Attacking (and Defending) the Maritime Radar System

Giacomo Longo, Enrico Russo, Alessandro Armando, and Alessio Merlo, *Senior Member, IEEE*

***Abstract*— The operation of radar equipment is one of the key facilities navigators use to gather situational awareness about their surroundings. With an ever-increasing need for always-running logistics and tighter shipping schedules, operators rely more on computerized instruments and their indications. As a result, modern ships have become complex cyber-physical systems in which sensors and computers constantly communicate and coordinate. In this work, we discuss novel threats related to the *radar system*, one of a ship's most security-sensitive components. In detail, we first discuss some new attacks capable of compromising the integrity of data displayed on a radar system, with potentially catastrophic impacts on the crew's situational awareness or safety. Then, we present a detection system to highlight anomalies in the radar video feed, requiring no modifications to the target ship configuration. Finally, we stimulate our detection system by performing the attacks inside a simulated environment. The experimental results indicate that the attacks are feasible, easy to carry out, and hard to detect. Moreover, they prove that the proposed detection technique is effective.**

***Index Terms*— Radar equipment, network security, marine navigation.**

I. Introduction

CONDUCTION of a vessel increasingly depends on the integration of Information Technology (IT) and Operational Technology (OT). The advantages are great: on the one hand, OT enables a reduction of costs and the execution of risky tasks by the crew through the automation of onboard operations associated with the mechanical and electrical subsystems. On the other hand, IT, and - more generally - Information and Communication Technologies (ICT), provide invaluable support to navigation planning, control, and monitoring. Commercial ships undertaking international voyages are subject to multilateral treaties mandating the installation of various electronic devices [1]. Combined with initiatives promoted by the International Maritime Organization (IMO), e.g. e-navigation [2], [3], such provisions have led to significant onboard system digitization, mostly based on a complex integration of several digital components. Among the others, the *Integrated Navigation System* (INS) lies at the core of this digitization. By gathering information and integrating functions from various electronic devices (e.g., the *radar*), the INS helps the operator plan, monitor, and control the navigation and contributes to improving the overall situational awareness [4]. During navigation, the *radar* plays a key role in forming the crew's situational awareness and thus in dealing with ship encounter situations and in the decision-making for collision avoidance [5]. Through the *Automatic Radar Plotting Aid* (ARPA) [6], the radar can automatically detect and calculate other ships' trajectories. Integration between the radar system and the INS components is supported by a navigation network and by leveraging two standard network protocols: NMEA 0183 [7] and ASTERIX CAT-240 [8]. The former enables interaction among all devices, while the latter supports video data transmission between the radar antennas and the displays.

From a cybersecurity standpoint, the key problem of the previous scenario is the assumption that all components' environment is trusted. For instance, NMEA and ASTERIX protocols assume that the navigation network and interconnected subsystems are reliable, and consequently, they do not envisage any cryptographic protection. Therefore, while such technologies improve the safety and effectiveness of navigation, integrating heterogeneous protocols and technologies on untrusted networks may lead to unexpected cyber attacks. For instance, Meland et al. [9] presented an overview of 46 maritime cyber security incidents in the last decade (2010-2020). While the overall number of cyber-attacks may appear relatively small compared to other sectors, their impact can be very high as they may directly affect the safety of people and the ship. Moreover, security vulnerabilities may be either harder or slower to fix. For example, a well-known security weakness of NMEA, associated with the INS [10], is still to be fixed because retrofitting the INS is expensive and time-consuming.

On the bright side, it is worth underlining that launching a successful cyber attack against a ship is more difficult than other sectors, i.e., INSs are typically offline, penetrating them through lateral movements from other networks and controlling an attack from the Internet may not be an option. Additionally, both the individual components and the configuration of the INS may vary from ship to ship. For these reasons, we argue that to increase the likelihood of a successful cyber attack, it is necessary to develop malware that meets the following requirements.

Manuscript received 16 August 2022; revised 8 May 2023; accepted 30 May 2023. Date of publication 1 June 2023; date of current version 21 June 2023. This work was supported in part by SERICS under the NRRP MUR program funded by the EU-NGEU under Grant PE00000014 and in part by Fincantieri E-phors S.p.A. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alexey Vinel. *(Corresponding author: Enrico Russo.)*

Giacomo Longo, Enrico Russo, and Alessandro Armando are with the Department of Informatics, Bioengineering, Robotics, and Systems Engineering (DIBRIS), University of Genoa, 16145 Genoa, Italy (e-mail: giacomo.longo@dibris.unige.it; enrico.russo@unige.it; alessandro.armando@unige.it).

Alessio Merlo is with the Centre for Higher Defense Studies (CASD), 00165 Rome, Italy (e-mail: alessio.merlo@ieee.org).

Digital Object Identifier 10.1109/TIFS.2023.3282132

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

R-1 Autonomous. It exhibits a high degree of autonomy in pursuing its objectives without external command-and-control servers or human support or guidance.

R-2 INS-targeted. It targets configurations required by international regulations on the design of INSs (see Section II-A) and their standard protocols, not demanding any additional prerequisites to function correctly.

R-3 Stealthy. Its behavior is hard to detect by monitoring endpoints or network traffic, i.e., it requires a moderate use of the CPU, memory resources, and network bandwidth to avoid detection. Moreover, —for the most sophisticated types of attacks— its effects are difficult to detect by the crew, i.e., it manipulates the radar display with realistic updates and is not exposed to cross-checking [11] with other situational awareness equipment.

This paper investigates the potential for developing such advanced attacks on the maritime radar system and proposes a method for detecting them.

We summarize our contributions as follows.

- We demonstrate that it is viable to develop malware incorporating **R-1**, **R-2**, and **R-3** for the purpose of targeting the radar system.
- By testing a mainstream commercial radar solution, we show that the inherent characteristics of the ASTERIX protocol can potentially provide malware with novel techniques for exploiting vulnerabilities or gaining strategic advantages.
- We present a network monitoring technique that detects such and unknown attacks against the radar system. It runs without requiring any changes to the existing INS configuration.
- We perform extensive experiments to evaluate the feasibility of the attacks and the effectiveness of our detection system.

A. Structure of the Paper

The paper is structured as follows. In Section II, we recall some preliminary notions. In Section III, we introduce the threat model and the attack techniques. In Section IV, we describe novel attacks exploiting the above techniques, and in Section VI, a system to detect them. In Section VII, we demonstrate the feasibility of the attacks and evaluate our detection system. In Section VIII, we review the related work. Finally, we conclude the paper in Section IX.

II. Background

In this section, we recall the relevant notions for correctly understanding the paper's content.

A. Bridge Network

On a ship, the bridge network connects sensors and equipment. Its typical configuration follows a homogeneous integration pattern in which multiple devices receive, process, and visualize data exchanged in a shared Ethernet network [10], where any connected endpoint can listen and add its messages to all broadcasted traffic. Similarly, any device can discover, listen and communicate with multicast flows via the standard IGMP protocol [12]. The main aim is to create a system, namely an Integrated Navigation System (INS), that promotes data fusion and synergy between different equipment operating independently.

Data mainly comes from a collection unit connecting onboard sensors and forwarding generated values to the network. Examples of such sensors are the Electronic Position Fixing System (EPFS), the Speed and Distance Measurement Equipment (SDME), the Compass, the Gyroscope, and the Automatic Identification System (AIS) [13] transponder (see below). Moreover, the navigation network hosts at least the two most essential navigational equipment: the radar system (see Section II-B) and a specialized digital navigation computer, namely the Electronic Chart Display and Information System (ECDIS).

The integration [4] is provided by the NMEA 0183 standard [7], i.e., an electrical and data exchange format between maritime electronics. NMEA leverages messages (or sentences) that include a start character followed by comma-delimited fields and a simple checksum terminated by a two-byte delimiter. *Talker* sentences are a type of message containing a two-letter talker identifier, a three-letter sentence type, and a variable number of fields.

For example, the talker sentence `$HETHS,33.2,A*1F` represents a message emitted by the gyroscope (`HE`), with a sentence type related to the true heading and status (`THS`), indicating a sensor heading measurement of 33.2°, sent automatically (`A`) and having a checksum of $\texttt{1F}_{16} = 31_{10}$.

NMEA also provides talker sentences for AIS.

AIS is a standard system for enhancing safety, e.g., reducing the risk of collisions, by exchanging information between ships and maritime authorities. For instance, ships periodically broadcast position reports that indicate their current course and speed. Reception and transmission occur over Very High Frequency (VHF) radio data links, and a converter forwards them from VHF to the INS network (and vice-versa) using `VDM` and `VDO` NMEA sentences. Such sentences allow INS equipment to integrate the received information, e.g., radar plotters can associate their targets with AIS data [14].

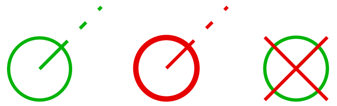
From a security standpoint, NMEA and AIS have severe issues since they do not support message authentication or provide solutions to exchanging information confidentially.

B. RADAR System

RAdio Detection And Ranging (RADAR) is a system that can detect surrounding objects using radio waves.

The whole radar system relies on different devices, but we consider only the two main ones: an *antenna* unit and a display unit, namely the *Plan Position Indicator* (PPI).

An antenna rotates 360 degrees about its vertical axis, radiating waves and receiving returning echoes from targets. Shipborne radars employ this mode of operation, known as *Primary Radar* [15, §1.101]. This contrasts with other fields, e.g. air-traffic control [16], leveraging *Secondary Radar* in which targets emit signals to be received by the antenna.



(a) Acquired (b) Dangerous (c) Lost

Fig. 1. ARPA target symbols on the PPI.

Each antenna has its specifications that differ between manufacturers and include the *rotation speed* and a resolution related to the *bearing* and *range*. The rotation speed specifies the speed at which the motor rotates an antenna. The bearing resolution, or *angular* resolution, determines the ability to separate targets at the same distance and close together. The range resolution determines the ability to resolve between two targets in the same direction but at slightly different distances.

The PPI is a circular display representing the antenna, with the ownship in the center. A radial trace sweeps in unison with the radar antenna around the central point. Each trace represents echo signals in plan position with bearing and range displayed in polar coordinates. Officers can configure the top of the display to represent different perspectives. In the *head-up* mode, the zero of the PPI represents the own ship's course, and the bearing of the displayed targets will be relative to its heading. In the *north-up* mode, the zero represents the true north, a heading marker represents the true course of the own ship, and all bearings of targets are actual.

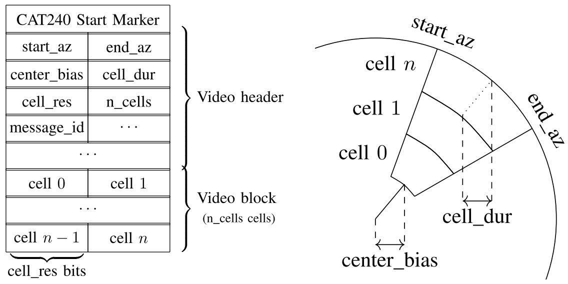
Digital PPIs must emulate the behavior of traditional radar scopes. In particular, every echo received must persist on the PPI for at least the time of half a rotation [17]. Moreover, standard regulations state that if a PPI receives multiple traces for the same rotation angle during the persistence time interval, it has to sum their echoes [18, §15.6.3.2.e].

Digital PPIs also add new capabilities over traditional radar scopes. For example, the echo *trail* allows officers to visually understand the movements of other ships, i.e., path and speed, by displaying a residual image at different times of an echo.

Radar systems can automatically provide an accurate estimate of such movements when they support Automatic Radar Plotting Aid (ARPA) [1, V§2.8]. Radar plotting allows a radar officer to follow a target over time, reconstructing its trajectory w.r.t. the own ship, estimating its course, speed, and range at the closest point of approach (DCPA), and the predicted time to CPA (TCPA). It is worth noting that when TCPA is a negative number, it signals an increasing trend, i.e., the target CPA is getting further from the ship.

An officer can acquire a target manually or automatically when it enters in configurable *acquisition zones*. It becomes acquired after it persists for 5 out of 10 consecutive scans [14, §3.3.3].

Once the acquisition occurs, the radar system tries to follow the movement of targets inside the image. Using EPFS and SDME sensors data allows the ARPA system to estimate a trajectory. If the estimation is successful, a target becomes a *tracked target* [18, Ann.G] and appears on display with the symbol depicted in Figure 1a. ARPA constantly evaluates the CPA and the TCPA status of each tracked target. Acquired targets that move inside the *guard zone*, i.e., a zone configured with a given radius (CPA) and time threshold (TCPA), generate an alarm and appear on display as dangerous (see Figure 1b). Finally, a tracked target is judged as lost when no return is received for nine consecutive scans and appears as depicted in Figure 1c. Within INS, the PPI propagates ARPA information via NMEA, e.g., using `TTM` (Tracked Target Message) sentences.



(a) ASTERIX packet (b) Presentation

Fig. 2. Correspondence between ASTERIX and the PPI.

C. ASTERIX

An antenna can transmit to the PPI via network using proprietary solutions [19], [20] or standards from ASTERIX [21].

ASTERIX is a suite of standard protocols for data exchange of radar information between systems proposed by EUROCONTROL. It contains a collection of message types, called *categories* or *CAT*, where *CAT-240*, namely Radar Video Transmission [8], identifies the one used to transfer video data from antennas to PPI.

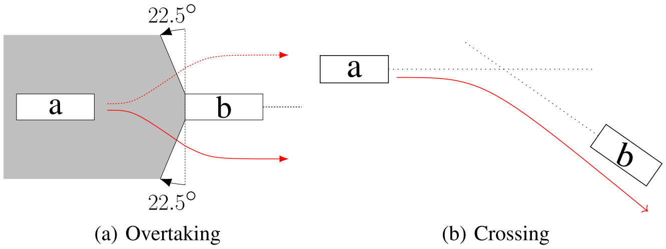
It is crucial to note that while the transfer of processed trajectories and acquired target information, also known as *acquisitions*, is permissible in other ASTERIX categories such as 048 [22] and 062 [23], CAT-240 solely provides its recipients with a polar video stream that must be analyzed by companion systems such as the ARPA.

Since 2009, radar manufacturers have adopted ASTERIX CAT-240 as the de-facto network video standard [24].

As sketched in Figure 2, each CAT-240 message combines a header and a video block and is related to an angle span. The header provides information about the block and metadata like time of day or the *System Identification Code* and *System Area Code* (SIC/SAC) that identify the transmitting antenna. Once decoded, the video block is a sequence of cells located on a polar coordinate system centered around the position of the transmitting antenna. The angle span is between *start_az* and *end_az*. Cells indicate the echo strength quantified using *cell_res* bits. Moreover, each cell starts at a distance ρ. It can be calculated by leveraging their homogeneity among the distance direction as $\rho = D \cdot (b + i) \cdot c/2$ where D and b are included in the header and represent the *cell duration* parameter and the *center bias*, respectively, while i is the cell index (0-based), and c is the light celerity.[1]

Regarding antennas' resolution (see Section II-B), the bearing resolution determines the minimum span between *start_az* and *end_az* while the range resolution determines

[1] Defined as 299792458 m/s in [8].



(a) Overtaking (b) Crossing

Fig. 3. Illustration of COLREGs situations.

the minimum *cell_dur*. Lastly, *message_id* is a sequence number used by the receiver to reorder packets.

From a security standpoint, as emphasized in [25] and [26], the ASTERIX protocol does not implement any authentication and encryption features.

D. COLREGs

In maritime navigation, vessels should obey the International Regulations for Preventing Collisions at Sea, namely COLlision REGulations (COLREGs), agreed to by the IMO in 1972 [27]. These rules specify maneuvers that ships must take in situations where a risk of collision occurs, also employing information from radar and ARPA.

This work addresses two COLREG rules: *overtaking* (rule 13) and *crossing* situations (rule 15).

In particular, rule 13 states that *"a vessel shall be deemed to be overtaking when coming up with another vessel from a direction more than 22.5 degrees abaft her beam"*. Figure 3a illustrates an overtaking situation. In such a situation, the vessel *a* must overtake *b*, and common practice on the water dictates that the overtaking boat should pass on the right-hand side of the slower vessel *b*.

Rule 15 states that *"when two power-driven vessels are crossing to involve risk of collision, the vessel which has the other on her own starboard side shall keep out of the way and shall, if the circumstances of the case admit, avoid crossing ahead of the other vessel"*. Figure 3b illustrates a crossing situation where the vessel *a* is in a collision course with *b* and must veer to its starboard so it does not cross ahead of *b*.

III. Attack Techniques

In Sections II-C and II-A, we highlight that ASTERIX and NMEA protocols do not support confidentiality and authentication between communicating components. Moreover, modern ships are equipped with an INS where such components communicate through multicasts or broadcasts, allowing anyone connected to overhear the exchanged packets. Protocols and INS configuration represent the attack vector.

A coarse-grained attack can merely inject ASTERIX packets with false echoes to hide or corrupt the image displayed on the PPI. This attack requires a little effort, generates the malfunction of an essential system for navigation, and yields a significant impact. In particular, it can pose moderate risks to the ship's operations and safety and could force the start of emergency procedures to return the vessel to port.

Nevertheless, we consider a novel adversary capable of executing *fine-grained* attacks. A fine-grained attack does not create malfunctions but alters the radar information without detection. It actively monitors the ship's status and only activates when potentially dangerous situations can happen. After it activates, it can operate in real-time on specific areas of the radar image, and the changes appear realistic to the operators. During the attack, it generates an amount of network traffic that does not appear anomalous compared to the one generated during the regular operation of the radar system. Moreover, it must perform all the above operations leveraging resources of INS components that may be limited in computing resources and with different hardware and operating systems. This attack relies on a deep and specific knowledge of its domain. It can pose a severe or catastrophic adverse effect, e.g., harm to individuals, major damage to the vessel and environment, and major financial loss.

The sections below present the assumptions under which adversaries operate and their techniques to perform from coarse-grained to fine-grained attacks.

A. Threat Model

This work focuses on stealthy malware that accesses the ship's navigation network to perform malicious activities. The attacker's goal is to reduce the *situational awareness* of the ship officers to cause a disruption in operation or significantly increase the probability of a safety-critical incident.

We assume that the ship under attack is equipped with an INS hosting a radar system compliant with regulations, performance standards, and behaviors described in Section II-B.

Although some INS possess external connection capabilities [10], we assume that the security of the navigation network is enforced with a restrictive policy, i.e., physically disconnected from other networks, including the Internet.

1) Attacker's Requirements: We consider adversaries as professional actors capable of gathering solid knowledge for generating or testing a novel attack. The framework for Maritime Cyber-Risk Assessment (MACRA) [28, T.1] models such adversaries as *Tier*$_3$ attackers. Such attackers have the ability and resources to install the malware by leveraging the maintenance operations [10] or the supply chain compromise technique [29, T1195]. Moreover, they can also leverage USB devices [30, p.32, 36] or vulnerabilities that bridge workstations suffer from [31], [32], and [33].

Once installed, the malware must operate stealthily and under the assumption that the network is isolated from the Internet. A traditional malware that drops additional malicious payloads and requires a command and control server is out of scope. Instead, the attack requires a targeted malware [34] that can operate autonomously and exploit the specific technology environment.

2) Attacker's Capabilities: Under the above assumptions, the adversary has different capabilities as follows. Since the malware runs on a host connected to the navigation network, it can overhear the cleartext NMEA and ASTERIX packets like any other INS component. NMEA traffic allows the malware to reconstruct and update the ship's state under attack by monitoring sensor devices and ARPA data. For example, it can monitor its position, bearing, speed, nearby vessels, or targets acquired by radar operators. ASTERIX traffic allows the malware to know what the PPI is displaying.

Furthermore, the malware can impersonate legitimate sensor devices and radar antennas by leveraging the lack of authentication in such protocols. For example, the malware can inject NMEA packets holding sentences with fake values from the Compass or the AIS transponder. The injected NMEA packets appear as legitimate data to NMEA devices. Likewise, the malware can inject ASTERIX packets holding messages with fake echoes to hijack the radar system.

We discuss radar hijacking techniques in the section below.

B. Radar Hijacking Techniques

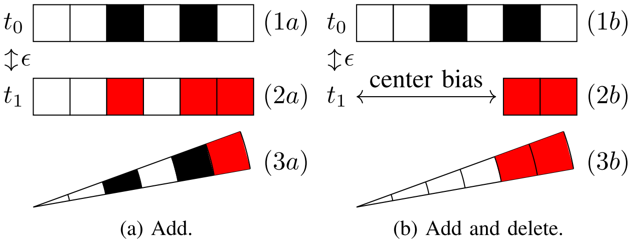
An adversary executes a radar hijacking attack to obtain the capabilities to *add* and *delete* targets on the PPI. Under the assumption that the PPI behavior follows standard regulations, we remind that it must satisfy two conditions (see Section II-B): (i) echoes must persist for at least the time of half a rotation, and (ii) if it receives a packet that overrides echoes during their persistence time, the PPI must *sum* old and new values.

Radar hijacking attacks leverage the injection of fake ASTERIX packets to modify echoes *immediately after* the PPI receives the actual values from the legitimate antenna. As a result of the two conditions above, the PPI always sums the fake and actual values. It is worth noting that the behavior from standard regulations restricts an attacker only to increase the strength of existing echoes, thereby only enabling the capabilities to *add* targets.

We experimented with the above restriction on the commercial radar of our testbed. Our tests try to delete the radar image by injecting the original packets after we update them with zero-strength echoes. The results showed that the PPI complies with the standard regulations as it sums values and prevents deleting the radar image.

In Figure 4a, we show an example of an attack that the malware can exploit using only the capability to add a target. For the sake of simplicity, we consider ASTERIX packets carrying a video block of six cells and related to the minimum angle span constrained by the bearing resolution of the antenna (e.g., one degree). We reduce the echo strengths to on/off values. This attack aims to add a fake echo to a trace that the PPI visualizes at a specific azimuth α. We assume that the PPI receives t_0 the ASTERIX message $(1a)$ for the azimuth α from the legitimate antenna at time. Consequently, the PPI visualizes a trace with the two echoes that the message holds in the third and fifth cells. In the meantime, the attacker can overhear $(1a)$ and create a new ASTERIX message $(2a)$ containing original echoes and the fake one in the sixth cell. Then, the attacker can inject the new message $(2a)$ into the navigation network at time $t_1 = t_0+\epsilon$, where ϵ is a small delay due to the attacker's operations. After the PPI receives $(2a)$, it visualizes the new trace $(3a)$ for α that sums the echoes of $(2a)$ with the ones of $(1a)$ cell by cell. Since ϵ is a negligible lag, i.e., in the order of milliseconds, a radar operator will not perceive the update.

To obtain the capability to delete targets, an attacker must create an outlier situation that a PPI handles by violating standard regulations. We obtained such a condition with the commercial radar of our testbed by applying a standard feature of the ASTERIX protocol. We injected packets that differ from the originals in the value of the echo strengths and the number of cells they contain. In particular, we decreased it by shifting the *center_bias* parameter by one in the packets header (see Section II-C). Due to this difference, the PPI under test replaces the displayed echoes with the most recent data of the injected packet, thus allowing us to acquire the capability of deleting existing echoes.



(a) Add.

(b) Add and delete.

Fig. 4. Hijacking techniques.

In Figure 4b, we show an example of an attack using the *center_bias* parameter. This second attack aims to replace the trace generated by $(1b)$ for α with a new one that deletes the echo of the third cell and adds an echo to the sixth cell. To this aim, the attacker creates the message $(2b)$ containing the two cells with echoes and with a shift of four cells from the center, i.e., center bias $= 4$. When the PPI receives such a message, it should keep the values of the first four cells of the visualized track $(1b)$ and sum the last two values of $(1b)$ with the ones of $(2b)$. Instead, the PPI replaces the existing trace with $(3b)$ corresponding to the most recent message $(2b)$, thus deleting the echo in the third cell.

We stress that an adversary can perform an attack that adds a target against any radar system. In contrast, the feasibility of an attack that deletes a target depends on the vendor-specific implementation of the PPI when the outlier situation we introduced above occurs. In Section IV-A, we show that the malware can automatically infer if the PPI under attack suffers from behavior similar to our testbed, allowing attackers the delete capability.

IV. Attack Description

In this section, we detail the inner behavior of the stealth malware that an adversary can exploit to execute an attack on a radar system.

Figure 5 shows its workflow. We map it to three steps that are inspired by the cyber kill chain [35], namely the *reconnaissance*, the *weaponization*, and the *delivery* steps.

We detail them below.

A. Reconnaissance

The *reconnaissance* step starts with the *traffic capture* task that captures the cleartext NMEA and ASTERIX traffic flowing into the bridge network. Then, the *ship state awareness* task analyzes the NMEA traffic to achieve situational awareness. The aim is twofold: detecting if the radar system

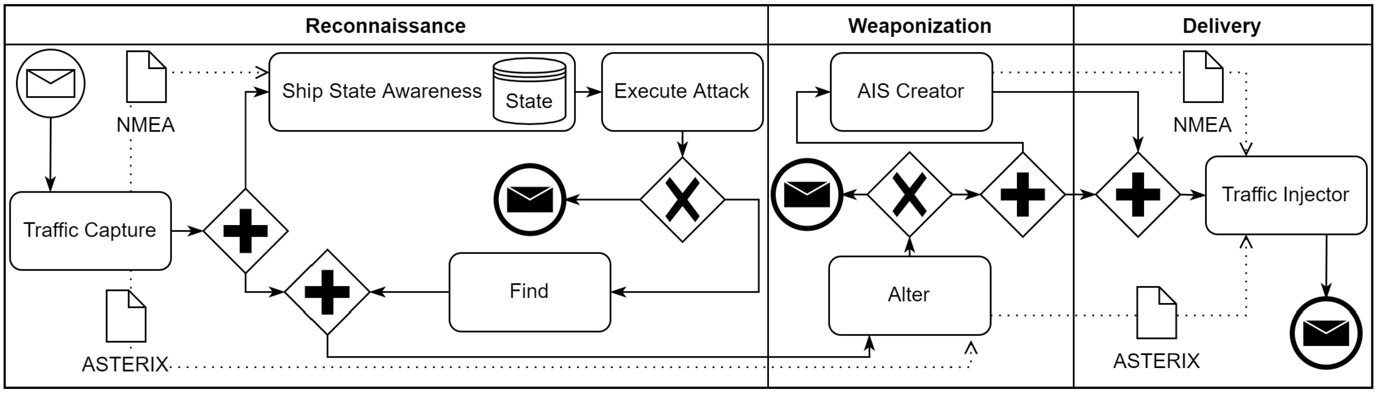


Fig. 5. The workflow of the stealth malware.

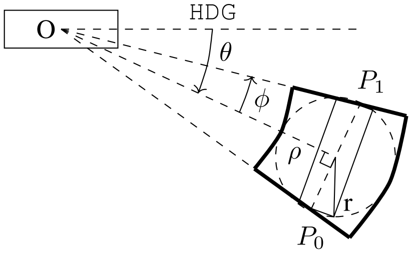


Fig. 6. An example annulus section and related quantities.

under attack allows adversaries to apply the delete capability and keeping updated the *state* of the ship under attack.

Detecting the delete capability requires a single and short test of an attack after the malware starts. At first, the malware listens for the `TTM` sentences from the ARPA system (see Section II-B). When it receives a `TTM`, it uses the position and bearing of the tracked target to execute a delete attack against the corresponding representation on the PPI. After the attack starts, it waits for the time of nine consecutive scans of the PPI. The ARPA system that stops sending `TTM`s for the target under attack or sends `TTM`s that contain a *lost status* means that the PPI grants the delete capability to the malware.

Updating the state requires reading data from NMEA sentences to track the ship telemetry, nearby vessels, tracked targets, and weather conditions.

The *execute attack* task leverages the values of the above state to automate the decision to start the attack (e.g., specific GPS location, the position of nearby vessels, or if it is night).

If an attack requires operating in a specific area of the radar image, a *find* task allows the malware to define the boundaries. It uses a *find* function that we detail below.

This step ends by forwarding the results of the find function and the captured ASTERIX packet to the *weaponization* step.

a) Find function: The find function returns a delimited zone of the radar image representing a given target, e.g., a ship or a waypoint on the ECDIS.

An example of such a zone is highlighted in Figure 6. It is an annulus sector centered on point O that has the latitude and longitude coordinates of the ship under attack and contains the bounding box of the target (bbox). Such a zone can be described with a tuple $\langle a_{min}, a_{max}, d_{min}, d_{max}\rangle$ where a_{min}, a_{max} and d_{min}, d_{max} are its ranges w.r.t. the angular and longitudinal dimensions, respectively.

Algorithm 1 The Algorithm of the Find Function

```
1: function FIND($\rho$, $\theta$, $w$, $h$, $sm_{\%}$)
2:     if $\rho = 0$ then return $\langle 0, 360, 0, \infty\rangle$
3:     $r \leftarrow \frac{\sqrt{w^2+h^2}}{2}$; $r^\star \leftarrow r(1 + sm_{\%})$
4:     $\phi = \text{atan2}(r^\star, \rho)$
5:     $a_{min/max} = C_{360}(\theta \pm \phi)$
6:     $d_{min} = \max\{0, \rho - r^\star\}$; $d_{max} = d + r^\star$
7:     return $\langle a_{min}, a_{max}, d_{min}, d_{max}\rangle$
```

Algorithm 1 represents the find function. In the algorithm, ρ is the distance between O and the bbox center $(0, \infty)$, θ is the bearing of O from the center of the bbox in arc degrees $[0, 360)$, w and h are the width and the height of the bbox $(0, \infty)$, and $sm_{\%}$ is the size margin of the bbox (Line 1). Since the find function is not always required, we consider $\rho = 0$ for returning a zone delimiting the entire radar image (Line 2). Otherwise, we approximate the target shape with a circle inscribing its bbox. A circle allows ignoring the orientation of the target rectangle during the calculation. The circle has a radius r corresponding to the half-diagonal of the bbox and it can also be expanded by a percentage factor $sm_{\%}$, obtaining the final radius $r^\star$ (Line 3). In this way, the buffer zone allows compensating position inaccuracies at the expense of a less precise find zone determination and an increase in the annulus surface area. To calculate the angle span ϕ (Line 4), we observe that a right triangle exists between O, the center of the bounding circle, and one of the two points $P_{0/1}$ tangent to the circumference and passing through O. Observing the symmetry of the problem concerning the vector joining O and the center of the bounding circle, we can calculate the angular range $a_{min/max}$ (Line 5) where $C_{360}(x)$ represents the 360-degrees based complement of x. Finally, knowing the distance ρ from O to the centroid, we can compute the range $d_{min/max}$ (Line 6) and return the tuple for the zone (Line 7).

Example 1: After having received an AIS message (see Section II-A) reporting the latitude (lat_t) and longitude (lon_t) of a target ship, we want to apply the find function for obtaining the tuple of its annulus section on the radar image.

To this aim, we must derive O, ρ, θ, w, and h parameters ($sm_{\%}$ is optional).

O can be acquired from NMEA sentences generated by EPFS systems (see Section II-A), e.g., `GGA`, `GLL`, `GNS`, or `RMB` sentences. We can obtain ρ and θ by calculating the geodesic distance and azimuth, e.g., using Vincenty's inverse formula [36], between O and $\langle lat_t, lon_t \rangle$ and transforming the resulting azimuth to the measuring ship's heading (`HDG`) relative azimuth. `HDG` can be found in sentences originated by compasses or gyroscopes, e.g., `HDT` and `THS` ones. Lastly, w and h can be obtained from AIS, i.e., *ship static and voyage related data* messages containing the target ship's size. □

B. Weaponization

This step starts by receiving return data from the find function and an ASTERIX packet. The *alter* task checks if the packet contains echoes related to the zone bounded by the find function. If so, it calls an *alter* function (see below) for creating a weaponized ASTERIX packet that can modify how the above echoes appear on the PPI.

As the attack may involve adding ghost ships or altering the course and speed of existing targets, the malware can use the *AIS creator* task that generates `VDM` sentences with data reflecting the changes occurring on the radar system. Consequently, INS equipment using AIS will display information consistent with the attack.

1) Alter Function: The alter function allows modifying echoes of an existing ASTERIX packet Pkt. Such an operation is performed by wrapping the execution of a user-provided variadic function $f : (Pkt \times F_o \times F_a) \rightarrow Pkt \bigcup \emptyset$ where Pkt is an existing ASTERIX packet, F_o is the result of the *find* function, and F_a are user-specified arguments belonging to the domain of the function f. Evaluation of alter returns the result of f, i.e., empty or an ASTERIX packet.

Example 2: In Example 1, we obtained the tuple of an annulus section related to a target ship. We want to create a *ghost ship* by copying its image into a different position.

To this aim, Algorithm 2 shows the implementation of a function that can be used with the alter one, i.e., `copy_ship`.

Algorithm 2 The Algorithm of `copy_ship`

```
 1: function COPY_SHIP($Pkt$, $a_{min}$, $a_{max}$, $d_{min}$, $d_{max}$, $o_a$, $o_d$)
 2:     if $Pkt.start\_az \ge a_{min}$ or $Pkt.end\_az \le a_{max}$ then
 3:         $i_o \leftarrow \text{ROUND}(\frac{o_d}{Pkt.cell\_dur \cdot c/2})$
 4:         cells $\leftarrow Pkt.cells$; mod $\leftarrow$ false
 5:         for $i \leftarrow 0, Pkt.n\_cells$ do
 6:             $\rho_{min} \leftarrow Pkt.cell\_dur \cdot (i + Pkt.center\_bias) \cdot \frac{c}{2}$
 7:             $\rho_{max} \leftarrow Pkt.cell\_dur \cdot (i + 1 + Pkt.center\_bias) \cdot \frac{c}{2}$
 8:             if $\rho_{min} \ge d_{min}$ and $\rho_{max} \le d_{max}$
                and $i + i_o \ge 0$ and $i + i_o < Pkt.n\_cells$ then
 9:                 $Pkt.cells[i + i_o] \leftarrow cells[i]$; mod $\leftarrow$ true
10:         if mod then
11:             $Pkt.start\_az \leftarrow C_{360}(Pkt.start\_az + o_a)$
12:             $Pkt.end\_az \leftarrow C_{360}(Pkt.end\_az + o_a)$
13:             return $Pkt$
14:     return
```

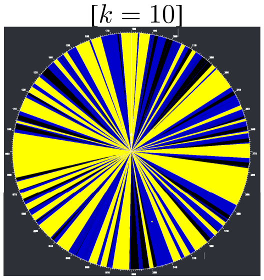
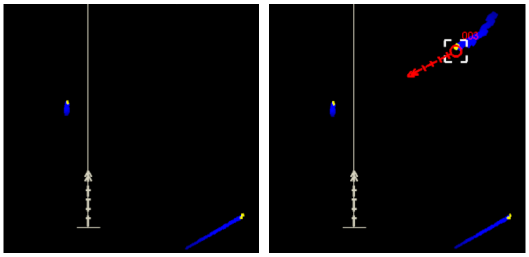


Fig. 7. DoS attack.



(a) Real (b) Attacked

Fig. 8. Comparison of video feeds during V-B1.

In the algorithm, Pkt is the original packet, a_{min}, a_{max}, d_{min}, d_{max} are the values of the tuple, and o_a, o_d are the angle and distance offsets at which the copy should be placed (Line 1). The function changes Pkt only if the start and end angles included in the headers of Pkt, i.e., $Pkt.start\_az$ and $Pkt.end\_az$, are not in the angular range between a_{min} and a_{max} (Line 2). Then, it calculates the cell index distance offset i_o (Line 3) and copies the original video cell contents in a support variable (Line 4). Following, for each cell in Pkt (Line 5), it calculates the minimum ρ_{min} (Line 6) and maximum ρ_{max} (Line 7) covered distances as detailed in Section II-C. If a cell 1) has the covered distance included in the range between d_{min} and d_{max}, and 2) copying its value would not exceed the bounds of the video block (Line 8), the algorithm copies the original cell value into the offset position (Line 9). Finally, the function modifies the packet Pkt azimuthal span returning it (Lines 11-13), otherwise empty (Line 14). □

C. Delivery

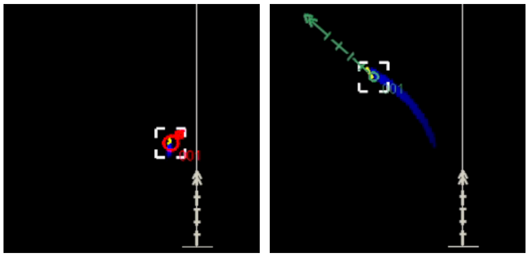
The *delivery* step starts by receiving the weaponized NMEA and ASTERIX packets. The *traffic injector* task injects such packets into the navigation network. As the involved protocols do not support authentication, it forwards them to the multicast or broadcast addresses that INS equipment and the PPI use to communicate in the bridge network. Once the above equipment consumes the weaponized packets, they display the hijacked image and data.

V. Radar Hijacking

This section will discuss two novel classes of attacks for radar hijacking leveraging the previous techniques.

A. Denial of Service Attack

A Denial-of-Service (DoS) attack aims at rendering the PPI unusable and leaving the ship without means of safe



(a) Real (b) Attacked

Fig. 9. Comparison of video feeds during V-B2.

navigation. Briefly, the adversary overlays sectors or the entire azimuthal range of the radar image by filling them with echoes. Below, we detail the steps introduced in Section IV.

1) Reconnaissance: We assume that a radar must continuously operate during navigation. For this reason, the malware does not need to implement specific checks during the *execute attack* task. Nevertheless, favorable conditions exist. For example, they apply when vessels navigate in darkness or congested areas. They can be assessed by overhearing NMEA sentences with the current time and position and AIS information.

Since the attack corrupts the entire display, the *find* task invokes the *find* function with $\rho = 0$ (see Section IV-A).

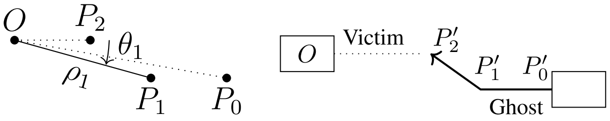
2) Weaponization: In the field of network security, many DoS attacks rely on the misuse of protocols that accept small requests and amplify the volume of traffic to overwhelm a resource of the victim. Protocols with a high amplification factor are the most effective since they require fewer resources to perform the attack and make adversaries harder to trace.

To execute a DoS against a radar system, the misuse of the ASTERIX protocol can enable a high amplification. The angle span and the configurable number and duration of cells (see Section II-C) are the amplification factors we use in the alter function of DoS attacks, namely the *DoS* function. Algorithm 3 represents the implementation of the *DoS* function. It aims to update the received packets with new ones containing echoes at maximum strength and covering the entire angle and distance span. As explained below, *i* and *k* are parameters used for controlling the injection rate.

Algorithm 3 Denial of Service Attack

```
1: function DoS($Pkt$, $a_{min}$, $a_{max}$, $d_{min}$, $d_{max}$, $i$, $k$)
2:     $Pkt.start\_az \leftarrow 0$; $Pkt.end\_az \leftarrow 360$
3:     $n \leftarrow \frac{32}{cell\_res}$; $Pkt.cell\_dur \leftarrow \frac{Pkt.cell\_dur \cdot Pkt.n\_cells}{n}$
4:     $Pkt.n\_cells \leftarrow n$; $Pkt.cells \leftarrow [2^{cell\_res}, \ldots, 2^{cell\_res}]$
5:     $i \leftarrow i + 1$
6:     if $i = k$ then $i \leftarrow 0$; return $Pkt$
7:     return
```

For covering the entire angle span, we set the *start_az* to 0 and *end_az* to 360 (Line 2). We use the minimum number n of cells w.r.t. the constraints the ASTERIX protocol sets for the distance span. This solution creates a video block that is as small as possible. To calculate n and rescale the packet *cell_dur* accordingly, we use the minimum number of bits in a video block, i.e., 32, and the current cell resolution (Line 3). Then, we replace the number of cells in the original packet with n and set each at maximum strength (Line 4).



(a) Stored. (b) Initialized.

Fig. 10. An example trajectory comprised of three points.

Since each altered packet covers a much greater angular span than the original one, adversaries can achieve the desired result without executing an injection at each received packet. They can set k to constrain that an attack happens once every $\frac{1}{k}$ legitimate packets. A high value of k further increases the amplification factor but widens the area of the original image visible between each injection. In the function, i refers to a persistent counter that increments after each call (Line 5). Once i equals k, the modified packet is returned, and i is reset to 0 (Line 6). Otherwise, a null value is returned, indicating that no injection must occur (Line 7).

This attack does not require creating new AIS sentences; the step ends without running the *AIS creator task*.

3) Delivery: Whenever the *DoS* function returns a non-null value, the *traffic injector* task transmits the modified packet to the multicast address of the radar system. In Figure 7, we show how the PPI looks after a DoS attack using $k = 10$.

B. PPI Poisoning Attack

A PPI poisoning attack alters specific sections of the image on the PPI in real time, inducing the crew to make wrong decisions or fail to carry out the required actions.

This class of attacks is especially harmful during navigation in congested waters where the risk of collision is rather high. The danger increases further in restricted visibility since navigation relies on the instruments under attack.

Ships in these risky situations avoid collisions by collectively interacting following COLREGs (see Section II-D). A radar under this attack may lead the victim to assess COLREGs with wrong assumptions. In such a scenario, a vessel behaves differently than expected by others, and the risk of collisions remains high.

In the following, we show two implementations of this type of attack. The first relies only on adding new echoes and can be executed on all radar systems, while the second requires a PPI granting the delete capability (see Section III-B).

1) Ghost Ship: A ghost ship is a fictitious target that this attack adds to the radar image. It appears as changing in time by following a trajectory, i.e., a set of waypoints and speed pairs. Below we describe each step of the attack.

a) Reconnaissance: Malware can keep a list of trajectories in the *state* database of the *ship state awareness* task. As an example, we consider the trajectory that is represented in Figure 10a. It comprises the three points P_0, P_1 and P_2 to be undertaken at a constant speed $S_0 = S_1 = S_2$. Each point is specified in a polar coordinate system w.r.t. an origin point O.

We program our malware to use this trajectory to reproduce a COLREG crossing condition (see Section II-D) and force the victim to perform an unexpected evasive maneuver in a congested area. To this aim, the *execute attack* task overhears AIS and ARPA sentences and triggers the attack when (i) at least 2 ships within a 6 *nm* radius are present, and (ii) no ships are already present in the starboard bow area of the victim.

Once triggered, the malware initializes the waypoints P'_0, P'_1, and P'_2 to create the crossing situation as depicted in Figure 10b. It uses the victim position as the origin O and obtains their absolute coordinates (lat, lon) by using a geodesic formula (e.g., see [36]). The *find* task manages the evolution of the ghost ship's position along the initialized trajectory once every Δt time. In particular, it applies the system of equations detailed below to generate a realistic behavior.

$$\underline{x}(t+\Delta t) = v(\underline{x}(t),\ COG(t),\ SOG(t) \cdot \Delta t) \tag{1}$$

$$\Delta C(t) = C(t) - COG(t) \tag{2}$$

$$\omega(t) = \Omega \cdot sgn \begin{cases} \Delta C(t) + 360 & \text{if } \Delta C(t) < -180 \\ \Delta C(t) - 360 & \text{if } \Delta C(t) > 180 \\ \Delta C(t) & \textit{otherwise} \end{cases} \tag{3}$$

$$COG(t+\Delta t) = C_{360}(COG(t) + \omega(t) \cdot \Delta t) \tag{4}$$

$$a(t) = A \cdot sgn(S(t) - SOG(t)) \tag{5}$$

$$SOG(t+\Delta t) = SOG(t) + a(t) \cdot \Delta t \tag{6}$$

The task at a time t_0 initializes $\underline{x}$ to P'_0, $C(t_0)$ and $COG(t_0)$ to the bearing between P_0 and P_1, $S(t_0)$ and $SOG(t_0)$ to S_0. $C(t)$ changes according to the closest points of the trajectory. Ω and A are two constants constraining the maximum rotation speed and acceleration for the ghost ship.

The position $\underline{x}$ evolves according to the current course and speed by using a geodesic destination formula v (Eq. 1). Eq. 2 and Eq. 3 calculate the angular velocity ω. It is an on-off feedback control for the COG variable w.r.t the target COG C. Then, COG rotates according to ω (Eq. 4). C_{360} is the function detailed in Section IV-A0.a. The acceleration a is an on-off feedback control for SOG w.r.t. the target $S(t) = S_0$ (Eq. 5). Finally, SOG accelerates according to a (Eq. 6).

The *find* task ends by calling the find function with the $\underline{x}$ returned by the above system.

b) Weaponization: In this step, the *alter* task has to draw the ghost ship according to the annulus section returned by the *find* function. It can leverage an implementation of the *alter* function similar to Example 2. For brevity, we omit an in-depth description of the raster algorithm used for drawing the ghost ship image onto the cells.

Finally, the *AIS creator* task uses $\underline{x}$, COG, and SOG from the *FIND* task to synthesize the corresponding `VDM` sentence.

c) Delivery: The *traffic injector* injects weaponized AIS and ASTERIX packets. As a result, NMEA devices show the position of the ghost ship as real and PPIs display the video feed as in Figure 8b instead of the real one as in Figure 8a. We set the PPI in head-up mode, and we enable trails (see Section II-B). In both cases, the PPI displays two real targets, and in Figure 8b, it also shows the ghost ship to the starboard bow of the victim. In particular, the ghost ship's trail resembles the trajectory represented in Figure 10b. Moreover, the radar system acquires the ghost ship as a valid target, and ARPA marks it as a dangerous one (see Section II-B).

The results above show that the malware can reproduce the conditions luring operators to execute an evasive maneuver.

2) Ship Trajectory Hijack: A ship trajectory hijack exploits the delete capability to modify the trajectory of an existing target in the radar image. As an example, we consider the victim in an overtaking situation (see Section II-D). The adversaries aim to modify the trajectory of the vessel being overtaken so that no evasive maneuvers seem to be required.

Briefly, the attack requires deleting the real target's echo and adding a ghost ship with the new trajectory.

a) Reconnaissance: The *execute attack* task overhears AIS and ARPA sentences and triggers the attack when a target (i) goes at a slower speed w.r.t. the victim, and (ii) has an angle between its beam and the victim bow of at least 22.5°. After triggering, the attack initializes a hijacked trajectory T as the one depicted in Figure 10, but exchanges the order between P_0 and P_2. T has speeds $S_0 = S_1 = S_2$ set to a value exceeding the victim's one.

The *find* task executes two find operations: the first returns the annulus section of the overtaken ship (see Example 1), and the second resembles the ghost ship attack using T.

b) Weaponization: This step invokes two implementations of the alter function according to the results of the two find functions above. The first takes the annulus section of the overtaken ship as input and deletes its echoes. Its implementation relies on setting the echo strengths to 0 in the annulus section and altering the *center_bias* value to force the PPI to replace echoes (see Section III-B). The second follows the implementation as in the ghost ship attack.

Finally, the *AIS Creator* task generates `VDM` sentences according to the modified trajectory.

c) Delivery: During this attack, the weaponized AIS messages have to coexist with the real ones. A solution to make the malicious one prevail needs that the *traffic injector* task injects them at a time interval less than 2s, i.e., less than the one set in the standard (see [13]).

Poisoned PPIs display the video feed as in Figure 9b instead of the real one as in Figure 9a. In the real scenario, ARPA marks the overtaken ship as dangerous, and the victim appears out of a safe distance. In the attacked scenario, PPI shows the overtaken ship performed a maneuver that led it to get out of the overtaking situation. Again, the malware creates the conditions to lure the victim as desired.

VI. Detection

As previously mentioned, the performance of radar equipment must comply with standards and regulations established by IMO. Moreover, the operation strictly follows the manufacturer specifications, e.g., resolution or speed of antennas, and depends on onboard configurations, e.g., SIC/SAC or IP addresses, that do not change over time. As a result, a list of rules that constrain standards and regulations, manufacturer specifications, and onboard configurations can determine the expected behavior of a radar system.

For this reason, we design the detection solution as a policy enforcement system where policies define the conditions

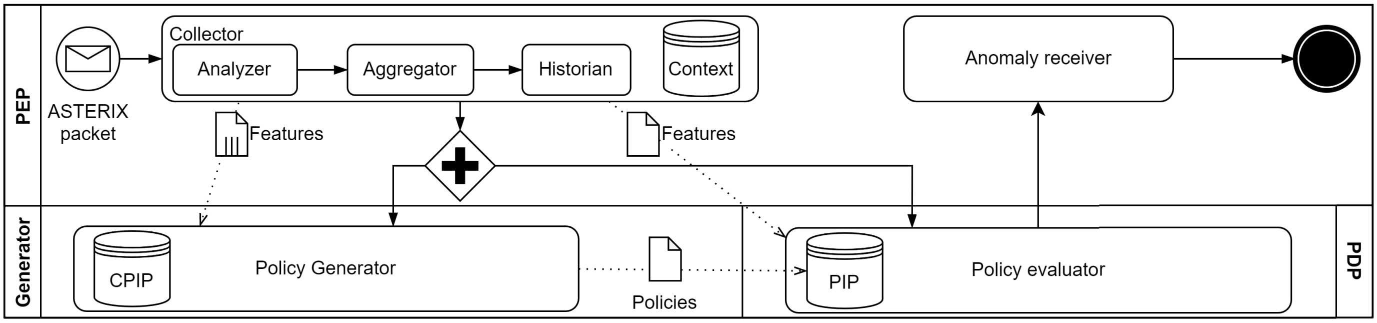


Fig. 11. The workflow of the detection system.

under which a radar system is operating as expected. The above policies can be expressed on values, their calculated aggregations, e.g., mean or variance, or frequency distribution obtained from the information carried by ASTERIX packets. To keep the solution as much general as possible, it takes as input *candidate policies* (see Section VI-B). A candidate policy contains conditions that specify its eligibility for the radar system under monitoring and uses variables to refer to quantities that depend on single manufacturers or onboard configurations. Our solution automatically infers the eligibility of candidate policies and their variables' values after receiving a proper amount of ASTERIX traffic.

The benefits are twofold: (i) it can automatically tailor to every ship configuration, (ii) it can detect all the attacks that aim at violating the normal operation of a radar system since it models the expected behavior in any running configuration. Moreover, our solution operates by connecting to the bridge network and listening for the multicast traffic like the other INS equipment. It does not require onboard systems redesign, standardization, and certification.

In Figure 11, we depict the workflow of our detection solution. Next, we present each task in detail.

A. Collector

The *collector* implements packet capture and analysis as part of our system Policy Enforcement Point (PEP) functionality. It connects to the bridge network, receives the ASTERIX traffic via multicast or broadcast, and has preconfigured the unique IDs of antennas admitted to transmitting data to the PPI. The collector relies on three components: the *analyzer*, the *aggregator*, and the *historian*.

The *analyzer* parses each packet and returns the unique ID of the sender antenna and values from the CAT-240 header (see Section II-C), e.g., center bias or covered distance.

The *aggregator* keeps a buffer of past values that the analyzer returns and calculates their aggregations, e.g., mean and variance, or frequency distribution. Aggregations occur after each revolution.

The *historian* returns a time series of aggregator values using data stored in the context database.

As a result, the collector creates the *feature* F of the received packet. F is a tuple $\langle S, D, A, A^n\rangle$ where S has the unique ID of the sender antenna, D has the data from the analyzer, A has the quantities produced by the aggregator, and A^n is the time series produced by the historian. The current F is stored in the context database that keeps the last available features for every S.

The task ends by forwarding F to the *policy evaluator*, and a set F_g consisting of F and the latest stored feature for each subject $s \neq F.S$ present in the context database to the *policy generator*.

B. Policy Generator

The *policy generator* relies on a list of candidate policies P_c stored in the CPIP database and the set of features F_g received from the *collector*. It can generate the policies to be applied by the *policy evaluator* in response to the observed input data.

Candidate policies P_c are tuples $\langle P_a, T\rangle$. Within P_c, the *activation policy* P_a is a function $F_g \rightarrow \mathbb{B} \bigcup \{undecided\}$ used to determine if a given P_c is applicable to the current system configuration. $T : F_g \rightarrow P_f$ is a *transformation function* that takes as input a set of features F_g and returns a *policy evaluator* compatible policy $P_f : F \rightarrow \mathbb{B} \bigcup \{undecided\}$. When the policy generator receives a feature set F_g from the collector, it evaluates the P_a associated with each candidate policy. Evaluation of P_a to *false* signals that P_c has been deemed incompatible with the observed data. Conversely, for a *true* verdict, T is evaluated with the same argument as P_a to generate a policy P_f that is subsequently transferred to the PIP database. While undecided results are ignored, boolean verdicts also remove the examined P_c from the CPIP.

To clarify the process of policy generation, we propose the following example.

Example 3: According to international standards [18], [37], an antenna should scan clockwise, continuously, and automatically through 360° of azimuth. To this aim, in a single antenna configuration, we want to create a candidate policy for imposing the azimuthal span to lie within three standard deviations w.r.t. its estimated mean, i.e., the 68-95-99.7 rule. Each element of A^n contains, among others, the azimuthal span sample mean μ_{az}, and the biased sample variance of the azimuthal span s_{az}. Generating an applicable policy depends on a reasonable estimation of the mean and standard deviation parameters. A possible heuristic is imposing the sample variance of the observed means and variances to be below some thresholds α and β. An activation policy P_a matching this description, characterized by the design parameters α and β, is

$$P_a = Var(\underline{\mu_{az}}) \leq \alpha \bigwedge Var(\underline{s_{az}}) \leq \beta$$

Upon the triggering of P_a, evaluation of the transformation function T will produce a policy P_f.

$$T(F) = P_f = (\overline{\mu_{az}} - 3 \cdot \sigma_{az}) \leq \mu_{az} \leq (\overline{\mu_{az}} + 3 \cdot \sigma_{az})$$

Such policy will consist of a single clause enforcing the azimuthal span value to be between $\mu - 3\sigma$ and $\mu + 3\sigma$, where μ and σ are the mean and standard deviation obtained from the samples which triggered P_a.

□

C. Policy Evaluator and Anomaly Receiver

The *policy evaluator* implements our system's Policy Decision Point (PDP) functionality. It evaluates policies in the PIP against the features F received from the collector. If any policy violation occurs, it returns an anomaly containing the description of the violated policy and the feature that triggered it.

Finally, the *anomaly receiver* implements the functionality of PEP that enforces PDP decisions. In particular, it collects anomalies and executes an action accordingly. For instance, it might generate alerts targeted at the bridge alert management systems [38] or by feeding dedicated solutions as we proposed in our implementation.

VII. Experimental Evaluation

In this section, we demonstrate the practical feasibility of the attacks against a radar system and evaluate our detection system during their execution.

A. Setting

As a testbed for attacks, we leveraged our cyber range [39] that is integrated with the Shil [40] infrastructure. It emulates a realistic ship navigation network, device sensors, and equipment as detailed in II-A. In particular, it hosts our extension of the Bridge Command (BC) [41] ship and radar simulator that implements an add-on for transmitting radar data using the ASTERIX CAT-240 protocol. A version of such testbed was released as open-source software [42]. Thus, we simulated the sensor devices by transmitting data using NMEA and a radar antenna with an accuracy compatible with the performance standards, i.e., a bearing resolution of 1° and a range resolution of 10.85 meters in the range scale of 12 nautical miles. We used a digital PPI produced by a leading manufacturer and widely adopted in naval and commercial ships. Finally, we connected two Debian GNU/Linux 11 virtual machines hosted by VMWare ESXi 7.0U3 and configured with 1 Intel Xeon Gold 6252N at 2.3GHz, 4GB of RAM, and 30GB of storage. The first acts as a bridge workstation and runs a Proof-of-Concept (PoC) implementation of the malware. The PoC has been developed in Rust [43], amounts to 3761 lines of non-library code, and supports cross-compiling to different architectures. The version we used is a Linux executable file with a size of 1171KiB.

Lastly, the second virtual machine hosts a PoC of our detection system. We realized it using Rust for implementing core tasks, Open Policy Agent (OPA) [44] for the policy engine, and Lua [45] as the scripting language for defining transformation functions of *candidate* policies. The *anomaly receiver* runs as a secondary PPI. When it receives an anomaly, it shows an acknowledgeable alarm and highlights what sectors of the radar image are affected by the anomaly.

B. Results

We generated on BC 25 instances of three scenarios that set the environment for executing the attacks detailed in Section V. Assuming that $\mathcal{U}$ is the uniform random distribution, each instance features a number $\mathcal{U}_{\{2,8\}}$ of 350m ships.[2] We placed them at a distance of $\mathcal{U}_{[3.5,5]}$ nautical miles and $\pm U_{[10,80]}$ degrees w.r.t. the bow of the victim and moved them at a random speed of $\mathcal{U}_{[2,12]}$ knots. For the Ghost Ship attack, we also added the ghost ship that moved at a speed of 10 knots and with the trajectory depicted in Figure 8. For the Ship Trajectory Hijack attack, we added a ship with a speed of $\mathcal{U}_{[2.5,5.0]}$ knots to create an overtaking situation with the victim. The victim ship moved at a speed of 10 knots, and all the vessels kept their course and speed constant.

The radar under test received ASTERIX data from BC and tracked the surrounding vessels using ARPA with the TCPA default alarm (see Section II-B) at 15 minutes.

Each experiment lasted 60 seconds for the DoS, 120 seconds for the Ghost Ship, and 600 seconds for the Ship Trajectory Hijack. At the same time, we run our detection solution configured with six policies. We divided them into two groups, namely *categorical* or *statistical*.

A categorical policy detects if a given field assumes a specific value with a probability greater than 0.99. After it activates, the generated policy enforces the above value when it exists. We configured categorical policies for the $center\_bias$ (P_1) and n_cells (P_2) fields.

A statistical policy verifies if it can construct an estimator for a field value. After it activates, the generated policy tests if the given field is consistent with the null hypothesis on the constructed estimator. We configured statistical policies for (i) the azimuthal span (see Example 3), i.e., enforcing the rotation speed to be constant in between packets (P_3), (ii) the monotonicity of the $message\_id$ field (P_4), (iii) the number of entries belonging to each aggregation, i.e., enforcing a constant rotation speed within a revolution (P_5), and (iv) the number of aggregation in the historian within a fixed time period, i.e., enforcing a constant rotation speed across revolutions (P_6).

For each experiment, we recorded performance figures (i.e., CPU and RAM usage and statistics about ASTERIX traffic) of the malware and the detection system (see Table I and II).

We used ARPA to evaluate the outcome of the attacks.

We deemed DoS attacks successful if the PPI corruption caused at least one of the following two impacts: (i) it lost tracking of a target, or (ii) returned nonphysical data about targets (e.g., a speed $\geq 100[m/s]$, or an acceleration at a rate $\geq 10[m/s^2]$). Results showed that DoS attacks had a success rate of 100%.

For PPI poisoning attacks, we considered both the accuracy of the malware w.r.t. the trajectory to reproduce and the

[2] we used 0.15 as our $sm_{\%}$ for $find$.

TABLE I

Attack Performances

Attack	ASTERIX Packets (MiB)					CPU (%)		RAM (B)	
	Legitimate		Attacker		$\frac{\sum A}{\sum(L+A)}$				
	μ	σ	μ	σ		μ	σ	μ	σ
V-A	32.90	8.01	0.04	0.0004	0.012%	3.660	1.595	3261	147.8
V-B1	35.37	1.66	0.05	0.009	0.144%	3.934	1.497	3270	96.8
V-B2	104.99	3.26	1.22	0.25	1.148%	4.226	1.575	3323	131.3

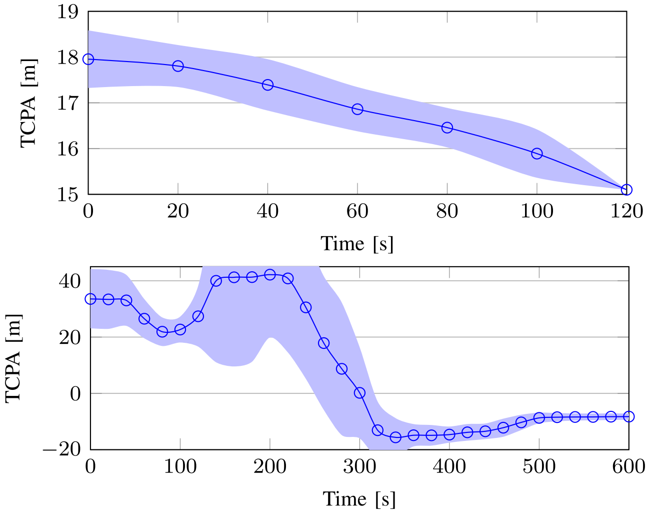


Fig. 12. Evolution of the TCPA during attacks V-B1 (left) and V-B2 (right).

TABLE II

Detection Performances

Attack	CPU (%)		RAM (KiB)	
	μ	σ	μ	σ
V-A	8.830	2.605	181.00	1.350
V-B1	10.304	4.645	181.51	1.187
V-B2	17.681	2.937	584.87	3.540

success conditions. We measured for each TTM the absolute error between the desired courses and speeds and the ones emitted by ARPA to estimate the accuracy. In 29 out of 50 (58%) cases, the course did not deviate by more than 1°, with every trajectory within 10°. In 40 out of 50 (80%) cases, the speed did not deviate by more than 0.1 knots, with every speed within 0.5 knots. To assess the success of each experiment, we considered the TCPA for the ghost and hijacked ships. In Figure 12, we present the evolution of the TCPA value during all the experiments by highlighting the complete range of the distribution and the average trend. Ghost Ship attacks required as a successful result the TCPA of the ghost ship to decrease to the collision alert threshold. On the contrary, Trajectory Hijacking attacks required the TCPA of the overtaken ship to grow up to indicate an increasing trend, i.e., a negative value. Results showed that the TCPA always complied with the expected trend leading the two attacks to a success rate of 100%.

Finally, we considered the accuracy of our detection system. In Table III, we summarize the total packets for each attack by identifying them as legitimate or malicious and how our detection system classified them in terms of true or false positives. In Table IV, we outline each attack's policies triggered during the experiments by considering the percentage of malicious packets they matched.

C. Discussion

Based on the requirements described in Section I and the techniques introduced in Section III, we put forward the following considerations concerning our malware and the feasibility of the attacks.

First, the malware can obtain the information it needs to reconstruct the state of the attacked vessel and keep it updated only by overhearing NMEA traffic (see Section IV-A). This capability allows it to determine the best time and data to execute the attack independently, without communicating outside the INS, as required by I.

The experiments conducted in our cyber range with multi-ship scenarios and the commercial PPI prove their feasibility. No special modifications need to be made to the targeted radar system and ship, confirming that the malware respects I and allows for a wide range of applicability. In addition, we show that the malware can exploit features of the ASTERIX protocol to its advantage. For DoS attacks, ASTERIX provides an amplification factor that allows the malware to operate with a significantly smaller network footprint, i.e., by sending fewer packets containing widened and lengthened cells, sharing the area of the original feed in 0.012% of its bandwidth.

We prove that legitimate modifications to the ASTERIX header for hijacking attacks can force our commercial radar display into non-compliant behavior, enabling our malware with the delete capability (see Section III-B). The above

TABLE III

DETECTION RESULTS

Attack	Packets		Detection			
	Legit	Attack	True positive		False positive	
V-A	816676	29012	29012	(100.0%)	967	(0.114%)
V-B1	877957	1266	1256	(99.21%)	900	(0.102%)
V-B2	2606273	30266	30266	(100.0%)	2340	(0.088%)

TABLE IV

DETECTION RATE OF MALICIOUS PACKETS FOR EACH POLICY

Attack	P_1 (%)	P_2 (%)	P_3 (%)	P_4 (%)	P_5 (%)	P_6 (%)
V-A	0.000	0.000	100.0	0.103	0.000	0.000
V-B1	0.000	0.000	0.000	0.000	99.21	0.000
V-B2	100.0	100.0	20.43	0.135	100.0	0.000

results should raise vendor awareness that verifying their systems properly handle all the protocol features is a strong cybersecurity requirement.

I refers to the malware's performance and the attacks' realism. In terms of performance, Table I shows that the malware never utilized more than 4KiB of RAM, and the maximum increase in CPU load was only 8.95%.[3] Additionally, the amount of malicious traffic generated by the malware never surpasses 1.148% of the legitimate traffic. It should be emphasized that all tests were performed with a 350m ship, i.e., a radar target of size comparable to some of the largest oceanic vessels. Combining this size with our chosen $sm_{\%}$ in Algorithm 1, the malware operated on a surface area corresponding to a 400m target. This choice allowed us to test it against the worst possible scenario in which a large area has to be altered. For this reason, the results represent an upper bound on the amount of computing and network resources required.

Overall, we demonstrate that the malware is lightweight, easily cross-compiled, and does not require significant resources. As a result, an attacker could install it and execute the attacks on various INS configurations, including both legacy and embedded systems.

Regarding the realism of the attacks, the results demonstrate that the attacks can accurately simulate a vessel's behavior on a predetermined trajectory. This feature coupled with the injection of AIS traffic to cheat the cross-checking with INS equipment shows high deception capability against maritime operators and the potential to cause catastrophic impacts.

The above results and performances allow the malware to meet the I requirement.

Analyzing the results of the detection system, we show that policies enforcing the performance standards for an antenna (P_3, P_5, P_6), ASTERIX protocol specifications (P_4), and the expected behavior inferred from the shipboard configurations (P_1, P_2) enabled the detection of all the attacks with high accuracy. The resulting performance, depicted in Table II highlighted that our system requires minimal resources while operating on the live radar feed. Lastly, it is worth noting that the system processes only packet headers and can ensure similar performance figures on other antennas, even with higher resolutions.

VIII. RELATED WORK

Radar systems are essential for maintaining navigation, transportation, and airspace safety and security. Therefore, various literature concentrates on attacks aimed at these systems, but only a limited amount of it focuses on primary maritime radars (see Section II-B).

In [46], the authors describe a taxonomy of possible attacks targeted at marine radar systems and provide a prototypical implementation for each identified class. Our attacks fit into their taxonomy as they are of the *denial of service* and *object manipulation* types. Furthermore, they can be classified as *sophisticated context-aware radar manipulation*. The way they carry out the attacks does not embody I and fails to meet I because their approach relies on a man-in-the-middle (MITM) configuration [47] and uses a proprietary protocol. Although the attacks can add or remove objects, which partially satisfies the requirement I, they do not account for realistic updates as we do for the PPI poisoning attacks. Finally, they do not provide details about the consumed resources and performance associated with their execution.

Kessler [48] reported that in late 2017 a cyber-consulting company successfully attacked a ship's radar. After attacking the INS network from the Internet, they gained access to the radar workstation and altered the display by deleting targets, thus blinding the ship. This work shows the feasibility of compromising radar systems, but execution details were not provided, making comparing our attacks and malware requirements impossible.

Like in our threat model, Hareide et al. [49] consider INSs as isolated systems. They achieve a successful attack by using a USB key to inject their malware into the Windows workstation running the electronic chart system. Although the attack is not aimed at the radar system, similar to us, their malware meets I as it can run without any external control and can be triggered at a specific GPS position.

From a defense perspective, while the literature has already presented some anomaly detection solutions concerning acquisitions by secondary radars [50], [51], to the best of our knowledge no work addresses detection at the polar video level of primary radars.

[3] 68-95-99.7 rule applied on Table I.

Finally, an update to the protocols and devices might be considered, as suggested in [25] and [52]. However, our proposed solution is more immediately applicable than the alternative of requiring redesign, re-certification, and rectification of relevant standards and related equipment.

IX. Conclusion

In this paper, we identified configurations and standard protocols commonly used in ships and related to INSs that are vulnerable to novel attacks targeted at maritime radar systems. We demonstrated how a suitably equipped attacker could inject targeted malware by leveraging the specific technological environment to execute the attacks autonomously.

Radar is an essential aid to ensure safe navigation, and these attacks' consequences are significant. We showed that they could lead to a high-impact disruption of normal operativity up to stealthy alterations causing awareness mismatches between the victim and other ships nearby and increasing the potential for hazardous situations.

We also developed a detection system able to recognize such attacks with high accuracy. The distinguishing features of our proposal are (i) the self-adaptation to each onboard configuration, (ii) the modeling of regulatory and expected behavior to identify known and unknown attacks, (iii) the possibility of running it without altering onboard systems, and (iv) the minimal resource footprint.

Future directions include proposing training activities on our cyber range to improve maritime operators' awareness in response to these new types of attacks.

Acknowledgment

This work was carried out while Giacomo Longo was enrolled in the Italian National Doctorate on Artificial Intelligence run by the Sapienza University of Rome in collaboration with the University of Genoa.

References

[1] *SOLAS 2020 Consolidated Edition*, IMO-Publication, Int. Maritime Org., London, U.K., 2020.

[2] *Strategy for the Development and Implementation of E-Navigation*, document MSC 85/26/Add.1 ANNEX 20, International Maritime Organization, 2008. [Online]. Available: https://u.garr.it/DSA6f

[3] *E-Navigation Strategy Implementation Plan—Update 1*, document MSC.1/Circ.1595, International Maritime Organization, 2018. [Online]. Available: https://u.garr.it/OqShS

[4] *Adoption of the Revised Performance Standards for Integrated Navigation Systems (INS)*, document RESOLUTION MSC.252(83), International Maritime Organization, 2007. [Online]. Available: https://u.garr.it/qxdve

[5] J. Wu, J. Thorne-Large, and P. Zhang, "Safety first: The risk of over-reliance on technology in navigation," *J. Transp. Saf. Secur.*, vol. 14, pp. 1–28, Apr. 2021, doi: 10.1080/19439962.2021.1909681.

[6] *Performance Standards for Automatic Radar Plotting Aids (ARPAs)*, document RESOLUTION A.823(19), International Maritime Organization, 1995.

[7] *61162-1 Maritime Navigation and Radiocommunication Equipment and Systems—Digital Interfaces—Part 1: Single Talker and Multiple Listeners*, Int. Electrotech. Commission, Geneva, Switzerland, 2016.

[8] *Specification for Surveillance Data Exchange—ASTERIX Category 240: Radar Video Transmission*, 1st ed., document EUROCONTROL-SPEC-0149-240, 2015. [Online]. Available: https://www.eurocontrol.int/publication/cat240-eurocontrol-specification-surveillance-data-exchange-asterix

[9] P. H. Meland, K. Bernsmed, E. Wille, Ø. J. Rødseth, and D. A. Nesheim, "A retrospective analysis of maritime cyber security incidents," *TransNav, Int. J. Mar. Navigat. Saf. Sea Transp.*, vol. 15, no. 3, pp. 519–530, 2021, doi: 10.12716/1001.15.03.04.

[10] M. S. Lund, J. E. Gulland, O. S. Hareide, Ø. Jøsok, and K. O. C. Weum, "Integrity of integrated navigation systems," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, May 2018, pp. 1–5, doi: 10.1109/CNS.2018.8433151.

[11] M. Kristic, S. Žułkin, D. Brčic, and M. Car, "Overreliance on ECDIS technology: A challenge for safe navigation," *TransNav, Int. J. Mar. Navigat. Saf. Sea Transp.*, vol. 15, no. 2, pp. 277–287, 2021, doi: 10.12716/1001.15.02.02.

[12] B. Cain, D. S. E. Deering, B. Fenner, I. Kouvelas, and A. Thyagarajan, *Internet Group Management Protocol, Version 3*, document RFC 3376, Oct. 2002. [Online]. Available: https://rfc-editor.org/rfc/rfc3376.txt

[13] *Technical Characteristics for an Automatic Identification System Using Time Division Multiple Access in the VHF Maritime Mobile Frequency Band*, Standard M.1371, Rev. 5, International Telecommunication Union.

[14] *Adoption of the Revised Performance Standards for Radar Equipment*, document RESOLUTION MSC.192(79), International Maritime Organization, 2004. [Online]. Available: https://u.garr.it/zrLcN

[15] *Radio Regulations*, Int. Telecommun. Union, Geneva, Switzerland, 2020. [Online]. Available: https://www.itu.int/en/publications/ITU-R/pages/publications.aspx?parent=R-REG-RR-2020&media=electronic

[16] *Annex 10 to the Convention on International Civil Aviation—Aeronautical Telecommunications*, Surveillance and Collision Avoidance Systems, Int. Civil Aviation Org., Montreal, QC, Canada, Rev. 5, vol. 4.

[17] *Radar Navigation and Maneuvering Board Manual*, 7th ed. OceanGrafix LLC, 2001. [Online]. Available: https://msi.nga.mil/Publications/RNMB

[18] *Maritime Navigation and Radiocommunication Equipment and Systems—Shipborne Radar—Performance Requirements, Methods of Testing and Required Test Results*, Standard IEC 62388, International Electrotechnical Commission, Rev. 2013.

[19] A. Dabrowski, S. Busch, and R. Stelzer, "A digital interface for imagery and control of a Navico/Lowrance broadband radar," in *Robotic Sailing*. Berlin, Germany: Springer, 2011, pp. 169–181, doi: 10.1007/978-3-642-22836-0_12.

[20] *Radar PI OpenCPN Plugin*. Accessed: Nov. 5, 2021. [Online]. Available: https://github.com/opencpn-radar-pi/radar_pi

[21] *Specification for Surveillance Data Exchange—Part 1: All Purpose Structured EUROCONTROL Surveillance Information Exchange (ASTERIX)*, 3rd ed., document EUROCONTROL-SPEC-0149, 2020. [Online]. Available: https://www.eurocontrol.int/publication/eurocontrol-specification-surveillance-data-exchange-part-i

[22] *Specification for Surveillance Data Exchange—ASTERIX Category 048: Monoradar Target Reports*, 1st ed., document EUROCONTROL-SPEC-0149-4, 2022. [Online]. Available: https://u.garr.it/MfjiK

[23] *Specification for Surveillance Data Exchange—ASTERIX Category 062: SDPS Track Reports*, 1st ed., document EUROCONTROL-SPEC-0149-9, 2023. [Online]. Available: https://u.garr.it/Thbb6

[24] D. G. Johnson and M. R. Warren, "Using ASTERIX CAT-240 for radar video distribution—practical considerations from deployed applications," in *Proc. 9th Int. Radar Symp.*, Dec. 2013. [Online]. Available: https://u.garr.it/BQNV1

[25] M. Jančík, D. Johannes, and P. Jonáš, "Security enhancements of the surveillance data exchange protocol 'ASTERIX,'" *DEStech Trans. Comput. Sci. Eng.*, 2019, doi: 10.12783/dtcse/cscbd2019/30009.

[26] T. de Riberolles, J. Song, Y. Zou, G. Silvestre, and N. Larrieu, "Characterizing radar network traffic: A first step towards spoofing attack detection," in *Proc. IEEE Aerosp. Conf.*, Mar. 2020, pp. 1–8, doi: 10.1109/AERO47225.2020.9172292.

[27] *Convention on the International Regulations for Preventing Collisions at Sea*, Int. Maritime Org., London, U.K., 1972. [Online]. Available: https://u.garr.it/QY1HZ

[28] K. Tam and K. Jones, "MaCRA: A model-based framework for maritime cyber-risk assessment," *WMU J. Maritime Affairs*, vol. 18, no. 1, pp. 129–163, Mar. 2019, doi: 10.1007/s13437-019-00162-2.

[29] *MITRE ATT&CK—Supply Chain Compromise*, Mitre Corporation, McLean, VA, USA, 2023. [Online]. Available: https://attack.mitre.org/techniques/T1195/

[30] "The guidelines on cyber security onboard ships," Version 4, BIMCO; CLIA; ICS; Intercargo; Intermanager; Intertanko; IUMI; OCIMF; World Shipping Council, Bagsværd, Denmark, Tech. Rep., 2021. [Online]. Available: https://u.garr.it/RODcs

[31] B. Svilicic, D. Brčić, S. Žuškin, and D. Kalebić, "Raising awareness on cyber security of ECDIS," *TransNav, Int. J. Mar. Navigat. Saf. Sea Transp.*, vol. 13, no. 1, pp. 231–236, 2019, doi: 10.12716/1001.13.01.24.

[32] M. S. Lund, O. S. Hareide, and O. Jøsok. (2018). *An Attack on an Integrated Navigation System*. [Online]. Available: https://brage.bibsys.no/xmlui/handle/11250/2568320

[33] Y. Dyryavyy, "Preparing for cyber battleships—Electronic chart display and information system security," NCC Group, Manchester, U.K., Tech. Rep., 2014. [Online]. Available: https://u.garr.it/ctabM

[34] The MITRE Corporation. *CAPEC-542: Targeted Malware*. Accessed: Feb. 14, 2022. [Online]. Available: https://capec.mitre.org/data/definitions/542.html

[35] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," in *Leading Issues in Information Warfare and Security Research*, vol. 1. Bethesda, MD, USA: Lockheed Martin Corporation, 2011. [Online]. Available: https://u.garr.it/SD5l3

[36] T. Vincenty, "Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations," *Surv. Rev.*, vol. 23, no. 176, pp. 88–93, Apr. 1975, doi: 10.1179/sre.1975.23.176.88.

[37] *Performance Standards for Radar Equipment*, document RESOLUTION A.477(XII), International Maritime Organization, 1981. [Online]. Available: https://u.garr.it/G6iJP

[38] *Adoption of Performance Standards for Bridge Alert Management*, document MSC.302(87), International Maritime Organization, 2010. [Online]. Available: https://u.garr.it/Knbc7

[39] E. Russo, G. Costa, and A. Armando, "Building next generation cyber ranges with CRACK," *Comput. Secur.*, vol. 95, Aug. 2020, Art. no. 101837, doi: 10.1016/j.cose.2020.101837.

[40] F. D'Agostino, D. Kaza, G.-P. Schiapparelli, and F. Silvestro, "The ShIL project: A new laboratory infrastructure for co-simulation of multi-domain marine applications," in *Proc. AEIT Int. Annu. Conf. (AEIT)*, Sep. 2020, pp. 1–6, doi: 10.23919/aeit50178.2020.9241110.

[41] J. Packer. *Bridge Command*. Accessed: Jan. 29, 2022. [Online]. Available: https://www.bridgecommand.co.uk/

[42] G. Longo, A. Orlich, S. Musante, A. Merlo, and E. Russo, "MaCySTe: A virtual testbed for maritime cybersecurity," *SoftwareX*, 2023, Art. no. 101426.

[43] N. D. Matsakis and F. S. Klock, "The rust language," *ACM SIGAda Ada Lett.*, vol. 34, no. 3, pp. 103–104, Nov. 2014, doi: 10.1145/2692956.2663188.

[44] Cloud Native Computing Foundation. *Open Policy Agent*. Accessed: Jan. 29, 2022. [Online]. Available: https://www.openpolicyagent.org/

[45] R. Ierusalimschy, L. H. de Figueiredo, and W. C. Filho, "Lua—An extensible extension language," *Softw., Pract. Exper.*, vol. 26, no. 6, pp. 635–652, Jun. 1996, doi: 10.1002/(sici)1097-024x(199606)26:6<635::aid-spe26>3.0.co;2-p.

[46] K. Wolsing et al., "Network attacks against marine radar systems: A taxonomy, simulation environment, and dataset," in *Proc. IEEE 47th Conf. Local Comput. Netw. (LCN)*, Sep. 2022, pp. 114–122.

[47] Mitre Corporation. *MITRE ATT&CK—Man in the Middle*. Accessed: Feb. 14, 2022. [Online]. Available: https://collaborate.mitre.org/attackics/index.php/Technique/T0830

[48] G. C. Kessler, "Cybersecurity in the maritime domain," *USCG Proc. Mar. Saf. Secur. Council*, vol. 76, no. 1, p. 34, 2019. [Online]. Available: https://commons.erau.edu/publication/1318

[49] O. S. Hareide, Ø. Jøsok, M. S. Lund, R. Ostnes, and K. Helkala, "Enhancing navigator competence by demonstrating maritime cyber security," *J. Navigat.*, vol. 71, no. 5, pp. 1025–1039, Sep. 2018.

[50] S. Cohen, E. Levy, A. Shaked, T. Cohen, Y. Elovici, and A. Shabtai, "RadArnomaly: Protecting radar systems from data manipulation attacks," *Sensors*, vol. 22, no. 11, p. 4259, Jun. 2022, doi: 10.3390/s22114259.

[51] T. de Riberolles, Y. Zou, G. Silvestre, E. Lochin, and J. Song, "Anomaly detection for ICS based on deep learning: A use case for aeronautical radar data," *Ann. Telecommun.*, vol. 77, nos. 11–12, pp. 749–761, Jan. 2022, doi: 10.1007/s12243-021-00902-7.

[52] E. E. Casanovas, T. E. Buchaillot, and F. Baigorria, "Vulnerability of radar protocol and proposed mitigation," in *Proc. ITU Kaleidoscope, Trust Inf. Soc. (K)*, Dec. 2015, pp. 1–6, doi: 10.1109/Kaleidoscope.2015.7383631.

Giacomo Longo received the B.S. and M.S. degrees in computer engineering from the University of Genoa in 2018 and 2021, respectively. He is currently pursuing the Italian National Ph.D. degree in AI and cybersecurity. His research interests include systems security analysis and virtualization.

Enrico Russo received the M.Sc. degree in computer science and the Ph.D. degree in computer science and systems engineering from the University of Genoa in 2001 and 2021, respectively. He joined as an Assistant Professor with DIBRIS, University of Genova, in 2021. His research interests include cyber range systems and maritime cybersecurity.

Alessandro Armando received the M.Eng. and Ph.D. degrees in computer engineering from the University of Genova. His appointments include a position as a research fellow with The University of Edinburgh and one with INRIA-Lorraine, France. He is currently a Professor with the University of Genova, where he teaches computer security and founded and coordinated the master's in cybersecurity and data protection. In 2011, he founded (and led until 2016) the Security and Trust Research Unit, Bruno Kessler Foundation, Trento. He is also the Vice Director of the CINI National Cybersecurity Laboratory. He has been the coordinator and the team leader in several national and EU research projects. He contributed to discovering an authentication flaw in the SAML 2.0 web-browser SSO profile and a severe man-in-the-middle attack on the SAML-based SSO for Google Apps.

Alessio Merlo (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Genoa in 2010. He is currently a Full Professor in computer engineering with the Centre for Higher Defence Studies (CASD), Rome, Italy. He has published more than 120 scientific papers in international conferences and journals. His research interests include mobile security, where he contributed to discovering several high-risk vulnerabilities both in applications and the android OS and system security.

Open Access funding provided by 'Università degli Studi di Genova' within the CRUI CARE Agreement