Spying by SPi, I got the Birds-Eye

Awais Yousaf[1][0000−0001−8593−8273], Lee Ling Yi Kalvin[2], Meixuan Li[1][0009−0004−8515−2659], and Jianying Zhou[1][0000−0003−0594−0432]

[1] iTrust, Centre for Research in Cyber Security
Singapore University of Technology and Design (SUTD)
Singapore.
{awais_yousaf,jianying_zhou}@sutd.edu.sg
meixuan_li@alumni.sutd.edu.sg
[2] Information Systems Technology and Design (ISTD)
Singapore University of Technology and Design (SUTD)
Singapore.
kalvin_lee@alumni.sutd.edu.sg

Abstract. This paper explores a covert network mapping approach to target autonomous ships using a discreet and elusive spying machine called SPi. Our research aims to design SPi as a small pocket-sized device, enabling it to remain concealed while scanning the ship's information technology (IT) or operational technology (OT) systems. Once connected to the ship's network, SPi initiates information gathering through our customized script. Additionally, SPi establishes a covert communication channel with an external attacker sailing outside the target ship. This enables the attacker to remotely access the SPi device to collect data, gain SPi control, and utilize SPi as a platform for launching further attacks on the ship. The focus of this research lies on ensuring SPi's adaptability to diverse network configurations on target ship systems. We developed a spying machine that can effectively evade detection, extract comprehensive information about the ship's IT and OT systems, and serves as a staging factor for launching future attacks.

Keywords: Spying Machine · Covert Mapping · Network Scanning · Autonomous Ship · Operational Technology · Maritime Cybersecurity

1 Introduction

The maritime industry is very important for global trade. It is responsible for transporting approximately 80% of the world's goods [25] and maritime trade grew 2.4% in 2023 [26]. With the emergence of autonomous ships in recent years, it has garnered significant attention due to their potential to revolutionize maritime operations, enhance efficiency, and improve safety. Autonomous ships use advanced technologies, including artificial intelligence and machine learning algorithms, enabling them to navigate and operate without direct human intervention. The appeal of increased operational efficiency, reduced human error, and improved safety has generated substantial hype surrounding autonomous

ships. Autonomous ships rely on various technologies such as autonomous navigational systems, intelligent engine monitoring and control, SCADA systems and cluster of Internet of Things (IoT) devices and sensors to realize these benefits. However, as the maritime industry embraces these technological advancements, concerns regarding cybersecurity threats and vulnerabilities have become increasingly prominent. The reliance on interconnected systems and technologies within autonomous ships introduces a new dimension of risk [18].

Adversaries are keenly aware of the potential impact of cyber-attacks on these vessels [30], including their potential to disrupt operations, compromise safety, and facilitate unauthorized access to critical systems. The focus of this paper is to explore the development of a secret spying device, hereafter referred to as SPi, which aims to compromise targeted systems by leveraging Ethernet connections. The SPi is designed to scan the ship's Information Technology (IT) and Operational Technology (OT) infrastructure covertly, establish a persistent foothold and gather extensive information about the onboard systems. Furthermore, it aims to establish a secret connection with an external attacker sailing outside the target ship, providing him a bird's eye view of the target ship's IT and OT system.

This work includes the development, evaluation, and testing of the SPi device for black box scanning of network resources of a Cyber Physical System (CPS) onboard semi-autonomous [22] passenger ferry. The primary objective is to design a sophisticated **S**pying machine based on Raspberry **Pi** (SPi), tailored to infiltrate CPS. The design emphasizes the adaptability of SPi to network topologies and different configurations found in the network infrastructure of CPS. SPi will be equipped with tools and scripts to covertly gather substantial amount of information about the target CPS onboard autonomous ship. Additionally, the SPi will establish a remote connection with an attacker sailing outside of target ship, providing him birds-eye view of the target CPS without raising suspicions. Finally, based on the collected information, SPi will be evaluated to assess its usefulness as a staging platform by launching potential cyber attacks. For this purpose, a conceptual attack scenario will be tested for the demonstration of usefulness of the mapping and scanning results of SPi in subsection-4.1.

Rest of the paper is organized as follows. Section-2 highlights notable work related to maritime cybersecurity. Section-3 explains the working of SPi machine along with required configurations, scripting and automation of attack. Results and analysis of extracted birds-eye view of target CPS is presented in Section-4. A proof of concept implementation is provided in subsection-4.1 whereas attack model is summarized in subsection-4.2. Mitigation measures to counter the effect of SPi are suggested in subsection-4.3. Strengths and limitations associated with the usage of SPi for real world maritime scenarios are discussed in the subsection-4.4. Moreover, risk analysis of using SPi device against different kinds of ships is conducted in Section-5. Finally, conclusions are drawn in Section-6.

2 Related Work

Dynamics of cybersecurity threats and their evolving nature pose many ongoing challenges for autonomous ships. Therefore, ship operators and stakeholders must remain vigilant and continuously update security measures to address emerging threats. Autonomous ships face significant cybersecurity challenges, including the risk of cyber-attacks, vulnerabilities in integrated systems, and manipulation of tracking system.

Network scanning has been researched and significant contributions have been made to advance the detection and protection of industrial control system equipment. A practical approach for scanning and protecting industrial control system equipment is presented in [31]. The research team focuses on detecting networked industrial control system equipment by leveraging the Nmap scanning framework and the Modbus communication protocol. The paper provides a comprehensive and detailed overview of the scanning process tailored explicitly for Schneider Programmable Logic Controllers (PLCs). The authors used Nmap Scripting Engine (NSE) script to scan the target PLCs. To identify a Modbus device, Function Code (FC) 43 of Modbus protocol is used. This meticulous approach enables the extraction of critical information from the scanned Schneider PLCs. During the conducted experiments, the capabilities and the effectiveness of the proposed scanning method was extensively evaluated. The results demonstrated the successful identification and extraction of critical properties associated with the Schneider PLCs. The extracted information encompassed essential details such as vendor name, network module specifications, microprocessor specifications, firmware version, memory card model, project information, project revisions, and last modified dates.

In the context of Industrial Control Systems (ICS), both passive and active scanning techniques play crucial roles in identifying vulnerabilities, assessing the security posture of the network, and ensuring the integrity of onboard autonomous ship systems. Reconnaissance whether active or passive is one of the most important component of any cybersecurity operation [9]. Passive scanning involves the monitoring and analysis of network traffic without actively probing or interacting with the target systems. It aims to identify anomalies, detect suspicious activities, and gain insights into the network infrastructure behavior. On the other hand, active scanning is an approach that actively probes the network infrastructure to discover potential vulnerabilities, weaknesses, or misconfigurations [21]. It involves sending crafted network packets or specific queries to the target systems to elicit responses, analyze the behavior of services, and identify potential entry points for exploitation.

As mentioned earlier, autonomous ships face various cybersecurity challenges due to interconnected nature of their systems [4, 6, 13] and these challenges must be addressed to ensure their safe and secure operation. One of the primary challenges is the potential for cyber-attacks targeting the ship's control systems, communication networks, and infrastructure. Unauthorized access or control over critical ship functions, such as navigation, propulsion, or cargo can have severe consequences, leading to physical damage, safety risks, reputational or financial

losses. For example, in [6] authors explain the challenges associated with Automatic Identification System (AIS) for command and control in maritime sector. AIS is a tracking system widely used in the maritime industry for vessel identification and collision avoidance. Malicious actors can exploit AIS vulnerabilities to manipulate vessel navigational information, leading to potential collisions or cyber hijacking of autonomous ship. Similarly work in [13] discusses cyber-attacks against autonomous ships. These attacks can target various elements, such as communication systems, sensor networks, or control systems potentially disrupting ship operations, compromising safety, or stealing sensitive data. The paper emphasized the importance of implementing robust cybersecurity measures, including secure communication protocols, access controls, and intrusion detection systems to mitigate the cyber attacks.

Similarly, the research presented in [23] highlights a significant security challenge related to autonomous ships. The paper demonstrated the vulnerability of Unmanned Surface Vehicle (USV) to injection and replay attacks on its distributed Guidance, Navigation, and Control (GNC) systems. Through injection and replay attacks, the paper showcases how an attacker can manipulate navigational parameters and take control of the underactuated USV. Later in their work, an authenticated encryption is proposed as mitigation measures for prevention of injection and replay attacks.

An in-depth survey on five well know cyberspace scanners is presented in [19] that discusses the system architecture, scanning frequency, supported internet protocols and information gathering capability of Shodan, Censys, Fofa, BinaryEdge, and ZoomEye. These are automatic scanning tools and their results are made publicly available. Another survey on the network scanning tools is provided in [27] which divides the network scanning tools into two groups based on the mode of their sharing of scanning results. Among the first group of tools that share their results publicly are Shodan, Censys, Thingful, ZoomEye, and PunkSPIDER. In contrast, the second group of scanning tools, which are user-based and require user interactions, consists of Nessus, Vega, Skipfish, Acunetix, Vulners and DRUNK. A maritime use case specific passive network scanner based on Shodan and Censys is proposed in [5] that is heavily dependent on National Marine Electronics Association (NMEA) messages for extraction of physical (Location, Speed, Time, Heading, etc.) and cyber (IP addresses, Ports, Services, etc.) properties. While there are numerous widely-used network scanning tools, their effectiveness can be limited when stealth and compact form factors are required. Hence a light weight but maritime specific network scanning tools is developed in this work which is based on Nmap scan, Modbus scan and Ethernet IP scan for network reconnaissance. Following reconnaissance as an initial step, SPi serves as a staging factor which enables an attacker to launch cyber attacks on shipboard maritime OT systems. Lastly, the proposed reconnaissance tool (SPi) is more suitable for the isolated targets like ships sailing in open waters.

3 Spying Machine based on Raspberry Pi - (SPi)

The development of Spying machine based on Raspberry Pi involves lot of hardware and software configurations and setups on Raspberry Pi board. The required configurations and setups enable SPi machine to automatically carry out the network scanning when maliciously plugged into a ship network. During the scanning, it maintains a stealthy covert communication channel between target ship and an attacker boat sailing near the target ship. The selection of Raspberry Pi board is crucial in implementing and testing the SPi machine. In the beginning, Raspberry Pi 3B with 1GB of RAM was used. However, during the testing phase of SPi, it became apparent that the Raspberry Pi 3B needed more processing power and speed for optimal scanning. Therefore, an upgrade to a Raspberry Pi 4 with 8GB of RAM and a more powerful processor (Quad core Cortex-A72) is decided. Moreover, SPi is equipped with Power over Ethernet (PoE) HAT which provides dual benefits. PoE HAT provides power to SPi as well as it provides connection to target network via Ethernet cable. Ethernet cable is preferred since most ships have their USB ports disabled to prevent connection of unauthorized or malicious USB devices to the ship's systems.

Step by step spying by SPi is depicted in Figure-1. Each step is numbered as **(step_number)** for clarity, and these steps will be further explained in detail in the subsequent sections of this paper.

3.1 Auto Login

First, Kali Linux distribution with LightDM display manager is installed on Raspberry Pi board. To enable the *auto login* feature upon getting power via PoE on target ship newtork, modifications to two configuration files in the system are required. The first file can be access at the location */etc/lightdm/lightdm.conf*. We need to make changes to two lines by uncommenting them (removing the "#" symbol) and modifying them as *autologin-user=kali*. By setting this line to the user account *kali*, we configure the system to automatically log in as the user *kali* upon booting or starting the Graphical User Interface (GUI) session. In other words, when the system starts up, it won't prompt for a *username* and *password*. Second change in this file is the setting of timeout equals to zero i.e., *autologin-user-timeout=0*. By doing this, delay between the completion of the boot process and the automatic login of the *kali* user is eliminated. The second file to amend is */etc/pam.d/lightdm-autologin*. In this file, we need to comment out the line *auth required pam_succeed_if.so user != root quiet_success*. By adding a "#" symbol to the start of this line, *auto login* functionality for the *root* user is disabled. Auto Login is depicted as step (1) in Figure-1.

3.2 SPi as Hidden Access Point

Secondly in step (2), SPi is configured to work as Hidden Access Point (HAP). This enables an attacker sailing outside the target ship to connect and establish a covert connection with SPi. HAP is created by using two separate software

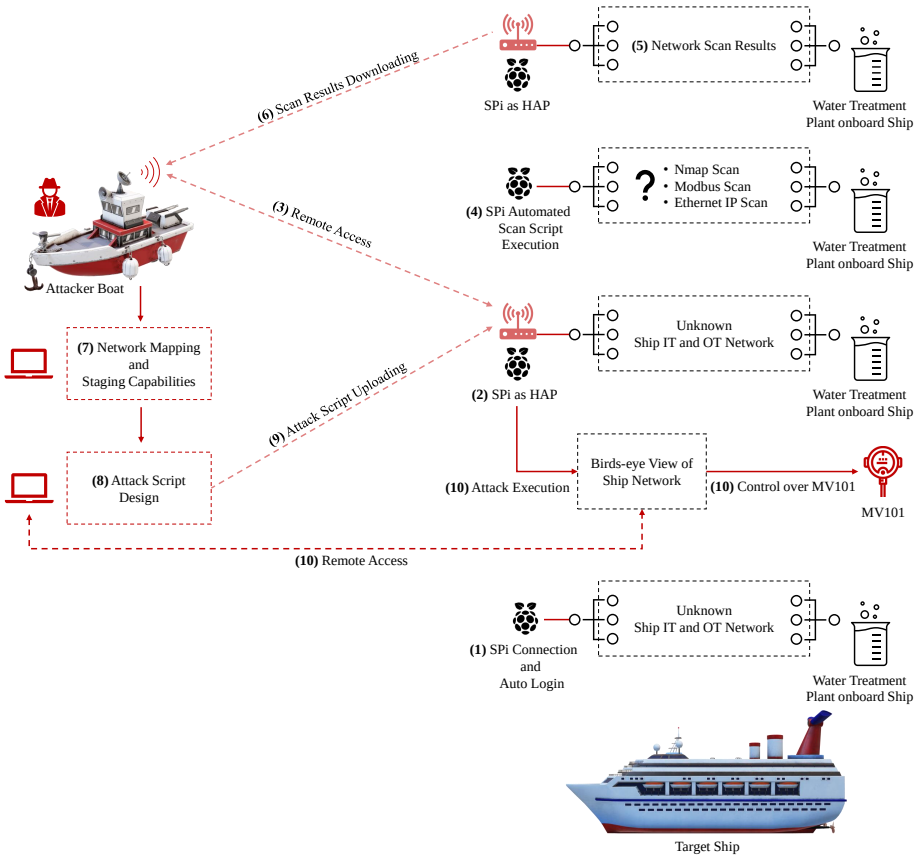

Fig. 1. Step by Step Spying by SPi

tools in Kali Linux namely Host Access Point Daemon (*Hostapd*) and DNS Masquerade (*dnsmasq*). *Hostapd* will manage the wireless access point functionality, enabling devices to connect and communicate wirelessly and remotely, while *dnsmasq* handles Domain Name System (DNS) and Dynamic Host Configuration Protocol (DHCP) services, simplifying the network configuration process for connected devices. *dnsmasq* acts as a DNS cache, storing recent DNS query results locally, which speeds up DNS queries. Additionally, it offers DHCP services, automatically assigning IP addresses to devices on the network, simplifying the process of joining HAP and obtaining necessary network settings easily. Using both tools together, the SPi machine functions as HAP, allowing the remote attacker to connect and access it. This is essential for the later part of the experiment when the attacker would need to establish a connection to the SPi and download the scanned results from the SPi machine.

3.3 Remote Access

Virtual Network Computing (VNC) is a software that facilitates remote access and control of a computer over a network [2]. Users can interact with the remote desktop as if they are physically present at the computer. We opted to use X11VNC in this paper. It allows to establish a VNC server on a machine (SPi), enabling remote access and control of its desktop. After installing VNC Sever on SPi machine using package manager of Kali Linux, configurations like setup of password, display settings and authentication method are done. Once we set the password, X11VNC will save the password to a file named *.vnc/passwd* in the home directory of the user running the X11VNC. To start VNC server on SPi machine, we used the command *x11vnc -ncache 10 -auth guess -nap -forever -loop -repeat -rfbauth /home/kali/.vnc/passwd -rfbport 5900 -noncache* which allows the server running indefinitely. Moreover, to ensure VNC Server will continue to run until the user stops it or reboots SPi and it automatically starts during the boot process, a bash script is written for this purpose. Once setup of VNC Server is completed, we head over to the attacker laptop to install VNC Viewer which is client side of VNC Server. This enables an attacker sailing outside the target ship to have full access of SPi machine when required (step (3)).

3.4 Python Script

The SPi is designed to automatically perform network scanning of target network and creates a birds-eye view of unknown network of IT and OT systems. To achieve this, a python script is written that first scans the network (step (4)) for live hosts and then proceed to scan for PLCs and other devices before saving the results (step (5)). It leverages on existing Python libraries, including Nmap, netifaces, pycomm3, and pymodbus to carry out the network scanning and mapping tasks. In the following, we break down the main functions of the Python script and explain how they work.

- *main()*: The main function is the starting point of the script and initiates the network scanning process. It calls many other functions. For sake of brevity, only the usage of most important functions is provided below. After scanning is finished, this function stores results in different files on SPi machine. Total 25 files (Figure-2) are produced with the execution of this function. Details of the files stored on SPi are discussed in Section-4.
- *get_network_range(interface)*: This function calculates the network range based on the IP address and subnet mask of the specified network interface by using the *netifaces* library to retrieve the IP address and subnet mask. Subsequently it modifies the IP address to obtain the network base IP address by setting the last part to zero, representing the starting address of the network range (e.g., 192.168.1.0). This ensures that the network scanning process targets only devices within the same network segment, avoiding unnecessary scanning of devices outside the network's scope. Lastly, it returns the calculated network range, enabling targeted and efficient network scanning.

- *scan_network(interface)*: This function performs network scanning using the Nmap library. It discovers live hosts on the network through Internet Control Message Protocol (ICMP) host discovery. For each live host, it performs a full TCP port scan to identify open ports. The function then extracts additional information and returns a list *scan_results* which contains information about the discovered devices, including their IP addresses, MAC addresses (if available), and lists of open ports. The scan results *scan_results* serve as a foundation for other scanning function in the later part of the script.
- *scan_ethernetip_plcs(scan_results)*: This function filters the *scan_results* list to include only devices with open port 44818 (common network port used for industrial communications). It then performs scans on these devices using the 'LogixDriver' class. For each Programmable Logic Controller (PLC) device, the function retrieves controller information and program tags from the PLC. Then the scan results of this step are saved in separate files for each PLC device. This function provides valuable insights about the configuration and programming of Ethernet/IP PLCs on the network. It is a vital part of the network scanning process, allowing the script to focus on specific devices and collect detailed information about the identified PLCs.
- *scan_modbus_slaves(scan_results)*: This function allows the script to focus on Modbus enabled devices and collect relevant data from them. It filters the *scan_results* list to include only devices with open port 502 (common port for Modbus-TCP protocol [24]), indicating that their might be Modbus devices attached. It then performs scans on these Modbus devices and attempt to read their holding registers [7]. For each Modbus slave, the function extracts the contents from holding registers and saves the results in separate files for each device. This function provides valuable insights into the Modbus enabled devices on the network and the data accessible through Modbus communication.

3.5 Script Automation

The primary objective of the SPi is to covertly perform scanning and mapping of target network along with collection of important information. To achieve this, automation of the script (step (4)) responsible for scanning the target network is done using Kali Linux services. There are several ways to enable script automation in Kali and one of the most widely used method is through *Systemd*. *Systemd* is a system and service manager for Linux operating systems. It is designed to start and manage services, handle system processes, and provide various functionality related to process control, logging, and resource management. By utilizing *Systemd*, a service unit *scan.service* is created that executes the Python script automatically at system start-up or at scheduled intervals. This ensures that the spying device can function seamlessly without the need for continuous manual initiation. *Systemd* service unit is saved as */etc/system/system/scan.service* with execution permission granted and the contents of *scan.service* is shown in the Listing-1.1. Lastly, we need to reload the *Systemd Manager* configuration to update its services list and to enable the automatic

execution of *scan.service* at start-up. Status of *scan.service* can be verified by using *sudo systemctl status scan.service* command on SPi machine.

Listing 1.1. Contents of Systemd Service

```
[Unit]
Description=Network Scan Script
After=network.target

[Service]
ExecStartPre=/bin/sleep 5
ExecStart=/usr/bin/python3 /home/kali/scan/scan.py
User=kali

[Install]
WantedBy=multi-user.target
```

4 Results and Analysis

An innovative spying machine SPi is designed that is capable of scanning and mapping the network infrastructure of a ship. As a test case, water treatment plant onboard passenger ship is decided to test the efficacy of SPi machine. However, the challenge lies in the limited accessibility to real ships for testing purposes. To overcome this hurdle, we have strategically chosen to conduct experiments using a water treatment testbed that faithfully simulates the operational conditions of a water treatment plant. Details of the water treatment testbed used in this work are available in [20]. Despite the contextual differences, the similarities in system architecture allow us to utilize this testbed effectively, providing a valuable and representative environment for validating the efficacy of SPi machine. It is assumed that SPi is plugged-in (step (1)) to a ship network by a cyber mercenary or disgruntled employee [30] and it has enough signal strength to communicate with an attacker sailing outside the target ship. Once SPi is plugged-in, it gets power via PoE, automatically boots is Kali OS, logged-in and establishes a connection with the ship network. After establishing a connection, it uses *scan.service* that automatically runs python script to initiate the scanning process. Scanning process scans for network mapping as well as protocol discovery in target network. Finally, after rigorous scanning, it keeps the results in different files as shown in Figure-2. Breakdown of collected scan results (total 25 files) are summarized in Table-1. Scanning and collection of results are depicted as step (4) and step (5) respectively in Figure-1.

Once an attacker sailing outside the target ship reaches within the range of wireless signal, it establishes a connection to HAP. The Secure Shell (SSH) Protocol is used for securely sending commands to SPi. After gaining remote access to SPi via VNC Viewer, the outsider attacker can transfer scan result files from the SPi machine to his laptop using Secure Copy Protocol (SCP) (step (6)). In addition, outsider attacker can also upload files to the SPi machine, allowing him to serve as a staging platform (step (7)) for potential future attacks.

Table 1. Breakdown of Collected Results

Scans	Results Collected
Nmap Scan	1
Modbus Scan	2
Ethernet IP Scan	22

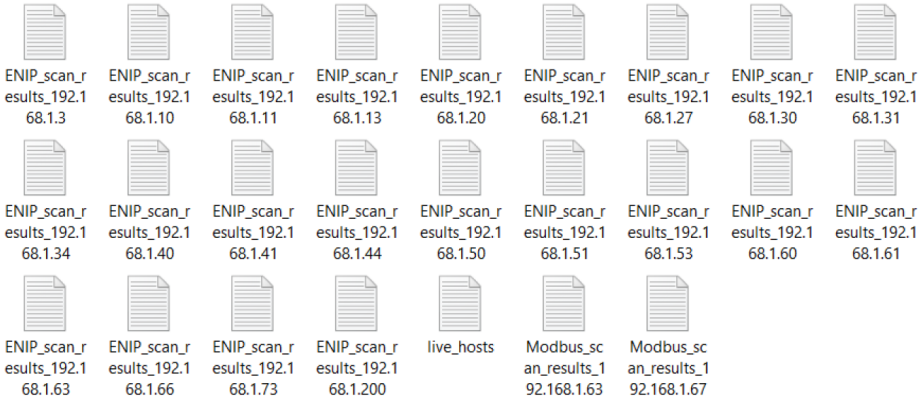

Fig. 2. Scan Results Downloaded from SPi

The Nmap scan conducted on the target network provides valuable insights, identifying a total of 39 live hosts. Among these live hosts, 22 of them had port 44818 open, prompting the python script to initiate a secondary Ethernet IP scan. The purpose of this secondary scan is to identify any processes connected to these hosts, providing essential information about the network operational status and potential vulnerabilities. Additionally, 2 of the live hosts were found to have port 502 open. This allowed the python script to conduct another secondary scan, this time using the Modbus protocol. The aim of this Modbus scan is to determine whether any processes were connected to these specific hosts, gaining insight into the communication and functionality of the connected devices. The identification of open ports and the subsequent secondary scans represent critical steps in comprehending the network structure, topology, and the devices connected to it.

The information collected from the scanning of Modbus protocol revealed limited details. Only the IP addresses of the PLCs (masters) are revealed. Details about the physical devices (slaves) like valves, pumps and sensors controlled by respective PLCs are not discovered. There could be several possibilities for this outcome, such as lack of response due to network configurations or the PLC is simply just a spare or inactive device with no active processes or physical devices linked to it. It is common in industrial environment to practice redundancy; spare PLCs are often kept as backups or as part of training requirements. In such a situation, the spare PLC may be physically present on the network but not

actively controlling any processes or devices at the time of scanning. As a result, the scanning process may only detect the IP address of the spare PLCs without uncovering any associated processes (physical devices) connected to it.

In contrast to the result from Modbus scan, the information collected from tertiary scanning of Ethernet IP protocol revealed much more information with a total of 22 results are collected. Out of 22 results collected, it is observed that 7 results are simply empty, suggesting that the scanned IPs may not be associated with PLCs and a possibility is that these IPs may belong to other network devices or systems. Remaining results show twelve files with six identical results which means SPi is scanning 6 systems along with 6 redundant counterparts. As mentioned above, it is common in CPS to have redundancy, therefore, it is suspected that the scan results that are similar but associated with different IP addresses are representing redundant PLCs. To validate our inference about redundancy of PLCs, an open source tool called WinMerge [3] is used to compare and identify the similarities in identical files (results). WinMerge identified only two differences, the bulk of the contents are similar except for the serial numbers, IP addresses and status. Later, this is also verified manually from the documentation of six stage water treatment testbed that each stage has two PLCs (a primary and a redundant hot-standby) [20].

After extracting scan results from SPi, deeper analysis of results is conducted on attacker's laptop. For sake of brevity, only the results obtained from the file named *ENIP_scan_results_192.168.1.10.txt* (see Figure-2) are explained here. Deeper analysis revealed basic information about the PLCs, such as vendor and production information. A search on the internet with such information can easily reveal vendor specific development platform for PLC programming. This is an important information because the attacker can use it to craft his future attacks. The in-depth analysis also reveals the existence of over 200 *program_tags* intricately linked to the PLC. These *program_tags* serve as key identifiers for various aspects of the system under target, that include an array of commands and services. As we go through the extensive data extracted from SPi machine, the search narrowed down to tags that starts with Human Machine Interface (HMI). The reason for homing in on this term is due to the fact that HMI can expose lot of additional information about the system under target. For example, further analysis revealed a specific tag, named *HMI_P101*, which appears to be of particular interest. Further examining it give out another term *PMP_UDT*. This finding raises the possibility that *HMI_P101* may potentially represent a pump within the system. Using the same approach, we managed to uncover more components that are linked to PLC 192.168.1.10/192.168.1.11 (PLC with IP 192.168.1.11 is in redundant hot-standby mode) and drawn as a subset of network diagram as shown in Figure-3.

Similarly using the same approach, the whole network mapping (step (7)) is created after analyzing all the scanning results obtained by SPi machine. The birds-eye view (step (10)) of extrapolated network topology is depicted in Figure-4. From Figure-4, it is clear that the target network has six subnets with each subnet has two PLCs. Finally, the resultant network mapping (Figure-4)

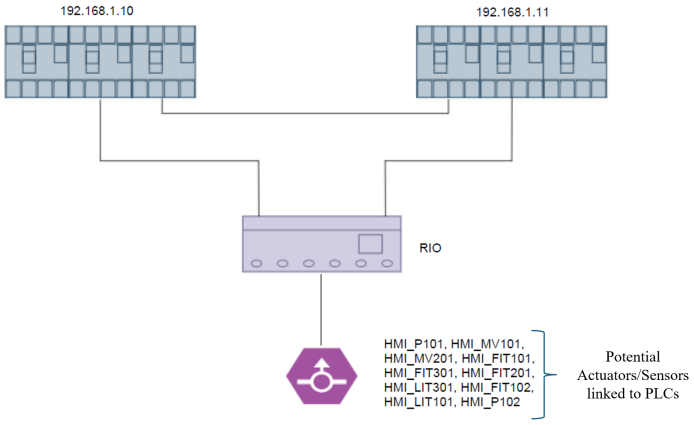

Fig. 3. Subnet based on SPi Scan Results

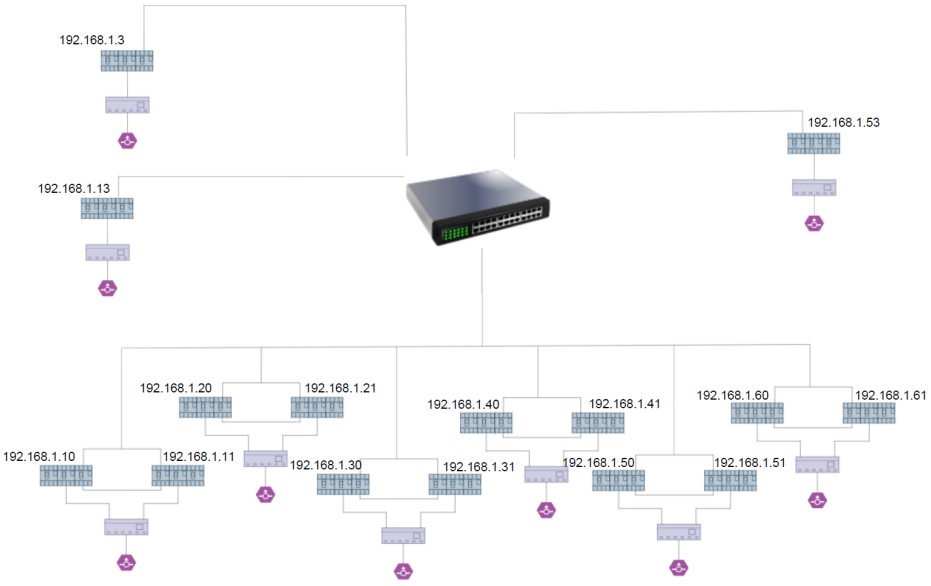

Fig. 4. Birds-eye View of System from Layer-1 Switch

is compared with the actual network architecture of six stage water treatment testbed [20] which proves the accuracy and efficacy of SPi machine.

4.1 Proof of Concept

To reinforce the credibility of the data obtained from our analysis in Section-4, a practical but simple cyber-attack is crafted (step (8)) to gain control over Mo-

torized Valve (MV101). An attack script is written and first transmitted (step (9)) to the SPi from the attacker's laptop via SCP protocol. Subsequently, an SSH connection is established between the SPi and attacker's laptop to take control over the SPi machine. This control over the SPi is necessary to facilitate the execution of the cyber attack script (step (10)). After gaining control of SPi, attack script is executed and we successfully gained control over MV101 (step (10)). The objective of testing cyber attack is to ascertain the real world applicability of SPi in cyber domain and reinforce the accuracy of the mapping results we extrapolated through scanning and analysis. This validation through a cyber attack serves as a litmus test for the efficacy of our findings. By gaining control over MV101, the attack script demonstrated that the information gathered from the scan was accurate and relevant. This proof of concept serves as strong evidence that the SPi can be used as a powerful platform for conducting security assessments and identification of potential vulnerabilities in a target industrial control systems.

4.2 Attack Model Summary

In this subsection, attack model for step by step spying by SPi (Figure-1) is summarized with the help of MITRE Adversarial Tactics, Techniques and Common Knowledge (ATT&CK) ICS matrix [1, 30]. At the time of writing this paper, ATT&CK ICS matrix comprises of twelve tactics and ninety two techniques. Using ICS matrix of ATT&CK, each step of Figure-1 is mapped to (tactic, technique) pairs and all the MITRE ATT&CK tactics and techniques used in the attack model of SPi are tabulated in Table-2.

As mentioned in the start of Section-4 that SPi is connected to the ship network by a cyber mercenary or disgruntled employee as a first step, this step can be mapped to ATT&CK (TA0108, T0817) or (TA0108, T0848) pairs. In the step (2), SPi is configured to act as HAP that can closely be mapped to (TA0108, T0860) pair. In step (3), SPi onboard target ship establishes a remote connection (TA0108, T0886) with an attacker boat sailing outside the target ship. In step (4), SPi executes scan script (TA0104, T0853) for Nmap, Modbus and Ethernet IP scanning (Table-1) which generate results (Figure-2) as step (5). After discovery of network (TA0102, T0842), results are remotely downloaded to the target ship wirelessly via HAP (TA0102, T0887) in step (6). Based on the extracted results from SPi, step (7) involves creating a birds-eye view of the target ship network, (TA0102, T0846) and (TA0102, T0888), which is helpful for establishing a staging factor for scripting and launching (TA0104, T0853) of further attacks in step (8). Step (9) is simply the lateral transfer of attack script to SPi machine (TA0109, T0867). Finally in the last step (10), execution of specially crafted attack script provides command and control of OT systems of the target ship to an attacker (TA0101, T0885) which through brute force I/O (TA0106, T0806) or manipulation of the control of actuators (e.g., MV101), can severely impact (TA0105, T0831) the passengers on board ship by simply wasting precious water.

Table 2. SPi Attack Modeling with MITRE ATT&CK

MITRE ATT&CK				
Matrix	**Tactics (ID)**	**Step**	**Techniques**	**ID**
ICS	Initial Access (TA0108)	1	Drive-by Compromise	T0817
			Rogue Master	T0848
		2	Wireless Compromise	T0860
		3	Remote Services	T0886
	Execution (TA0104)	4	Scripting	T0853
	Discovery (TA0102)	5	Network Sniffing	T0842
		6	Wireless Sniffing	T0887
		7	Remote System Discovery	T0846
			Remote System Information Discovery	T0888
	Execution (TA0104)	8	Scripting	T0853
	Lateral Movement (TA0109)	9	Lateral Tool Transfer	T0867
	Command and Control (TA0101)	10	Commonly Used Port	T0885
	Impair Process Control (TA0106)		Brute Force I/O	T0806
	Impact (TA0105)		Manipulation of Control	T0831

4.3 Countermeasures

The development and testing of the SPi device has highlighted a significant cybersecurity risk facing autonomous ships. The ability to covertly map IT and OT networks onboard ship can potentially pave the way for more disruptive cyberattacks. The experimental results have shown how a low-cost, innocuous device like a Raspberry Pi can be weaponized into an offensive security tool to stealthily map and potentially hijack critical systems on autonomous ships. While the attack demonstrated in the proof of concept shows the malicious control over a single motorized valve, the implications of such attacks could have severe consequences in the real-world scenarios. Attackers can potentially disrupt cargo operations, navigation systems, or even cause environmental disasters. Defending against threats similar to those demonstrated with SPi demand robust and effective defense for any maritime CPS.

Firewalls and Intrusion Detection Systems (IDS) can act as the ship's first line of defenses. A cyber defense architecture presented in [14] for monitoring of Modbus traffic in networked control system could be used for prevention of SPi like attacks. In their work, an alert system is developed that alerts the system if a malicious IP sends too many ICMP ping request to target network. Their system is specifically tailored for Modbus protocol and it can prevent modbus scans. Similarly, a firewall named as SCADAWall demonstrated in [17] could provide effective defense against SPi. Unlike traditional deep packet inspection based firewalls, SCADAWall is equipped with Comprehensive Packet Inspection (CPI) techniques that provide protection against wide range of protocols by analyzing 'Data Fields' of respective protocols.

Software Defined Networking (SDN) can also serve as an active defense against port scanning attacks because it separates control plane from data plane in an ICS. An implementation of Intrusion Detection and Prevention System (IDPS) using SDN is presented in [8] which can be used against port scanning attacks. One possibility to defend against SPi attacks is to strategically modify their proposed Port Bingo (PB) algorithm. As the PB algorithm is designed to prioritize the most valuable packet destination ports for anomaly detection, by incorporating all ports used in the target CPS into the *top_tcp_port_probes* list of the PB algorithm [8], it becomes possible to comprehensively monitor and analyze network traffic. In the event that all requests to access priority ports originate from the same source then this could indicate a potential cyber attack. In such cases, the PB algorithm can be leveraged as preventive countermeasures to identify the malicious requests and discard them.

Similarly, another possible defensive mechanism can be deployed onboard ship by induction of deceptive virtual hosts [28]. Deceptive virtual hosts can attract SPi like machines to themselves, misguiding and forcing them to create a false birds-eye view of the target system. Additional mitigation techniques, such as those outlined in [16] and [10], may also be considered to address network scanning attacks against maritime vessels. The cyber protection of autonomous ships is an ongoing endeavor requiring continuous improvements and adaptation to the evolving threat landscape. Collaborative efforts between maritime industries and cybersecurity researchers are vital to stay ahead of adversaries and ensure secure and smooth sailing of autonomous ships.

4.4 Strengths and Limitations

SPi is a compact device engineered for seamless and automated network scanning with precision and efficiency. Its small form factor makes it ideal to plant it discreetly in almost any CPS environment. Its small form factor also means it draws little attention, blending seamlessly into hardware setups. Once in place, it conducts autonomous network scanning, identifying open ports, vulnerable devices, and exploitable configurations without raising alarms. Moreover, it provides remote access to an attacker sailing outside the target ship for command and control. Therefore with SPi, infiltration becomes faster, stealthier, and significantly more efficient.

There are two major limitations associated with the success of SPi based cyber attacks. First is the requirement of connection of SPi with onboard network of ship systems that requires a member of cyber mercenary or disgruntled employee. Without insider help, spying by SPi device for target systems is just an illusion. Second limitation is linked with the separation between an attacker boat and target ship. Without enough signal strength, an attacker boat is required to be in close proximity to the target ship to establish covert wireless communication channel. Range extenders [11] can be used with Raspberry Pi boards, but they introduce additional challenges such as increased power requirements and hiding of SPi device due to the increase in form factor. Additionally, range

extenders would make SPi device bulky that complicates the installation of SPi device even more harder for an insider malicious actor.

5 Risk Analysis

In this section, the risk analysis of using SPi device against traditional surface vessels, which constitute a significant portion of maritime traffic compared to autonomous vessels, has been conducted. There are many risk analysis frameworks exist today but they are all based on labour, time and knowledge intensive steps [29]. For sake of brevity, a traditional but simple and effective risk ranking method known as Damage, Reproducibility, Exploitability, Affected User, and Discoverability (DREAD) [15] is used to determine the severity of the risk of using SPi against surface vessels. Results of the DREAD risk assessment are summarized in Table-3. Here, all the constituents of DREAD are measured on the scale of 0 to 10 and all the values are selected after a brainstorming session. Moreover, *Risk* is defined as a sum of all the individual constituents of DREAD framework.

Table 3. DREAD Risk Analysis of using SPi against Surface Vessels

Attack	Types of Ship	D	R	E	A	D	Risk
SPi	Tanker	5	4	8	9	8	**34**
	Gas Carrier	5	4	8	9	8	**34**
	Container Ship	8	4	8	10	8	**38**
	Bulk Carrier	8	4	8	10	8	**38**
	Passenger Ship	10	9	6	10	6	**41**
Risk Ratings:							
	Low: 0 - 15		**Medium: 16 - 35**				**High: 36 - 50**

Damage (D) represents capability of a cyber attack to cause injury, harm or destruction. $D = 0$ means 'no damage' whereas $D = 10$ stands for 'maximum harm'. In Table-3, we selected $D = 5$ for tankers and gas carriers because liquid and gas cargo are well protected physically and even thought the success of SPi, it would not be able to cause significant damage to such vessels. Similarly, $D = 8$ is chosen for bulk carriers and container ships because after establishing a covert channel with onboard SPi, an attacker can cause significant damage (e.g. falling of containers) to the cargo by spoofing the rolling angle of the ship due to which ship stability systems respond erroneously. Lastly, $D = 10$ is chosen for a passenger ship because malfunction with the stability of a ship could result severe harm or even death to the passengers.

Reproducibility (R) represents the ease of replication of a cyber attack. $R = 0$ means 'impossible' to replicate and $R = 10$ depict 'easy' to replicate. For a

passenger ship, $R = 9$ is selected because it is relatively easier to hire and board a cyber mercenary onto a passenger ship to install a SPi device onto onboard ship network. For all other types of ships, it is usually very hard ($R = 4$) to have services of malicious insider. Exploitability (E) refers to the required efforts for an attacker to successfully exploit the vulnerability present onboard ship systems. $E = 0$ means 'high level of efforts and resources' are required to exploit a vulnerability whereas $E = 10$ represents 'effortlessness' for an attacker. Except for the passenger ship, $E = 8$ is selected because usually onboard ship systems are quite old and not regularly updated due to the complexities involved during the long lifespan of a ship. Therefore due to the old and outdated systems, it is easy for an attacker to exploit the vulnerabilities. For the passenger ship, $E = 6$ is used because passenger ships are normally equipped with modern systems and regularly updated for passengers safety and safe journey.

Affected User (A) represents the impact of cyber attack on the users. $A = 0$ means cyber attack has 'no impact' on any user and $A = 10$ shows 'everyone is affected'. Value of A is higher for almost any kind of ships. For a tanker or a gas carrier, the SPi device could delay the vessel's arrival at its destination by launching Denial of Service (DoS) on the propulsion system or spoofing attacks on the navigation systems [12]. Due to the delay in the shipment, a businesses could suffer from the financial and reputational losses. Similarly, for a bulk carrier or a container ship, damage to the cargo or falling of a container into the sea could result in financial losses, supply chain disruption, insurance costs and reputational damage along with possible environmental impact. For a passenger ship, after gaining control of onboard ship systems via covert channel between attacker boat and target ship, an attacker could cause critical systems of the ship to malfunction (e.g., power outage, sever pitching and rolling or even capsizing [6, 30]), resulting in a severe impact on the onboard passengers and crew.

Discoverability (D) refers to the ease of finding a vulnerability in a system. $D = 0$ means 'difficult to discover' whereas $D = 10$ represents 'easy to discover' a vulnerability onboard ship systems. Like the rationale behind the selection of exploitability values, discoverability for a passenger ship is also relatively low because it is hard to find vulnerabilities in an updated and secure by design ship systems. After the calculations of *Risk* metric as a sum of individual constituents of DREAD, risk ratings are defined as '**Low**', '**Medium**' and '**High**' and colored with green, orange and red respectively. Risk scores with color ratings are tabulated for five different kinds of ships in Table-3.

6 Conclusion

In this research, we successfully developed and deployed a compact spying machine based on the Raspberry Pi platform - SPi. The SPi was designed to automate the scanning and mapping process and provide insights into potential vulnerabilities within a targeted CPS. Additionally, it also serves as a staging platform for conducting further attacks. Upon connecting SPi to target system, it conducts series of scans on target system to identify live hosts, open ports,

and specific communication protocols. The results of these scans are meticulously analyzed to provide a comprehensive overview of the network structure, topology, and potential vulnerabilities for further exploitation. Specifically, the detailed analysis of Ethernet IP scan results revealed essential information about target system that includes mechanical actuators and sensors that are connected to each PLC, thus providing a bird's eye view of the system. Furthermore, we were able to validate our findings through a practical cyber attack scenario, in which we successfully took control over a motorized valve. The tested scenario further confirms the accuracy and efficacy of SPi device.

As we look ahead, we recognize the potential to expand our research by incorporating the capability of scanning additional protocols, such as NMEA communication protocol used for a navigational bridge of a ship. This expansion will enable us to address security challenges specific to autonomous navigational bridge of a vessels and will further enhance the versatility of SPi and its applicability to diverse critical infrastructure settings.

Acknowledgment

The research is supported by the National Research Foundation, Singapore, under its National Satellite of Excellence Programme "Design Science and Technology for Secure Critical Infrastructure: Phase II" (Award No: NRF-NCR25-NSOE05-0001). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

1. Mitre att&ck®. https://attack.mitre.org/ (Accessed on October 2, 2024), https://attack.mitre.org/
2. Realvnc® - remote access software for desktop and mobile | realvnc. https://www.realvnc.com/en/ (Accessed on October 2, 2024), https://www.realvnc.com/en/
3. Winmerge - you will see the difference.... https://winmerge.org/?lang=en (Accessed on October 2, 2024), https://winmerge.org/?lang=en
4. Akpan, F., Bendiab, G., Shiaeles, S., Karamperidis, S., Michaloliakos, M.: Cybersecurity challenges in the maritime sector. Network 2022, Vol. 2, Pages 123-138 **2**, 123–138 (3 2022). https://doi.org/10.3390/NETWORK2010009, https://www.mdpi.com/2673-8732/2/1/9/htmhttps://www.mdpi.com/2673-8732/2/1/9
5. Amro, A.: Cyber-physical tracking of iot devices: A maritime use case. In: Norsk IKT-konferanse for forskning og utdanning. No. 3 (2021)
6. Amro, A., Gkioulos, V.: From click to sink: Utilizing ais for command and control in maritime cyber attacks. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **13556 LNCS**, 535–553 (2022). https://doi.org/10.1007/978-3-031-17143-7_26/FIGURES/7, https://link.springer.com/chapter/10.1007/978-3-031-17143-7_26

7. Bai, Q., Jin, B., Wang, D., Wang, Y., Liu, X.: Compact modbus tcp/ip protocol for data acquisition systems based on limited hardware resources. Journal of Instrumentation **13**, T04004 (4 2018). https://doi.org/10.1088/1748-0221/13/04/T04004, https://iopscience.iop.org/article/10.1088/1748-0221/13/04/T04004https://iopscience.iop.org/article/10.1088/1748-0221/13/04/T04004/meta
8. Birkinshaw, C., Rouka, E., Vassilakis, V.G.: Implementing an intrusion detection and prevention system using software-defined networking: Defending against port-scanning and denial-of-service attacks. Journal of Network and Computer Applications **136**, 71–85 (6 2019). https://doi.org/10.1016/J.JNCA.2019.03.005
9. Coffey, K., Maglaras, L.A., Smith, R., Janicke, H., Ferrag, M.A., Derhab, A., Mukherjee, M., Rallis, S., Yousaf, A.: Vulnerability Assessment of Cyber Security for SCADA Systems, pp. 59–80. Springer International Publishing (2018). https://doi.org/10.1007/978-3-319-92624-7_3, https://doi.org/10.1007/978-3-319-92624-7_3
10. Duarte, E.K.: An end-to-end defense mechanism for industrial real-time networks (2020)
11. Harum, N., Yusof, N.A.M., Zakaria, N.A.: The development of personal portable wireless range extender for ieee 802.11. In: CSSR 3rd International Conference On Science & Social Research (2016)
12. Junior, W.C.L., de Moraes, C.C., de Albuquerque, C.E., Machado, R.C.S., de Sá, A.O.: A triggering mechanism for cyber-attacks in naval sensors and systems. Sensors 2021, Vol. 21, Page 3195 **21**, 3195 (5 2021). https://doi.org/10.3390/S21093195, https://www.mdpi.com/1424-8220/21/9/3195/htmhttps://www.mdpi.com/1424-8220/21/9/3195
13. Kavallieratos, G., Katsikas, S., Gkioulos, V.: Cyber-attacks against the autonomous ship. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **11387 LNCS**, 20–36 (2019). https://doi.org/10.1007/978-3-030-12786-2_2/FIGURES/3, https://link.springer.com/chapter/10.1007/978-3-030-12786-2_2
14. Kim, C., Robinson, D.: Modbus monitoring for networked control systems of cyber-defensive architecture. 11th Annual IEEE International Systems Conference, SysCon 2017 - Proceedings (5 2017). https://doi.org/10.1109/SYSCON.2017.7934750
15. Kim, K.H., Kim, K., Kim, H.K.: Stride-based threat modeling and dread evaluation for the distributed control system in the oil refinery. ETRI Journal **44**, 991–1003 (2022). https://doi.org/https://doi.org/10.4218/etrij.2021-0181, https://onlinelibrary.wiley.com/doi/abs/10.4218/etrij.2021-0181
16. Kompella, R.R., Singh, S., Varghese, G.: On scalable attack detection in the network. IEEE/ACM Transactions on Networking **15**, 14–25 (2 2007). https://doi.org/10.1109/TNET.2006.890115
17. Li, D., Guo, H., Zhou, J., Zhou, L., Wong, J.W.: Scadawall: A cpi-enabled firewall model for scada security. Computers & Security **80**, 134–154 (1 2019). https://doi.org/10.1016/J.COSE.2018.10.002
18. Li, M., Yousaf, A., Goh, M., Zhou, J., Chattopadhyay, S.: Guidelines for cyber risk management in autonomous shipping. In: Andreoni, M. (ed.) Applied Cryptography and Network Security Workshops. pp. 143–161. Springer Nature Switzerland (2024)
19. Li, R., Shen, M., Yu, H., Li, C., Duan, P., Zhu, L.: A survey on cyberspace search engines. In: Lu, W., Wen, Q., Zhang, Y., Lang, B., Wen, W., Yan, H., Li, C., Ding, L., Li, R., Zhou, Y. (eds.) Cyber Security. pp. 206–214. Springer Singapore (2020)

20. Mathur, A.P., Tippenhauer, N.O.: Swat: A water treatment testbed for research and training on ics security. 2016 International Workshop on Cyber-physical Systems for Smart Water Networks, CySWater 2016 pp. 31–36 (5 2016). https://doi.org/10.1109/CYSWATER.2016.7469060
21. Pospisil, O., Blazek, P., Fujdiak, R., Misurec, J.: Active scanning in the industrial control systems. Proceedings - 2021 International Symposium on Computer Science and Intelligent Controls, ISCSIC 2021 pp. 227–232 (2021). https://doi.org/10.1109/ISCSIC54682.2021.00049
22. Rødseth, Ø.J., Wennersberg, L.A.L., Nordahl, H.: Levels of autonomy for ships. Journal of Physics: Conference Series **2311**, 012018 (7 2022). https://doi.org/10.1088/1742-6596/2311/1/012018, https://iopscience.iop.org/article/10.1088/1742-6596/2311/1/012018https://iopscience.iop.org/article/10.1088/1742-6596/2311/1/012018/meta
23. Solnør, P., Øystein Volden, Gryte, K., Petrovic, S., Fossen, T.I.: Hijacking of unmanned surface vehicles: A demonstration of attacks and countermeasures in the field. Journal of Field Robotics **39**, 631–649 (8 2022). https://doi.org/10.1002/ROB.22068, https://onlinelibrary.wiley.com/doi/full/10.1002/rob.22068https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.22068https://onlinelibrary.wiley.com/doi/10.1002/rob.22068
24. Swales, A.: Open modbus/tcp specification. Schneider Electric **29**, 19 (1999)
25. on Trade, U.N.C., Development: Review of maritime transport 2022 (11 2022)
26. on Trade, U.N.C., Development: Review of maritime transport 2023 (9 2023)
27. Tundis, A., Mazurczyk, W., Mühlhäuser, M.: A review of network vulnerabilities scanning tools: Types, capabilities and functioning. ACM International Conference Proceeding Series (8 2018). https://doi.org/10.1145/3230833.3233287, https://dl.acm.org/doi/10.1145/3230833.3233287
28. Vollmer, T., Manic, M.: Cyber-physical system security with deceptive virtual hosts for industrial control networks. IEEE Transactions on Industrial Informatics **10**, 1337–1347 (2014). https://doi.org/10.1109/TII.2014.2304633
29. Yousaf, A., Amro, A., Kwa, P.T.H., Li, M., Zhou, J.: Cyber risk assessment of cyber-enabled autonomous cargo vessel. International Journal of Critical Infrastructure Protection **46**, 100695 (2024). https://doi.org/https://doi.org/10.1016/j.ijcip.2024.100695, https://www.sciencedirect.com/science/article/pii/S1874548224000362
30. Yousaf, A., Zhou, J.: From sinking to saving: Mitre att&ck and d3fend frameworks for maritime cybersecurity. International Journal of Information Security (2024). https://doi.org/10.1007/s10207-024-00812-4, https://doi.org/10.1007/s10207-024-00812-4
31. Zhou, G., Bai, J., Wang, B., Song, J.: A method of scanning industrial control system equipment. pp. 153–159. Atlantis Press (5 2017). https://doi.org/10.2991/icmeit-17.2017.28, https://doi.org/10.2991/icmeit-17.2017.28