

# Salty Seagull: A VSAT Honeynet to Follow the Bread Crumb of Attacks in Ship Networks <sup>\*</sup>

Georgios Michail Makrakis<sup>1</sup>[0000–0002–1280–6568], Jeroen Pijpker<sup>1, 3</sup>[0009–0008–8334–0655], Remco Hassing<sup>1</sup>[0009–0005–4372–0301], Rob Loves<sup>1</sup>[0009–0003–5406–9185], and Stephen McCombie<sup>2</sup>[0000–0002–6511–9382]

<sup>1</sup> Maritime IT Security Research Group, NHL Stenden, Emmen, The Netherlands  
{george.makrakis, jeroen.pijpker, remco.hassing, rob.loves}@nhlstenden.com

<sup>2</sup> Maritime IT Security Research Group, NHL Stenden, Leeuwarden, The Netherlands  
stephen.mccombie@nhlstenden.com

<sup>3</sup> Department of Computer Science, University of Groningen, Groningen, The Netherlands  
j.pijpker@rug.nl

**Abstract.** Cyber threats against the maritime industry have increased notably in recent years, highlighting the need for innovative cybersecurity approaches. Ships, as critical assets, possess highly specialized and interconnected network infrastructures, where their legacy systems and operational constraints further exacerbate their vulnerability to cyberattacks. To better understand this evolving threat landscape, we propose the use of cyber-deception techniques and in particular honeynets, as a means to gather valuable insights into ongoing attack campaigns targeting the maritime sector.

In this paper we present Salty Seagull, a honeynet conceived to simulate a VSAT system for ships. This environment mimics the operations of a functional VSAT system onboard and, at the same time, enables a user to interact with it through a Web dashboard and a CLI environment. Furthermore, based on existing vulnerabilities, we purposefully integrate them into our system to increase attacker engagement. We exposed our honeynet for 30 days to the Internet to assess its capability and measured the received interaction. Results show that while numerous generic attacks have been attempted, only one curious attacker with knowledge of the nature of the system and its vulnerabilities managed to access it, without however exploring its full potential.

**Keywords:** Cybersecurity, Maritime, VSAT, Honeynet

## 1 Introduction

As one of the most integral components of modern maritime, vessels are equipped with a broad array of cyber systems that encompass both Information Technol-

ogy (IT) and Operational Technology (OT), distributed across various operational domains within the ship. These include communication systems, propulsion and machinery control, power management, navigation, and cargo handling systems. Designed to operate with minimal human intervention, these systems aim to optimize vessel performance while ensuring safety and reliability. However, their increasing complexity and interconnectivity introduce a broad attack surface. Many of these systems depend on interconnected devices and digital services which, if compromised, could significantly disrupt vessel functionality and operational continuity [13, 2].

Previous studies have demonstrated that Very Small Aperture Terminals (VSATs), satellite-based communication systems commonly deployed to provide Internet connectivity to ships at sea, can be exploited through various attack vectors [18, 32]. Given their exposure and relatively accessible attack surface, VSAT systems are often regarded as a low-hanging fruit for adversaries seeking initial access. Once compromised, these systems may serve as a pivot point for further reconnaissance, services disruption, or intrusion into more critical onboard systems, thus posing a significant risk to maritime cybersecurity [5].

To observe and analyze the behavior of adversaries attempting to exploit digital systems, honeypots and honeynets can be strategically deployed. These cyber-deception systems simulate vulnerable targets, luring attackers into interacting with what they perceive to be legitimate services or devices. During these interactions, valuable insights can be gathered regarding the attacker’s movements and techniques. While honeypots are commonly used in traditional IT environments to simulate services (e.g. SSH, HTTP, or FTP), their deployment in Cyber-Physical Systems (CPS), such as those found aboard maritime vessels, presents unique challenges. These challenges stem from the domain-specific nature of the systems involved, which often include proprietary communication protocols and interactions with physical processes.

In this paper, we propose the design and implementation of a VSAT honeynet capable of attracting, recording, and analyzing early-stage interactions from potential attackers targeting shipboard communication systems. To ensure the system’s effectiveness and realism, our design is informed by the ICSvertase framework [10], while the implementation is grounded in the characteristics of real-world, Internet-exposed VSAT devices. To enhance the credibility of the simulation, the honeynet integrates a Web dashboard and Command Line Interface (CLI), alongside replayed voyage data, thereby mimicking the operational features of an actual maritime communication system. Stemmed from real-world VSAT vulnerabilities, we purposefully integrate them into our system to increase the chances of attacker interaction.

Based on the specific nature of the system we try to simulate, we aim to answer the following Research Questions (RQs):

- RQ1: Were there any actors exploiting specific vulnerabilities related to VSAT environments?
- RQ2: What were their interactions with Web dashboard and CLI environments?

- RQ3: Were there any persistence mechanisms attempted?

The collected results from 30 days of deployment indicate that despite the multiple attempts to access the system with generic exploits, only one attacker with knowledge of the nature of the system and its vulnerabilities managed to access it, and perform some familiarization actions inside the environment. Thus, we deduce that specific knowledge is required to meaningfully exploit such systems and gain access to the inner workings of a ship’s network. The anonymized raw data to reach to those conclusions are provided in <sup>1</sup>.

The rest of the manuscript is organized as follows: preliminary background information is presented in Section 2, while our the design of our system is detailed in Section 3. The evaluation of Salty Seagull is described in Section 4, followed by the concluding remarks in Section 5.

**Ethical Considerations:** The ultimate goal of this research is to improve the security of vessels and their satellite communication systems, by understanding the potential attackers’ interactions with them. To this end, all scanning and resources retrieval were conducted via the Shodan and Censys services, without any login attempts or other forms of active interaction with each accessed system.

## 2 Background

In the following, we briefly present the some background information useful to understand the remainder of the paper.

### 2.1 Honeypots and Honeynets

Honeypots are defined by Spitzner as a “security resource whose value lies in being probed, attacked, or compromised” [25]. Conceptually, they function as deliberately vulnerable systems that mimic real services or data to deceive and attract malicious actors. Their core objectives are to divert attackers from critical assets, gather detailed intelligence on adversarial behavior, and engage intruders long enough to monitor and analyze their actions. The origins of honeypots trace back to early deception systems described in Cliff Stoll’s *The Cuckoo’s Egg* [28] and Bill Cheswick’s *An Evening With Berferd* [4], where attackers were lured into controlled environments to better understand their behavior. These foundational works inspired the HoneyNet Project [26], which advanced the use of honeypots as tools for studying and defending against cyber threats. When multiple honeypots are deployed and interconnected, they form a honeynet—an architecture that has since evolved into a powerful method for capturing adversarial Tactics, Techniques, and Procedures (TTPs) [16], enhancing both defensive strategies and cyber threat intelligence capabilities.

---

<sup>1</sup> <https://doi.org/10.5281/zenodo.15469996>

## 2.2 IoT, IIoT and CPS Honeypots

A plethora of honeypots has been developed and used in the past to acquire information about their respective environments [12, 11, 9, 27]. Well known examples of tools used to create honeypots include Cowrie [17], Conpot [21], Glastopf [22], Dionaea [30], and T-Pot [1]. Honeypots have gained considerable attention in industrial environments due to their potential to attract adversaries, such as ransomware gangs, who target these systems for financial gain or adversaries that aim to inflict denial, manipulation and loss of control, view, and safety [7].

Franco et al. compiled a survey on honeypots deployed in the IoT, industrial IoT, CPS and Industrial Control Systems (ICS). They categorized the honeypots based on a multitude of parameters such as levels of interactions, roles, and scalability [6]. According to this research a significant challenge in deploying honeypots in such environments arises from their low-interaction nature. Most implementations rely on simplified versions of the targeted systems, often lacking the complex functionality and specific vulnerabilities that would typically exist in fully operational systems. This makes these honeypots relatively easy for attackers to identify, as they do not replicate the nuanced behavior of genuine industrial environments. Consequently, the lack of sophisticated interaction can limit the effectiveness of these honeypots in evading detection and capturing valuable intelligence.

## 2.3 Shipboard Systems and VSAT

In the context of CPS, consider the modern ship, which is equipped with a variety of integrated systems designed to enhance operational efficiency and support the crew. These systems encompass a wide range of functionalities, from autopilot controls to sensors monitoring environmental factors, such as water temperature. An example of shipboard systems can be found in Table 1.

Among the various systems in a ship, the VSAT is particularly relevant to this study. VSAT is part of SATCOM that enables vessels to maintain internet and television connectivity while at sea. Another use case is the transmission of fishing yield data to cloud services from fishing vessels [23]. It operates as a two-way satellite ground station, utilizing a dish antenna connected to a gateway that facilitates a Wide Area Network (WAN). VSATs commonly include Web as well as command line interfaces for configuration and maintenance tasks. Despite their importance in maintaining communications, VSAT systems have previously been found vulnerable to cyber threats [14, 15]. Exploiting these vulnerabilities could provide adversaries with a potential entry point to breach the internal network of the vessel.

## 3 System Design

In this section, we first introduce the threat model we operate on, then the main design considerations for our honeynet, and finally the technical aspects of the exposed services and the configuration of the environment.

Table 1: Examples of Shipboard Systems according to [19].

System	Components
Communication	Satellite Communication System (SATCOM)
	Integrated Communication System (ICS)
	Wireless Local Area Network (WLAN)
Propulsion, Machinery, and Power Control	Engine Governor System
	Fuel Oil System
	Alarm Monitoring & Control System
	Power Management System
	Emergency Generators and Batteries
Navigation	Electronic Chart Display and Information System (ECDIS)
	Radio Detection and Ranging (RADAR)
	Automatic Identification System (AIS)
	Global Positioning System (GPS)
	Dynamic Positioning System (DPS)
	Global Maritime Distress and Safety System (GMDSS)
	Voyage Data Recorder (VDR)
Cargo Management	Integrated Navigation System (INS)
	Cargo Control Room (CCR)
	Ballast Water System (BWS)

### 3.1 Threat Model

We assume that malicious actors targeting VSAT systems aim to explore these systems and potentially establish a foothold within the vessel’s network. These actors are likely to possess knowledge about the use of VSATs on ships and may have access to documentation that details their operation, which could be either publicly or privately available.

The primary objectives of these attackers include either disabling the VSAT system to create confusion for the crew and passengers, and/or exploiting the system to facilitate lateral movement within the ship’s internal network. To achieve these goals, attackers would typically leverage both documented and undocumented vulnerabilities, often after gathering basic information, such as version details or build numbers, through manual or automated scanning techniques. Additionally, we assume that attackers will be able to fingerprint the IP address of the VSAT system to ensure that they are targeting a legitimate system of a ship.

### 3.2 Design Considerations

With regards to the defined threat model, we leveraged the ICSvertase framework [10] to design our honeynet. Although our system is not strictly an ICS honeynet, we utilized the framework to guide our design, aligning it with components from *MITRE ATT&CK® for ICS* and *MITRE Engage™*. The main key considerations is what adversary behaviors should a honeynet be designed to capture, and the effective capture of them, while at the same time recognize the importance of capturing such behavior as well as, the ways to incentivize an adversary to exhibit such behaviors.

Thus according to the Engage approaches, we aim to *Collect* adversary tools, observe tactics, and other raw intelligence about the adversary’s activity, *Detect* adversary activity throughout an environment, *Direct* them towards an intended path and, *Reassure* them that the access to an environment is real by adding authenticity to deceptive components. The mapping of such activities and techniques used are shown in Tables 2 and 3.

Table 2: Engage Approaches Activities According to ICSvartase Requirements Analysis.

Approach	Activity
Collect	System Activity Monitoring (EAC0003)
	Software Manipulation (EAC0014)
Detect	Introduced Vulnerabilities (EAC0023)
	Network Analysis
Direct	Introduced Vulnerabilities (EAC0023)
Reassure	Information Manipulation (EAC0015)

Collection of activity logs can reveal adversary activity (EAC0003), while making changes to a system’s software properties and functions can achieve a desired effect (reveal deceptive artifacts and systems) (EAC0014). At the same time introduction of vulnerabilities will motivate the adversary to target specific resources (EAC0023) and the concealment and reveal of both facts and fictions will support a deception story (EAC0015).

The corresponding techniques include exploiting CLIs (T0807) for executing commands, using commonly used ports (T0885) to blend in with normal traffic, and leveraging default credentials (T0812) to gain unauthorized access. Adversaries may also restart or shut down devices (T0816) to disrupt operations, exploit vulnerabilities in public-facing applications (T0819) to gain access, and modify system firmware (T0857) for persistence. Finally, the identified techniques involve discovering remote systems (T0846) and gathering information about them (T0888) to facilitate further attacks, as well as using valid accounts (T0859) to maintain access and evade detection.

Based on the above, the design we propose relates to the traditional classification of *medium-interactive honeynets*. The design choices are made based on the assumption that we can acquire more precise results about techniques attackers might exploit to gain initial foothold to the systems of a vessel. We implement the Web interface of a *Sea Tel VSAT management portal* up to a point where a potential attacker can acquire information about the system and to make changes that however do not have any real impact on the system e.g., upload new firmware. The same applies to its accompanied CLI where a variety of commands can be issued with up to two arguments/options following them. While in this study we explore this specific system, we argue that VSAT sys-

Table 3: ATT&amp;CK Techniques According to ICSvertase Requirements Analysis.

Technique	Functional Feature	Data Component
Command-Line Interface (T0807)	Size : single ICS component : Protocols ICS component : OS Logging : file system Logging : processes	Application Log Content Command Execution Process Creation
Commonly Used Port (T0885)	Size : single ICS component : OS Logging : processes	Network Traffic Content Network Traffic Flow
Default Credentials (T0812)	Size : single ICS component : Protocols	Logon Session Creation Network Traffic Content
Device Restart/Shutdown (T0816)	Size : single ICS component : Runtime	Application Log Content Network Traffic Content Network Traffic Flow Device Alarm
Exploit Public-Facing Application (T0819)	Size : single ICS component : Protocols ICS component : Runtime	Application Log Content Network Traffic Content
System Firmware (T0857)	Size : single ICS component : Runtime ICS component : Bootloader Logging : file system	Application Log Content Firmware Modification Network Traffic Content Device Alarm
Remote System Discovery (T0846)	Size : single ICS component : OS Logging : processes	File Access Network Traffic Content Network Traffic Flow Process Creation
Remote System Information Discovery (T0888)	Size : single ICS component : OS Logging : processes	File Access Network Traffic Content Network Traffic Flow Process Creation
Valid Accounts (T0859)	Size : single ICS component : Protocols	Logon Session Creation Logon Session Metadata User Account Authentication

tems from other vendors might also include similar vulnerabilities that allow an attacker to gain access to a ship’s networks.

Given that the vulnerabilities associated with the studied system (CVE-2018-5267, CVE-2018-5266, CVE-2018-5071, CVE-2018-5728) include unauthenticated access to sensitive resources, we deliberately chose not to implement those that could be trivially exploited by automated bots or Web crawlers. Instead, we selectively implemented vulnerabilities related to the use of default credentials and exposure of sensitive information, which are more likely to attract targeted interest from actors with a specific focus on maritime systems and equipment. Importantly, all incoming connections to our system are logged. Therefore, when attempts are made to exploit any of the referenced CVEs, we interpret this as an indication that the adversary has conducted some degree of reconnaissance or prior research upon identifying our system as potentially vulnerable. The honeynet is composed of three parts:

- The Web service. The service responsible for the front-end and the back-end services of a Sea Tel VSAT.
- The Telnet service. The service that provides a CLI interface used to manage the VSAT.
- The VDRPlayer service. The service that enables the replay of voyage information to enhance deception.

Our system’s architecture is shown in Figure 1. Each of the components incorporates a robust logging system to record attacker trace for further analysis and investigation, based on the aforementioned *MITRE ATT&CK® for ICS* Techniques (provided inside brackets in the following subsections).

### 3.3 Web Service

The Web service consists of the proxy, the front-end service and the back-end service of the Web management portal of a Sea Tel VSAT. We describe each as follows:

**Proxy** A minimal configuration of the widely used Nginx proxy is deployed to serve as the entry point to the Web service. This setup allows potential adversaries to perform service discovery [T0846] and attempt to exploit a public-facing application [T0819], in alignment with known adversarial tactics. Nginx was selected due to its built-in support for detailed HTTP request and response logging in JSON format, which facilitates effective data analysis. To enhance the realism of the emulated system, we modified the HTTP response headers to mimic those typically returned by a Sea Tel VSAT.

**Front-end** The front-end of the system comprises static assets, such as HTML, CSS, and JavaScript files, that are served by the back-end component. The user is first presented with a login page, which acts as the landing interface. Upon



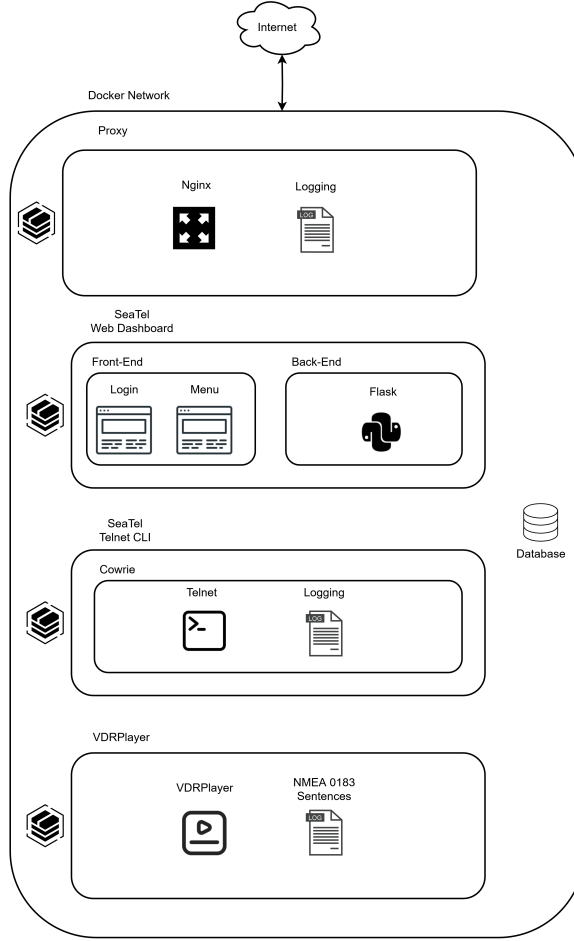


Fig. 1: The architecture of the proposed honeynet.

successful authentication, the appropriate menu page is displayed based on the user's role. The system defines three distinct user roles: "User", "SysAdmin", and "Dealer", each associated with specific access privileges. Consequently, all front-end resources must be served conditionally [T0859].

For instance, a user with the "User" role is primarily permitted to view operational information related to the satellite and antenna. The "SysAdmin" role provides access to system configuration options and diagnostic functionalities, while the "Dealer" role grants administrative privileges, including system commissioning and firmware updates. An illustrative example of the dashboard accessible to a "SysAdmin" user is presented in Figure 2.

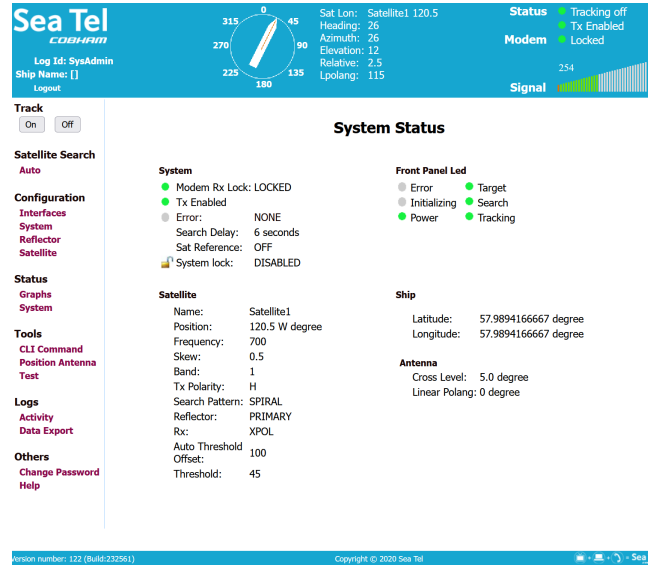


Fig. 2: An example of the VSAT menu page for the SysAdmin.

Those static files were cloned from other internet exposed VSATs as indicated in [3]. Notice, that some of them were needed to be acquired manually so it will not affect the operation of such exposed systems.

**Back-end** We implemented the back-end of the Web service in Python Flask. Since the majority of the front-end requests to the back-end are performed via the included JavaScript files, we deduced the data and format that the responses of such requests should have. To this end, we implemented the required endpoints to return all the necessary information based on a combination of realistic and random data. Information related to the nature of the ship such as the heading and coordinates, are drawn from the data replayed from the VDRPlayer Service (see section 3.5), while other, such as satellite position and antenna azimuth, are randomly created. At the same time, any changes made via the front-end, are processed and stored in a SQLite database to support the deception narrative.

To avoid the access of our Web-page from bots and spiders, Flask’s built-in authentication and authorization mechanisms were used to access each page. The default credentials [T0812] and roles for each user of the system, are stored in the SQLite database. Those credentials can be changed for each user, by accessing the respective page in the Web UI. File uploads for firmware [T0857] and configuration modifications, are saved to explore the potential of an attacker trying to modify the system and maintaining persistence.

Regarding the integrated CLI menu option [T0807], since the access to the vendor’s commands manual for the Web interface was not accessible anymore

in the public domain, we implemented only a subset of the commands that are found when accessing the system via the Telnet service.

### 3.4 Telnet Service

The Sea Tel VSAT system features a CLI for management, which operates on its Antenna Control Unit (ACU) and can be accessed via Telnet [T0807]. Although, not exposed by default, we implement it as such to give the impression of a misconfigured setup. To emulate this functionality, we utilize the widely adopted and academically recognized Cowrie SSH and Telnet honeypot [17]. However, since the VSAT CLI interface includes a distinct set of commands and error messages, it is necessary to customize the default Cowrie setup to better align with the real system [24].

To achieve this, we disabled SSH and implemented a subset of the VSAT CLI commands in Telnet, supporting up to two options for each command. An exception to this rule are commands that were further developed to enhance deception by displaying data from the VDRPlayer service or restarting the system [T0816], as detailed in Section 3.5. The default credentials used in the Web service are also applied to the Telnet service, ensuring consistency. If credentials are modified in one place, the change propagates due to the shared SQLite database. Furthermore, to improve realism, fake but plausible parameters such as MAC addresses and ship names are incorporated. All interactions with the Telnet service are logged in JSON format for subsequent analysis.

### 3.5 VDRPlayer Service

The Voyage Data Recorder (VDR) is an important component of any vessel as it collects and provides forensic evidence in the case of an incident/accident, and monitors system performance. A typical VDR system consists of an electronics unit that gathers data from GPS, heading and speed system, ECDIS, AIS, RADAR, voice feeds from bridge, and rudder response. [8]. Commonly it includes two hard drives that mirror each other for redundancy reasons.

By utilizing the open-source VDRPlayer tool [31], we replay such data to the network and feeding them to the back-end and Telnet services. The data are encoded as NMEA 0183 sentences [20], that the Web and Telnet services decode and process the data accordingly. This ensures that an attacker with access to both services receives the same data, making it believable that the system belongs to a genuine vessel. When the feed of the data ends, we repeat the replay to keep a constant feed of data to our honeynet.

### 3.6 Containerization

Each of the aforementioned services is deployed as a Docker container on a rented Virtual Private Server (VPS) from a commercial cloud provider. The deployment is managed using separate docker-compose.yml configuration files, one for the

Web service and another for the Telnet service, with the VDRPlayer included in the Web service configuration. These files define how the containers are launched, how logs are stored, and how the services interact within the honeynet.

An internal network is established to create the environment, enabling communication between the back-end, Telnet, and VDRPlayer services. For instance, the VDRPlayer replays NMEA data via UDP, which is consumed by both the back-end and Telnet services. The only externally exposed ports are port 80 for the Nginx proxy and port 23 for the Telnet service (T0885), minimizing the attack surface while preserving the honeynet’s intended functionality.

## 4 Evaluation

In this section, we present the experimental setup, analysis of the results from the deployment of our honeynet for 1 month, and some insights from the identified attacks.

### 4.1 Experimental Setup

To evaluate our system, we deployed it on a commercial VPS by a cloud provider located in Europe. The honeynet was hosted on a virtual machine equipped with 4GB of RAM and two virtual CPU cores, running Ubuntu 24.04 as the operating system. To emulate the behavior of a VSAT system, only ports 80 (HTTP) and 23 (Telnet) were left open and accessible from the Internet.

Previous studies in the domain of IT, and IoT/IIoT, CPS and ICS honeynets have explored the deployment of multiple instances distributed across various geographic regions [29]. In contrast, our objective is to realistically simulate the environment of a single vessel. Each ship typically has unique attributes, such as name, call sign, and location, making a single-instance deployment more appropriate for deception. This approach increases the likelihood that adversaries perceive our VSAT system as part of a genuine maritime asset.

To further enhance realism and reduce detectability, we channeled our exposed services through IP address blocks leased from a commercial provider offering both IPv4 and IPv6 addresses. This strategy mitigates the risk of our system being flagged as a honeynet based on a reverse lookup or reputation check of the cloud provider’s IP space. Specifically, we employed Generic Routing Encapsulation (GRE) to tunnel incoming traffic from the leased IP address to the public IP address of the VPS. Subsequently, internal Linux-based routing forwards the traffic to and from the internal Docker network hosting the containerized honeynet.

### 4.2 Data Analysis

This setup has been running from Apr 3, 2025 to May 3, 2025. In this period, we have gathered around 16 MB logs from the Web service and 22 MB from the Cowrie honeypot, making a total of 175,290 entries. Those entries include

attempts to access our honeynet services from 9,054 distinct IP addresses. Table 4 shows the distribution of Web and Telnet connections based on geographical location. In Figure 3, we see the change in log recorded for the two services per day. It is evident that the Telnet service received more attempts than the Web one.

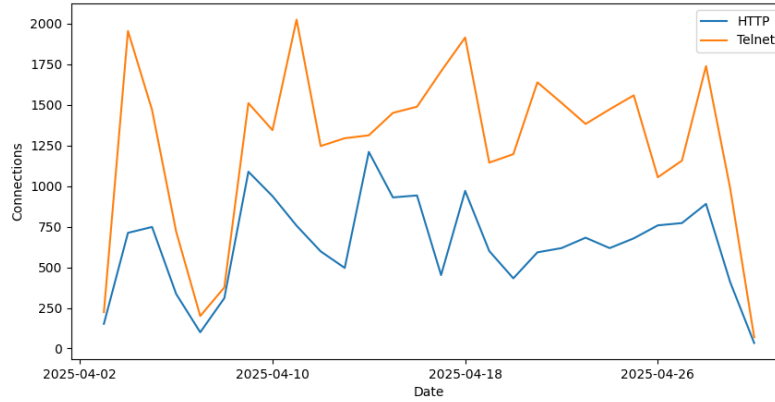


Fig. 3: Change in the number of log entries gathered per day.

Table 4: Top 20 connections received by geolocation.

Geolocation 1/2 Times		Geolocation 2/2 Times	
China	31,895	Indonesia	1,397
India	12,570	United Kingdom	1,210
USA	6,080	Iran	1,215
The Netherlands	4,839	Singapore	1,106
Russia	2,910	Vietnam	1,097
Taiwan	2,150	Bulgaria	937
South Korea	2,070	Argentina	937
Germany	1,749	Japan	858
Brazil	1,608	Sweden	818
Hong Kong	1,541	Türkiye	492

To answer **RQ1: Were there any actors exploiting specific vulnerabilities related to VSAT environments?**, we search in our logs for (a) access to the Web and Telnet services via the default credentials (CVE-2018-5266) and (b) direct access to the status of the VSAT via the `getSysStatus` endpoint (CVE-2018-5728). Until the writing of this paper, only one such attempt has been recorded in the login page of Web dashboard. On the contrary, generic attempts

Table 5: The 10 most used credential combinations in Telnet.

Username	Password	Times
admin	1234	1,178
root	aquario	1,010
root	admin	962
root	root	686
root	(empty)	670
root	hi3518	666
admin	admin	634
admin	password	632
ubnt	ubnt	630
admin	ujMko0admin	624

with login credentials, such as those in Table 5, have been logged in the Telnet service.

The role “User” has been used in one Telnet login attempt, that however had the combination of “*username=User,password=1001*” and “*username=User,password=User*”, which are invalid in the scope of this honeynet. In the attempt to access the Web dashboard, a single IP address used the credential combinations of “*username=(empty),password=(empty)*”, “*username=Dealer,password=seatel-2*”, “*username=dealer,password=seatel2*”, “*username=user,password=seatel2*” and “*username=User,password=seatel1*”. Only the last one is a valid combination providing access to the Menu interface of the Web dashboard. The only records for accessing the *getSysStatus* endpoint indicate that those attempts have been made after the authentication, and not by a direct access.

Among all the requests, we received 19,334 GET and 644 POST actions. Only 8,950 GET and 228 POST requests attempted to access endpoints served by our Web service. Besides those of the single successful login, the majority of the requests revolve mainly around the Login endpoint, which serves the landing page of the VSAT. Many others such as the GET request to */cgi-bin/iptest.cgi?cmd=iptest.cgi&-url=%60wget+http%3A%2F%2F209.200.246.240%3A8081%2Fmeow%60&-time=%!(NOVERB) HTTP/1.1*), or the POST one to */cgi-bin/account\_mgr.cgi*, indicate generic efforts to exploit potentially vulnerable services.

For **RQ2: What were their interactions with Web dashboard and CLI environments?**, we observe that a number of 196 IP addresses have attempted to access both services. Nevertheless, only logs from the Web dashboard exhibits some form of interaction, after the single successful login recorded. This constitutes of configuring the satellite (*/ConfigSat.html*), setting the antenna parameters (*/cgi-bin/setAntParams*), setting the position of the ship (*User-ShpPosSet.html*) viewing and exporting data from the VSAT (*/Viewlog.html* and */DataExport.html*), and attempting to access the Menu for the “Dealer” (*/MenuDealerGX.html*). Since there are no successful logins to the Telnet service, there are no meaningful interactions that can be correlated between the two honeynet services.

Regarding **RQ3: Were there any persistence mechanisms attempted?**, such as change of passwords, configuration or drop of modified firmware, no such attempts were made. There was only one attempt, in the Web service, to escalate privileges to the user “Dealer”, by attempting to directly access the corresponding Menu page while being authenticated as “User”. This lack of usage of persistence mechanisms could be attributed to the fact that, besides the change of the password of the user “User”, all the other endpoints that could potentially enable persistence mechanisms to be established, require the user to first be authenticated with a properly privileged account.

Overall, the above indicate only one attack from a curious party that is somewhat familiar with the particular VSAT system and its vulnerabilities, that is still exploring the nature of such systems. In this attack, the actor was knowledgeable about the user roles and their privileges of the VSAT, but only attempted some basic actions via the account with the lowest permissions. We intend to continue collecting data from our honeynet over the course of the next year. This could enrich the answers to the above RQs, and can provide us with a better understanding of any further adversaries that specifically look for and target VSAT systems.

### 4.3 Limitations

In the current implementation, Salty Seagull is limited to the simulation of a VSAT communication system, with no additional shipboard components integrated. Expanding the honeynet to include other critical subsystems, such as propulsion control and navigation systems, along with vulnerabilities that facilitate lateral movement, could significantly enhance the realism and depth of the deception environment. Nevertheless, the present configuration effectively illustrates how the compromise of an insecure communications system could serve as an initial entry point for adversaries targeting maritime assets.

It is important to note that a sophisticated adversary may attempt to cross-reference the replayed VDR information, such as latitude and longitude coordinates, with external maritime tracking data to assess the authenticity of the simulated vessel. As a potential enhancement, future work could involve the integration of live voyage data from operational commercial vessels. However, acquiring such real-time data introduces substantial challenges, particularly concerning privacy regulations and operational security constraints.

Finally, we acknowledge that the credibility and discoverability of the honeynet could be further strengthened by situating it within a network infrastructure associated with satellite telecommunications. Deploying the system over low Earth orbit satellite networks, such as Starlink, OneWeb, or Amazon’s Project Kuiper, may more convincingly reinforce the perception that the emulated VSAT system belongs to a legitimate maritime platform, thereby improving the effectiveness of the overall deception strategy.

## 5 Conclusions

To investigate the threat landscape associated with VSAT systems aboard maritime vessels, we developed a specialized VSAT honeynet designed to attract potential attackers targeting shipboard communication infrastructure. To replicate the operational characteristics of such CPS environments, we implemented both a Web-based dashboard and a CLI, each populated with simulated voyage data. The major findings of this work demonstrate that despite numerous attempts using generic exploits, only a single adversary demonstrated awareness of the system’s nature and associated vulnerabilities, successfully gaining access and performing initial reconnaissance within the environment. This suggests that meaningful exploitation of such systems requires specific domain knowledge, underscoring the complexity of compromising a ship’s network. Future improvements could include the expansion of the honeynet with more elaborate ship services and components to increase emulation fidelity.

**Acknowledgement.** We sincerely thank the anonymous reviewers for their insightful comments and valuable suggestions. We would like to acknowledge the students participating in the Hack@Sea Minor course in 2024 at NHL Stenden University of Applied Sciences, for their assistance regarding parts of the web development used for this work.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

**Data Availability.** The anonymized version of the data collected in this work are made available at <https://doi.org/10.5281/zenodo.15469996>.

## References

1. AG, D.T.: T-pot: A multi-honeypot platform. Honeynet Project (2025)
2. Akpan, F., Bendiab, G., Shiales, S., Karamperidis, S., Michaloliakos, M.: Cybersecurity challenges in the maritime sector. *Network* **2**(1), 123–138 (2022). <https://doi.org/10.3390/network2010009>, <https://www.mdpi.com/2673-8732/2/1/9>
3. Brouwer, S.: HoneyShip: A Maritime VSAT Honeypot to Collect Cyberattacks and Analyze Threats. Master’s thesis, Rijksuniversiteit Groningen, 9712 CP Groningen, Netherlands (2024)
4. Cheswick, B.: An evening with berferd in which a cracker is lured, endured, and studied. In: *Proc. Winter USENIX Conference*, San Francisco. pp. 20–24 (1992)
5. CYDOME: Lab dookhtegan cyber attack on iranian oil tankers disrupts operations, <https://cydome.io/lab-dookhtegan-cyber-attack-on-iranian-oil-tankers-disrupts-operations/>, accessed 2025-04-27
6. Franco, J., Aris, A., Canberk, B., Uluagac, A.S.: A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems. *IEEE Communications Surveys & Tutorials* **23**(4), 2351–2383 (2021)



7. Hilt, S., Maggi, F., Perine, C., Remorin, L., Rösler, M., Vosseler, R.: Caught in the act: Running a realistic factory honeypot to capture real threats. *Trend Micro Research* (2020)
8. IACS: Recommendations on voyage data recorder, <https://web.archive.org/web/20230202060115/https://iacs.org.uk/download/1871>, accessed 2025-04-01
9. Jiang, X., Wang, X.: “out-of-the-box” monitoring of vm-based high-interaction honeypots. In: *International Workshop on Recent Advances in Intrusion Detection*. pp. 198–218. Springer (2007)
10. Kempinski, S., Ichaarine, S., Sciancalepore, S., Zambon, E.: ICSvartase: A Framework for Purpose-based Design and Classification of ICS Honeypots. In: *Proceedings of the 18th International Conference on Availability, Reliability and Security*. pp. 1–10. ACM, Benevento Italy (Aug 2023). <https://doi.org/10.1145/3600160.3605020>, <https://dl.acm.org/doi/10.1145/3600160.3605020>
11. Koniaris, I., Papadimitriou, G., Nicopolitidis, P.: Analysis and visualization of ssh attacks using honeypots. In: *Eurocon 2013*. pp. 65–72. IEEE (2013)
12. Mahmoud, R.V., Pedersen, J.M.: Deploying a university honeypot: A case study. In: *CEUR Workshop Proceedings*. vol. 2443, pp. 27–38. CEUR Workshop Proceedings (2019)
13. Martin, L., Benson, B.: Ics/ot cybersecurity considerations for maritime transportation (2023), [https://hub.dragos.com/hubfs/116-Whitepapers/Dragos\\_WP\\_ICS\\_OTCybersecuritMaritimeTransp\\_Final%20\(1\).pdf](https://hub.dragos.com/hubfs/116-Whitepapers/Dragos_WP_ICS_OTCybersecuritMaritimeTransp_Final%20(1).pdf), accessed: 2025-04-03
14. MITRE: Cve-2018-5266, <https://nvd.nist.gov/vuln/detail/cve-2018-5266>, accessed 2025-04-03
15. MITRE: Cve-2018-5267, <https://nvd.nist.gov/vuln/detail/cve-2018-5267>, accessed 2025-04-03
16. MITRE: MITRE ATT&CK®, <https://attack.mitre.org/>, accessed 2025-04-03
17. Oosterhof, M.: Cowrie ssh/telnet honeypot, <https://github.com/micheloosterhof/cowrie>, accessed 2025-04-01
18. Pavur, J., Moser, D., Strohmeier, M., Lenders, V., Martinovic, I.: A tale of sea and sky on the security of maritime vsat communications. In: *2020 IEEE Symposium on Security and Privacy (SP)*. pp. 1384–1400. IEEE (2020)
19. Rajaram, P., Goh, M., Zhou, J.: Guidelines for cyber risk management in shipboard operational technology systems. In: *Journal of Physics: Conference Series*. vol. 2311, p. 012002. IOP Publishing (2022)
20. Raymond, E.S.: Nmea revealed. URL <https://gpsd.gitlab.io/gpsd/NMEA.html> (2019)
21. Rist, L., Vestergaard, J., Haslinger, D., Pasquale, A., Smith, J.: Conpot ics/scada honeypot. Honeynet Project (conpot.org) (2013)
22. Rist, L., Vetsch, S., Kossin, M., Mauer, M.: Know your tools: Glastopf-a dynamic, low-interaction web application honeypot. *The Honeynet Project* **4**, 2 (2010)
23. Rivieramm: Fishing vessel owners turn to vsat, <https://www.rivieramm.com/opinion/opinion/fishing-vessel-owners-turn-to-vsate-35069>, accessed 2025-04-01
24. SeaTel: Document ima cli protocol specification, <https://www.yumpu.com/en/document/read/50984924/document-ima-cli-protocol-specification-livewire-connections-ltd>, accessed 2025-04-03
25. Spitzner, L.: Honeypots: tracking hackers. Addison-Wesley Longman Publishing Co., Inc. (2002)
26. Spitzner, L.: The honeynet project: Trapping the hackers. *IEEE Security & Privacy* **1**(2), 15–23 (2003)

27. Srinivasa, S., Pedersen, J.M., Vasilomanolakis, E.: Deceptive directories and “vulnerable” logs: a honeypot study of the ldap and log4j attack landscape. In: 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). pp. 442–447. IEEE (2022)
28. Stoll, C.: The cuckoo’s egg: tracking a spy through the maze of computer espionage. Simon and Schuster (1989)
29. Tambe, A., Aung, Y.L., Sridharan, R., Ochoa, M., Tippenhauer, N.O., Shabtai, A., Elovici, Y.: Detection of threats to iot devices using scalable vpn-forwarded honeypots. In: Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy. pp. 85–96 (2019)
30. Tools, D.: Web honeypot, <https://github.com/DinoTools/dionaea/>, accessed 2025-04-01
31. transmitterdan: Vdrplayer - play voyage data recorder files over ip link., <https://github.com/transmitterdan/VDRplayer>, accessed 2025-04-01
32. Willbold, J., Schloegel, M., Bisping, R., Strohmeier, M., Holz, T., Lenders, V.: Vsaster: Uncovering inherent security issues in current vsat system practices. In: Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks. pp. 288–299 (2024)