing for changes in PLC programs, adding slave devices, and monitoring), was the third most visited port.

Port 502 was the fourth most exploited, indicating its popularity among attackers for interacting with the honeypot. This demonstrates Virtuepot's effectiveness in replicating the services of an original PLC emulated by OpenPLC.

Additionally, our analysis of the origin of the interaction IPs revealed that the sources of scanning were distributed similarly across both honeypot instances.

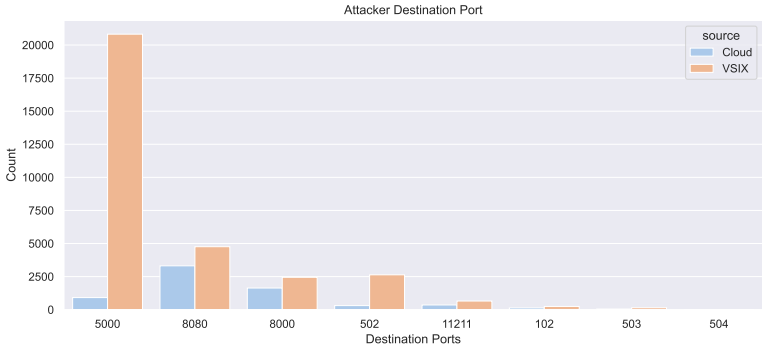

Figure 7.6: Services targeted

Using geolocation data, we created visual representations of the connection traffic for our honeypots. Figure 7.7 illustrates the connection traffic for the VSIX machine, while Figure 7.8 depicts the connection traffic for the Cloud machine. These figures show the number of connections originating from each country.

It's no surprise that the United States and China, which are frequently associated with a high volume of scanning activity, show a significant number of connections in both the VSIX and Cloud deployments. This highlights the global reach and widespread nature of scanning activity, with these two countries being prominent sources of traffic to our honeypots.

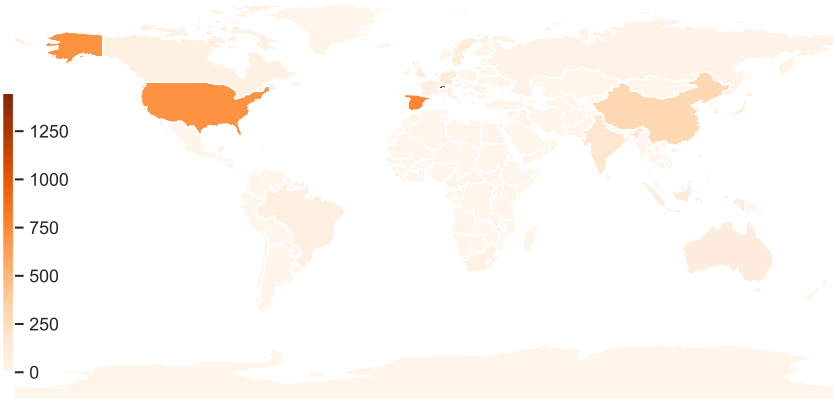

Figure 7.7: Countries with a large amount of noise in VSIX Machine

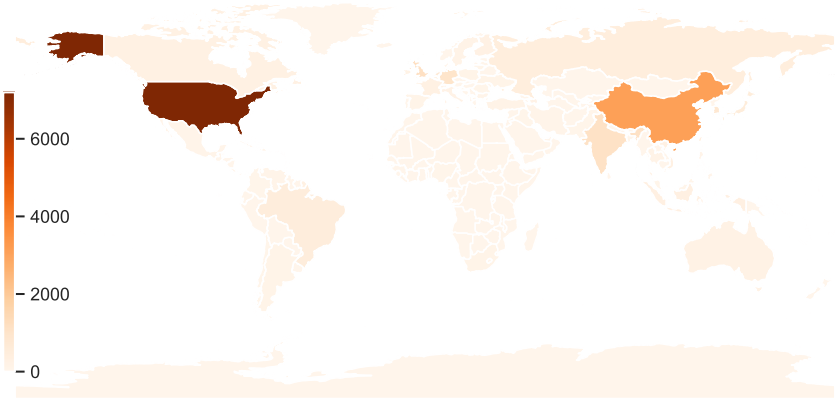

Figure 7.8: Countries with a large amount of noise in Cloud Machine

8 Conclusion

This comprehensive study investigated the covertness and attacker engagement capabilities of VIRTUEPOT, a novel honeypot designed to mimic ICS environments. Over a 61-day period, two VIRTUEPOT instances were deployed in contrasting environments – a VSIX IXP and a Public cloud (Digitalocean) provider – allowing for the examination of deployment-specific attack behaviour.

Findings indicate that VIRTUEPOT exhibits a promising level of Results, showing that VIRTUEPOT obtained a satisfactory degree of engagement during data collection, showing its ability to effectively reproduce an ICS environment. The honeypot's HMI component proved particularly effective in attracting and engaging attackers, suggesting that its realism is a major attraction. Analysis of attempted Modbus functions revealed a focus on both reconnaissance and potential ICS disruption, providing valuable insight into attacker goals within the simulated environment.

Attacker origins were primarily traced to the United States, China, Germany, Brazil and the Netherlands though the possibility of compromised systems masking true locations should be considered. Intriguingly, the VSIX IXP deployment attracted significantly better interactions than its cloud companion. The major difference between the Cloud and VSIX is the cloud deployment attracted S7 protocol traffic but in VSIX it is not found. This finding highlights how the perceived nature and proximity of a potential target can influence attacker behaviour.

These results contribute to the ongoing development of more deceptive ICS honeypots. VIRTUEPOT's design demonstrates the importance of ICS environment security, which is

crucial for managing critical infrastructure like power plants, water treatment facilities, and manufacturing processes, and faces unique security challenges. Further improvements to our work could be explored to enhance covertness, including the creation of more complex ICS capabilities and services that improve emulation quality. Additionally, longer observation periods and the integration of threat intelligence feeds would enrich the insights gained from VIRTUEPOT deployment.

Overall, this study underscores the value of honeypots as a proactive tool for gathering ICS-specific threat intelligence. By continuing to refine their design, we can significantly enhance our understanding of emerging attack vectors and strengthen the security posture of critical industrial control systems.

References

[1] "What"s the difference between OT, ICS, and SCADA? - kuppingercole.com," https://www.kuppingercole.com/blog/williamson/ot-ics-scada-whats-the-difference, [Accessed 02-06-2024].

[2] M. J. Keith Stouffer (NIST), "Capabilities Assessment for Securing Manufacturing Industrial Control Systems — csrc.nist.gov," https://csrc.nist.gov/pubs/pd/2017/03/09/securing-manufacturing-ics/final.

[3] Sectrio, "Threat Modeling Using the Purdue Model for ICS Security - securityboulevard.com," https://securityboulevard.com/2022/12/threat-modeling-using-the-purdue-model-for-ics-security-2/.

[4] N. Provos□, "A Virtual Honeypot Framework," https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/1.pdf, [Accessed 02-06-2024].

[5] N. Provos and T. Holz, *Virtual Honeypots - From Botnet Tracking to Intrusion Detection*. Addison-Wesley, 2008.

[6] B. Bencsáth, G. Pék, L. Buttyán, and M. Félegyházi, "The cousins of stuxnet: Duqu, flame, and gauss," *Future Internet*, vol. 4, no. 4, pp. 971–1003, 2012. [Online]. Available: https://www.mdpi.com/1999-5903/4/4/971

[7] Y. Hu, Y. Sun, Y. Wang, and Z. Wang, "An enhanced multi-stage semantic attack against industrial control systems," *IEEE Access*, vol. 7, pp. 156 871–156 882, 2019.

[8] S. Jaloudi, "Communication protocols of an industrial internet of things environment: A comparative study," *Future Internet*, vol. 11, p. 66, 2019.

[9] J. Franco, A. Aris, B. Canberk, and A. S. Uluagac, "A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems," *IEEE Communications Surveys and Tutorials*, vol. 23, pp. 2351–2383, 2021.

[10] G. Barbieri, M. Conti, N. O. Tippenhauer, and F. Turrin, "Assessing the use of insecure ics protocols via ixp network traffic analysis," *2021 International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–9, 2021.

[11] E. López-Morales, C. Rubio-Medrano, A. Doupé, Y. Shoshitaishvili, R. Wang, T. Bao, and G.-J. Ahn, "Honeyplc: A next-generation honeypot for industrial control systems," p. 279–291, 2020. [Online]. Available: https://doi.org/10.1145/3372297.3423356

[12] A. Jicha, M. Patton, and H. Chen, "Scada honeypots: An in-depth analysis of conpot," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, 2016, pp. 196–198.

[13] D. Buza, F. Juhász, G. Miru, M. Félegyházi, and T. Holczer, "Cryplh: Protecting smart energy systems from targeted attacks with a plc honeypot," in *International Workshop on Smart Grid Security*, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:18911942

[14] *SCADA HoneyNet Project: Building Honeypots for Industrial Networks*.

[15] M. Conti, F. Trolese, and F. Turrin, "Icspot: A high-interaction honeypot for industrial control systems," in *2022 International Symposium on Networks, Computers and Communications (ISNCC)*, 2022, pp. 1–4.

[16] C. Z. Kathryn Knerler, Ingrid Parker, "11 Strategies of a World-Class Cybersecurity Operations Center," https://www.mitre.org/sites/default/files/2022-04/11-strategies-of-a-world-class-cybersecurity-operations-center.pdf, [Accessed 02-06-2024].

[17] "GitHub - 0xnkc/virtuepot: A high interaction virtual ICS honeypot that simulates a PLC and provides physical process simulation. - github.com," https://github.com/0xnkc/virtuepot, [Accessed 02-06-2024].

[18] "virtuepot directory listing — archive.org," https://archive.org/download/virtuepot, [Accessed 04-06-2024].

[19] *International Journal of Recent Trends in Engineering and Research*, vol. 4, no. 2, p. 55–60, Feb. 2018. [Online]. Available: http://dx.doi.org/10.23883/ijrter.2018.4063.fr80v

[20] K. Stouffer, J. Falco, and K. Scarfone, "Nist special publication 800-82, guide to industrial control systems (ics) security," pp. 800–882, 01 2011.

[21] D. W. Pessen, "Ladder-diagram design for programmable controllers," *Automatica*, vol. 25, no. 3, p. 407–412, may 1989. [Online]. Available: https://doi.org/10.1016/0005-1098(89)90008-3

[22] T. Zabinski and T. Maczka, "Human system interface for manufacturing control — industrial implementation," in *3rd International Conference on Human System Interaction*. IEEE, May 2010. [Online]. Available: http://dx.doi.org/10.1109/hsi.2010.5514547

[23] T. J. Williams, "The purdue enterprise reference architecture," *In: Comput- ers in Industry 24.2–3*, Sept. 1994 pp. 141–158. issn: 0166-3615.

[24] T. D. Ashley, R. Kwon, S. N. G. Gourisetti, C. Katsis, C. A. Bonebrake, and P. A. Boyd, "Gamification of cybersecurity for workforce development in critical infrastructure," *IEEE Access*, vol. 10, pp. 112 487–112 501, 2022.

[25] D. Dolezilek, D. Gammel, and W. Fernandes, "Cybersecurity based on iec 62351 and iec 62443 for iec 61850 systems," in *15th International Conference on Developments in Power System Protection (DPSP 2020)*, 2020, pp. 1–6.

[26] "Industrial sector attacks on the rise: an annual overview by Kaspersky — kaspersky.com," https : / / www . kaspersky . com / about / press-releases / 2023 _ industrial-sector-attacks-on-the-rise-an-annual-overview-by-kaspersky, [Accessed 10-10-2023].

[27] I. Erkek and E. Irmak, "Cyber security of internet connected ics/scada devices and services," in *2021 International Conference on Information Security and Cryptology (ISCTURKEY)*, 2021, pp. 75–80.

[28] B. Tian, Y. Yao, L. Shi, S. Shao, Z. Liu, and C. Xu, "A novel sybil attack detection scheme for wireless sensor network," in *2013 5th IEEE International Conference on Broadband Network Multimedia Technology*, 2013, pp. 294–297.

[29] S. Al-Rabiaah, "The "stuxnet" virus of 2010 as an example of a "apt" and its "recent" variances," in *2018 21st Saudi Computer Society National Computer Conference (NCC)*, 2018, pp. 1–5.

[30] "TRITON Malware Targeting Safety Controllers ncsc.gov.uk," https://www.ncsc.gov.uk/information/triton-malware-targeting-safety-controllers, [Accessed 11-10-2023].

[31] "Techniques - ICS | MITRE ATT&CK&xAE; — attack.mitre.org," https://attack.mitre.org/techniques/ics/, [Accessed 02-06-2024].

[32] "Initial Access, Tactic TA0108 - ICS | MITRE ATT&CK&xAE; — attack.mitre.org," https://attack.mitre.org/tactics/TA0108/, [Accessed 02-06-2024].

[33] "Discovery, Tactic TA0102 - ICS | MITRE ATT&CK&xAE; — attack.mitre.org," https://attack.mitre.org/tactics/TA0102/, [Accessed 02-06-2024].

[34] "Command and Control, Tactic TA0101 - ICS | MITRE ATT&CK&xAE; — attack.mitre.org," https://attack.mitre.org/tactics/TA0101/, [Accessed 02-06-2024].

[35] "Impact, Tactic TA0105 - ICS | MITRE ATT&CK&xAE; — attack.mitre.org," https://attack.mitre.org/tactics/TA0105/, [Accessed 02-06-2024].

[36] "Indicator Removal, Technique T1070 - Enterprise | MITRE ATT&CK&xAE; — attack.mitre.org," https://attack.mitre.org/techniques/T1070/, [Accessed 02-06-2024].

[37] "The Gaspot Experiment: How Gas-Tank-Monitoring Systems Could Make Perfect Targets for Attackers - Security News — trendmicro.com," https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/the-gaspot-experiment, [Accessed 11-10-2023].

[38] "GitHub - sjhilt/GasPot: GasPot Released at Blackhat 2015 — github.com," https://github.com/sjhilt/GasPot, [Accessed 02-06-2024].

[39] Conpot, "Conpot — conpot.org," http://conpot.org/, [Accessed 02-06-2024].

[40] "GitHub - sk4ld/gridpot: Open source tools for realistic-behaving electric grid honeynets — github.com," https://github.com/sk4ld/gridpot, [Accessed 02-06-2024].

[41] "SCADA HoneyNet Project: Building Honeypots for Industrial Networks — scadahoneynet.sourceforge.net," https://scadahoneynet.sourceforge.net/, [Accessed 02-06-2024].

[42] D. Antonioli, A. Agrawal, and N. O. Tippenhauer, "Towards high-interaction virtual ics honeypots-in-a-box," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, ser. CPS-SPC '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 13–22. [Online]. Available: https://doi.org/10.1145/2994487.2994493

[43] J. You, S. Lv, Y. Sun, H. Wen, and L. Sun, "Honeyvp: A cost-effective hybrid honeypot architecture for industrial control systems," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[44] D. Nardella, "Snap7 Homepage — snap7.sourceforge.net," https://snap7.sourceforge.net/, [Accessed 02-06-2024].

[45] E. López-Morales, C. Rubio-Medrano, A. Doupé, Y. Shoshitaishvili, R. Wang, T. Bao, and G.-J. Ahn, "Honeyplc: A next-generation honeypot for industrial control systems," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 279–291. [Online]. Available: https://doi.org/10.1145/3372297.3423356

[46] J. B. J. L. Robert Jaromin, Barry Mullins, "Design and Implementation of Industrial Control System Emulators," https://inria.hal.science/hal-01456891/document, [Accessed 01-11-2023].

[47] "Data Privacy - ICS Group — ics-group.eu," https://www.ics-group.eu/en/information-und-service/data-privacy, [Accessed 01-11-2023].

[48] G. Cheng, Y. Lin, J. Yan, J. Zhao, and L. Bai, "Model-measurement data integrity attacks," *IEEE Transactions on Smart Grid*, vol. 14, no. 6, pp. 4741–4757, 2023.

[49] M. Cook, A. Marnerides, C. Johnson, and D. Pezaros, "A survey on industrial control system digital forensics: Challenges, advances and future directions," *IEEE Communications Surveys Tutorials*, vol. 25, no. 3, pp. 1705–1747, 2023.

[50] "Honeyd — honeyd.org," https://www.honeyd.org/, [Accessed 02-06-2024].

[51] Fyodor. (1998, October) Remote os detection via tcp/ip stack fingerprinting. [Online]. Available: http://www.nmap.org/nmap/nmap-fingerprinting-article.html

[52] O. Arkin and F. Yarochkin. (2002, August) Xprobe v2.0: A "fuzzy" approach to remote active operating system fingerprinting. [Online]. Available: http://www.xprobe2.org

[53] "GitHub - SCADA-LTS/Scada-LTS: Scada-LTS is an Open Source, web-based, multi-platform solution for building your own SCADA (Supervisory Control and Data Acquisition) system. — github.com," https://github.com/SCADA-LTS/Scada-LTS, [Accessed 02-06-2024].

[54] A. Dehlaghi-Ghadim, A. Balador, M. H. Moghadam, H. Hansson, and M. Conti, "Icssim — a framework for building industrial control systems security testbeds," *Computers in Industry*, vol. 148, p. 103906, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361523000568

[55] T. Williams, "The purdue enterprise reference architecture," *IFAC Proceedings Volumes*, vol. 26, no. 2, Part 4, pp. 559–564, 1993, 12th Triennal Wold Congress of the International Federation of Automatic control. Volume 4 Applications II, Sydney, Australia, 18-23 July. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017485326

[56] T. R. Alves, M. Buratto, F. M. de Souza, and T. V. Rodrigues, "Openplc: An open source alternative to automation," *IEEE Global Humanitarian Technology Conference (GHTC 2014)*, pp. 585–589, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:5626946

[57] "Zeek &; Book of Zeek (v5.1.0) — docs.zeek.org," https://docs.zeek.org/en/v5.1.0/, [Accessed 16-10-2023].

[58] "ELK Stack: Elasticsearch, Kibana, Beats & Logstash — elastic.co," https://www.elastic.co/elastic-stack, [Accessed 02-06-2024].

[59] "cisagov/ICSNPP: Industrial Control Systems Network Protocol Parsers — github.com," https://github.com/cisagov/ICSNPP.

[60] "A. Swales. 1999. Open Modbus/TCP specification. Schneider Electric 29 (1999), 3–19." A.Swales.1999.OpenModbus/TCPspecification.SchneiderElectric29(1999),3–19., [Accessed 05-12-2023].

[61] "GreyNoise is the source for understanding internet noise — greynoise.io," https://www.greynoise.io/, [Accessed 03-07-2024].

Acknowledgments

This thesis would not have been possible without the support of many people to whom I am deeply grateful.

First and foremost, I would like to thank my supervisor, Prof. Mauro Conti, for his unwavering support, insightful guidance, and continuous motivation throughout this journey.

I am also extremely thankful to my co-supervisor, Dr. Federico Turrin, for his expertise and constructive suggestions that have significantly enhanced the quality of my thesis. His dedication and patience have been a source of inspiration.

I sincerely thank VSIX for facilitating the installation of the honeypot at their IXP and enabling us to collect valuable data, which has been a crucial part of my thesis.

I want to acknowledge the support of my family, particularly my parents and my sister. Their unconditional love, belief in my abilities, and constant encouragement have been my pillars of strength, thanks to my friends, who have been my companions through both the highs and lows of this journey, your support and understanding have been invaluable. Thank you for always being with me.

I am finally very thankful to all the people who directly or indirectly contributed to the completion of this thesis. To all of them, my sincere thanks.