

Attacking (and defending) the Maritime Radar System

Giacomo Longo, Enrico Russo, Alessandro Armando, and Alessio Merlo

Abstract—Operation of radar equipment is one of the key facilities used by navigators to gather situational awareness about their surroundings. With an ever increasing need for always-running logistics and tighter shipping schedules, operators are relying more and more on computerized instruments and their indications. As a result, modern ships have become a complex cyber-physical system in which sensors and computers constantly communicate and coordinate.

In this work, we discuss novel threats related to the *radar system*, which is one of the most security-sensitive component on a ship. 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 even safety itself. Then, we present a detection system aimed at highlighting 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 of a simulated environment. The experimental results clearly indicate that the attacks are feasible, rather 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 is increasingly relying on both Information Technology (IT) and Operational Technology (OT). The advantages brought by their adoption cannot be overestimated as OT, through the automation of onboard operations associated with the mechanical and electrical subsystems, enables a reduction of costs as well as the execution of risky tasks by the crew and IT, and more generally Information and Communication Technologies (ICT), provide invaluable support to navigation planning, control and monitoring. As a matter of fact, commercial ships undertaking international voyages are subject to multilateral treaties mandating the installation of a variety of electronic devices [1]. Such provisions, combined with initiatives promoted by the International Maritime Organization (IMO), e.g. e-navigation [2], [3], have led to a significant onboard system digitization.

The Integrated Navigation System (INS) lies at the core of this digitization. By gathering information and integrating functions from a variety of electronic devices (e.g. the radar), the INS helps the operator to plan, monitor, and control the navigation and contributes to improving the overall situational awareness [4]. During navigation the radar plays a key role in the formation of 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 the interaction among all devices, the latter supports video data transmission between the radar antennas and the displays.

While these technologies contribute to improving the safety and effectiveness of navigation, they also expose ships to the Cybersecurity threat. Meland et al. [9] presented an overview of 46 maritime cyber security incidents occurred in the last decade (2010-2020). While the overall number of cyber attacks is relatively small when compared to other sectors, unfortunately the impact of cyber attacks in the maritime sector can be very high. The incident of the *Ever Given*¹ in the Suez Canal, although not due to a cyber attack, is a dire reminder of the magnitude of the disruptions that can occur in the maritime sector.

An attacker can have a variety of objectives ranging from the “mere” disruption of the operations aiming to inflict hefty economic losses or the payment of a ransom to the deliberate attempt to cause a collision. Since the crews make decisions by cross-checking between multiple systems and what they perceive when they look outside, an attack is likely to succeed when it takes this matter into account.

A key problem is that both the NMEA and ASTERIX protocols assume that the navigation network as well as the interconnected subsystems are trusted and no provision for cryptographic protection is therefore provided. The aforementioned cybersecurity incidents show that this trust assumption is no longer tenable and this security weakness of NMEA associated with the INS is widely recognized [10]. Even worse, due to the average life span of modern ships (up to 40 years) and the fact that retrofitting the INS is expensive and time-consuming, these weaknesses are here to stay. (It is however reasonable to expect that intrusion detection tools capable to identify unexpected network traffic and/or resource consumption deviations will eventually find their way into the INS.)

Yet, launching a successful cyber attack against a ship is not easy. INSs are typically offline and penetrating them through lateral movements from other networks and controlling an attack from the Internet may not be an option. Additionally,

G. Longo, E. Russo, A. Armando, and A. Merlo are with University of Genoa

¹https://en.wikipedia.org/wiki/2021_Suez_Canal_obstruction

both the individual components and the configuration of the INS may vary from ship to ship. For these reasons we argue that a cyber attack with reasonable chances of success requires the development of a malware that:

- exhibits a high degree of autonomy (e.g. the ability to pursue its objectives without human support or guidance),
- does not rely on the knowledge about the individual components and the actual configuration of the INS,
- is stealthy, i.e. its behavior is hard to detect by anomaly detection tools available at the host and/or network level (this implies that the malicious activity must be executed by requiring a moderate use of the CPU, of the memory resources and of network bandwidth to avoid behavioral fingerprinting) and—for the most sophisticated types of attacks—its effects are difficult to detect by the crew (this implies that the information shown on the displays is consistent with the other sources of information contributing to the situational awareness.)

Even if ships are offline during navigation, the injection of the malware into their INS can be carried out during management or upgrade of the INS.

Previous works partially consider the above constraints. In general, they do not discuss the security of the ASTERIX protocols along with the internals about attacks against radar systems.

Hareide et al. [11] 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. The malware can run without any external control, and they programmed it to trigger at a specific GPS place. It leverages a man-in-the-middle (MITM) attack [12] to inject false GPS values and force the chart system to show a faulty position.

Casanovas et al. [13] analyzed an equivalent standard protocol for surveillance information exchange among different aerial traffic control centers, namely ASTERIX CAT-032. They show how the lack of security mechanisms in such a protocol can lead to a MITM attack enabling the deletion and insertion of aircrafts and the update of their track, thus causing a misperception to air traffic control operators.

Kessler [14] 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, altered the display by deleting targets, and thus blinding the ship.

In this work, we introduce a novel class of attacks against maritime radar systems and we propose a method to detect them.

First, the attacks can be performed by malware acting on its own, without command-and-control servers, and able to determine when the ship's state is suitable to execute them. The malware can be easily adapted to each INS configuration. Moreover, the malware only exploit security weaknesses and specific features of ASTERIX and NMEA protocols and the configuration of the INS network. They do not require access to the radar workstation. The attacks can either corrupt and make the radar display unavailable or be sophisticated and stealthy up to modifying explicit details of the radar

image in real-time and with extreme realism. After modifying radar images, they generate consistent data for the other INS equipment. Finally, we show the malware performs all the operations requiring very little CPU and memory resources and a limited network bandwidth.

The contributions of our work are as follows.

- 1) We provide a high-level yet precise reconstruction of an Integrated Navigation System and its security assumptions.
- 2) We argue that these assumptions are no longer justified in the light of emerging cyber threats and actors and impact at a successful attack.
- 3) We show that security weaknesses can be exploited in such a way to disrupt situational awareness and lead to dramatic consequences.
- 4) We argue that crafting an attack of this type requires a sophisticated and determined attacker, but given the severity of the impact (e.g., life loss, environment, or economic), the threat should not be underestimated. State actors and criminal organizations have already been shown to have the skills, resources, and determination to plan and execute attacks with this (and ever greater) level of sophistication.
- 5) 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.

This paper is structured as follows. In Section II we recall some preliminary notions. In Section III we introduce the threat model and attack techniques to hijack a radar system. 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. Finally, we conclude the paper in Section VIII.

II. BACKGROUND

In this section, we recall the notions that are relevant for correctly understanding the content of the paper.

A. Ship navigation network

On a ship, the navigation network (see Figure 1) connects sensors and bridge systems. 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 own messages to all broadcasted traffic. Similarly, any device can discover, listen and communicate with multicast flows via the standard IGMP protocol [15].

The main aim is to ease creating a system, namely an Integrated Navigation System (INS), that promotes data fusion and synergy between different equipment operating independently.

A Serial to Ethernet converter is a collection unit for sensors devices installed on a ship that forwards data to the navigation network. The main sensors devices are the Electronic Position

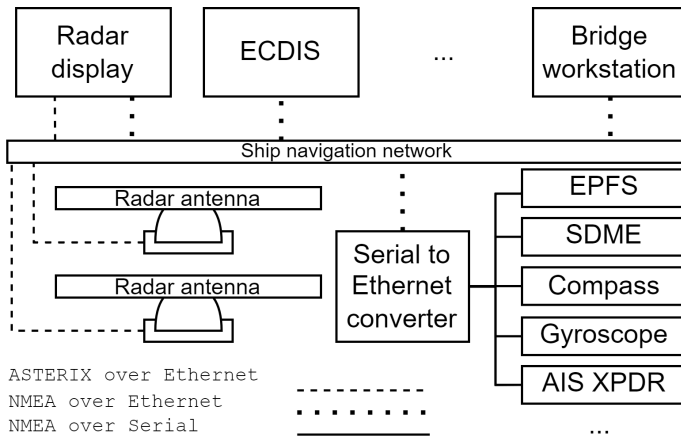


Fig. 1: Ship navigation network topology.

Fixing System (EPFS), the Speed and Distance Measurement Equipment (SDME), the Compass, the Gyroscope, and the Automatic Identification System (AIS) transponder (see Section II-C). The navigation network also hosts the two most essential navigational equipment: the radar system (see Section IV-A) and a specialized digital navigation computer, namely the Electronic Chart Display and Information System (ECDIS). Finally, one or more workstations are at the disposal of the deck personnel.

NMEA 0183 and ASTERIX are the core standards that enable the integration of all these sensors and equipment. We will briefly introduce them in the following sections.

B. NMEA 0183

The NMEA 0183 standard [7] defines an electrical and data exchange format between maritime electronics. Most of the sensor devices and systems installed on a ship communicate via NMEA [4].

Each message (or *sentence*) is comprised of a start character followed by comma-delimited fields and a simple checksum terminated by a two-byte delimiter. Of particular interest is the *talker sentence* format in which each message contains a two letter talker identifier, a three letter sentence type and a variable number of fields. An example sentence with the talker identifier follows.

```
$HETHS,33.2,A*1F
```

It represents a message emitted by the gyroscope (HE), with a sentence type related to the true heading and status (THS), and indicating a sensor heading measurement of 33.2° , sent automatically (A) and having a checksum of $1F_{16} = 31_{10}$.

From a security standpoint, this protocol has no built-in message authentication nor any confidentiality protection.

C. AIS

The Automatic Identification System (AIS) [16] is a standard system for data exchange between ships and other maritime authorities. Multiple standard message types are defined, covering a broad range of safety-enhancing functionality. For

instance, periodically broadcasted position reports indicate the current course and speed of ships, reducing the risk of collisions.

Reception and transmission are carried out over Very High Frequency (VHF) radio data links. Each AIS message is usually transported within NMEA talker sentences with types VDM and VDO, respectively for received and sent messages. Such encapsulation is often used for displaying and utilizing the received information on other INS devices. In particular, radar plotters can associate their tracks with received AIS information [17].

Since 2002, all ships engaged in international voyages above 300 gross tonnage [1] are mandated to install an AIS, with other regulations suggesting its performance requirements [18], how to perform the mandatory annual test [19] and guidelines for its correct operation [20].

The protocol presents no message authentication and - due to its broadcast nature - does not provide any mean to confidentially exchange information; albeit this issue has been investigated multiple times in the literature [21], [22], [23], the few regulations concerning the hardening of AIS are only devoted to the protection of its radio frequency band from rogue transmitters, but do not provide any solution to secure the AIS from insecure inputs [24].

D. RADAR system

RADAR system (RADAR) is a system that can detect surrounding objects using radio waves. The whole radar system relies on different devices but we can look at it as composed by 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. Antennas have their own specifications that differ between manufacturers. In particular, each specification includes the *rotation speed*, and a resolution related to the *bearing* and *range*. The rotation speed specifies the speed at which an antenna is rotated by the motor. The bearing resolution, or *angular* resolution, determines the ability of a radar system to separate targets at the same distance but at different direction. The range resolution determines the ability to resolve between two targets on the same direction, but at slightly different distances.

Echoes from an antenna can be transmitted to the PPI via Ethernet network using proprietary solutions [25], [26] or the standard protocol from ASTERIX (see Section II-E).

The PPI is a circular display representing the antenna, with the own ship 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. The top of the display may be configured 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 [27]. 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 [28, §15.6.3.2.e].

Digital PPIs also add new capabilities over traditional radar scopes. For example, the echo *trail* is used to visually understand the movements of other vessels, i.e., path and speed, by displaying a residual image at different times of an echo. Radar systems that support ARPA capabilities (see Section II-F) can automatically provide an accurate estimate of such movements.

E. ASTERIX

ASTERIX [29] is a suite of standard protocols for data exchange of radar information between systems proposed by EUROCONTROL. The ASTERIX standards identify a collection of message types, called categories or CAT. Of particular interest for this work is CAT-240, i.e., Radar Video Transmission [8], used to transfer video data from antennas to Plan Position Indicator (PPI) displays. After its specification in 2009, ASTERIX CAT-240 has been adopted by manufacturers as the de-facto network video standard [30].

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 quantized using $cell_res$ bits. Moreover, each cell starts at a distance ρ that can be calculated by leveraging their homogeneity among the distance direction as $\rho = D * (b + i) * c/2$ 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 celerity².

Referring to the resolution of antennas (see Section IV-A), the bearing resolution determines the minimum span between $start_az$ and end_az while the range resolution determines the minimum $cell_dur$.

Finally, $message_id$ is a sequence number used by the receiver to reorder packets.

From a security standpoint, as emphasized in [31], [32], the ASTERIX protocol does not implement any authentication and encryption features.

F. ARPA

The Automatic Radar Plotting Aid (ARPA) is a system integrated with radar displays that carry out the task of *radar plotting*. International law mandates ships exceeding 10000 gross tonnage [1, V§2.8] to be equipped with such a device.

Radar plotting allows a radar officer to follow a target over time, reconstructing its trajectory w.r.t. the own ship,

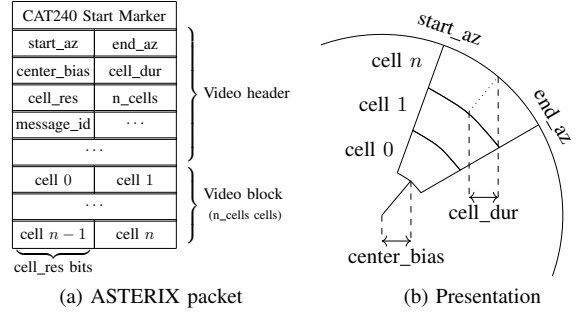


Fig. 2: Correspondence between ASTERIX and the PPI.

estimating its course, speed, 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.

A target can be acquired manually by an officer or automatically when it enters in configurable *acquisition zones*. A target becomes acquired after it persists for 5 out of 10 consecutive scans [17, §3.3.3].

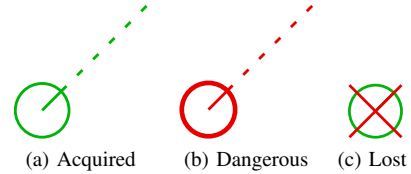


Fig. 3: ARPA target symbols on the PPI.

Once acquisition is performed, the radar system tries to follow the movement of targets inside of the image. This process, combined with the EPFS and SDME data allows the ARPA system to estimate a target's trajectory. Finally, if estimation is successful a target becomes a *tracked target* [28, Ann.G]. Targets that are being tracked appear on display with the symbol depicted in Figure 3a. Within INS, the radar system propagates information about such targets via NMEA and using the TTM (Tracked Target Message) sentences.

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. Targets that generate an alarm appear on display as dangerous and with the symbol depicted in Figure 3b.

Finally, a tracked target is judged as a lost target when no return is received for nine consecutive scans and appears on display with the symbol depicted in Figure 3c.

G. 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 [33]. These rules specify maneuvers that ships must

²299792458 m/s as in [8]

take in situations where a risk of collision occurs. A vessel may employ radar and ARPA to assess its relative position, angle of approach, and speed against near ships and determine such a risk.

This work addresses two COLREGs rules where only one of the involved vessels must maneuver: *overtaking*, and *crossing* situations.

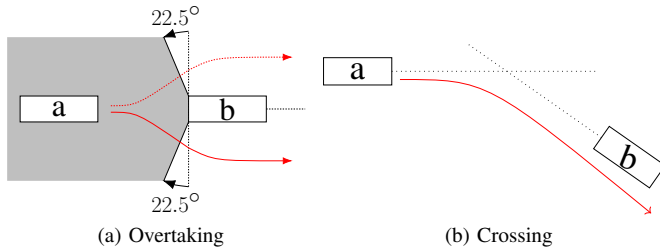


Fig. 4: Illustration of COLREGs situations.

In particular, rule 13 is about overtaking and 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”. Fig. 4a illustrates a 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 governs crossing situations and states that “when two power-driven vessels are crossing so as 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”. Fig. 4b illustrates a crossing situation. In such a situation, the vessel *a* is in a collision course with *b* and must veer to its starboard so that it does not cross ahead of *b*.

III. ATTACK TECHNIQUES

In this section, we present the threat model and the techniques to hijack a radar system.

A. Overview

In Sections II-E and II-B, we highlight that both ASTERIX and NMEA protocols do not support confidentiality and authentication between communicating components. Moreover, modern vessels are equipped with an INS (see Section II-A) where such components communicate through multicasts or broadcasts that allow anyone connected to listen to the exchanged packets. The feature of these 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 here consider a novel kind of adversary capable of executing *fine-grained* attacks. A fine-grained attack does not create malfunctions but alters the radar information

without being detected. 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 the domain it operates in and can pose a severe or catastrophic adverse effect, e.g., harm to individuals, major damage to the vessel and environment, and major financial loss.

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

B. Threat model

In this work, we focus on stealthy malware that accesses the ship 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 to significantly increase the probability of a safety-critical incident.

We assume that the ship under attack is equipped with an INS configured as detailed in Section II-A. In particular, we assume it hosts a radar system compliant with regulations, performance standards, and behaviors as described in Section II-D.

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.

Attacker’s requirements: We consider an adversary at least as a professional actor capable of gathering solid knowledge for generating or testing a novel attack. The framework for Maritime Cyber-Risk Assessment (MACRA) [34, T.1] models such an adversary as a *Tier₃* attacker. The abilities and resources of a *Tier₃* attacker allow the adversary to leverage the maintenance operations or the supply chain compromise technique [35, T1195] to install the malware. As reported in [10], multiple components of an INS are periodically updated, as well as bridge workstations present vulnerabilities that might allow the installation of malware [36], [37], [38] during a regular maintenance operation [39, p.32-36].

Once installed, the malware must operate stealthily and under the assumption that the network is isolated from the Internet. Traditional malware that drop additional malicious payloads and require a command and control server are out of scope. Instead, the attack requires a targeted malware [40] that can operate in autonomy and takes advantage of the specific technology environment.

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 any 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.

C. 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 IV): (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 consist in trying 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 5a, 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 at time t_0 the ASTERIX message (1a) for the azimuth α from the legitimate antenna. 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-E). 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.

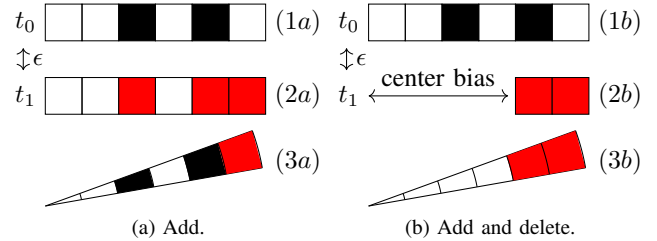


Fig. 5: Hijacking techniques.

In Figure 5b, we show an example of an attack using the *center_bias* parameter. The objective of this second attack is 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 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) that corresponds 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-B, we show that the malware can automatically infer if the PPI under attack suffers from behavior similar to our testbed and allows attackers the delete capability.

IV. ATTACK DESCRIPTION

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

A. Overview

Figure 6 shows the workflow of the stealth malware. We map it to three steps that are inspired by the cyber kill chain [41]. In the *reconnaissance* step, the malware evaluates its capabilities w.r.t. the current radar system (see Section IV), monitors the state of the ship, and determines whether an attack should start. In the *weaponization* step, the malware

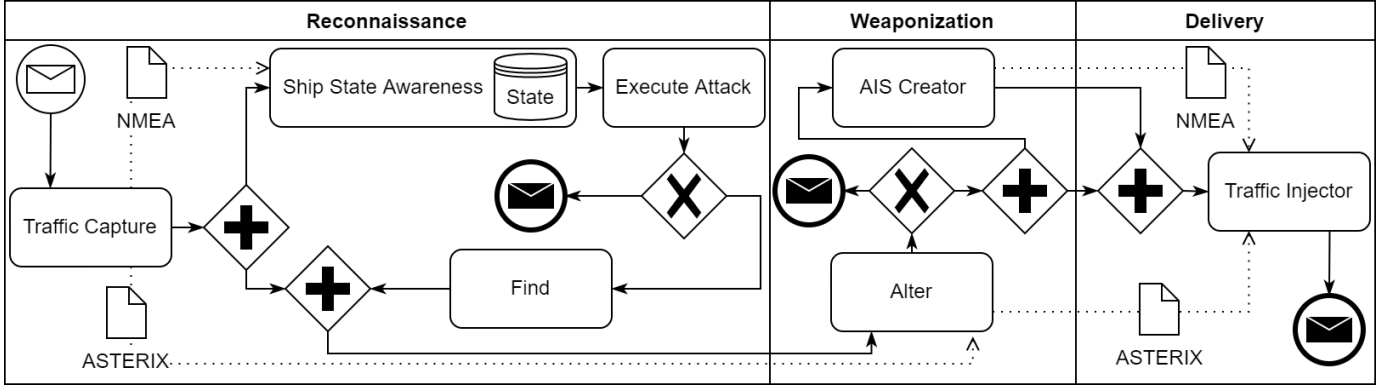


Fig. 6: The workflow of the stealth malware.

creates the ASTERIX and NMEA packets to alter the image on the PPI and updates the information on the INS devices, accordingly. Finally, in the *delivery* phase, the malware injects the weaponized packets into the bridge network. We detail below the three steps.

B. 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 the situation awareness. The aim is twofold: detecting if the radar system 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-F). 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 TTMs for the target under attack or sends TTMs 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 (like position, speed, and bearing), nearby vessels, tracked targets, and weather and environmental conditions.

The *execute attack* task leverages the values of the above state to automate the decision to start the attack. For example, the malware could trigger an attack at a specific GPS location, based on 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.

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 7. It is an annulus sector centered on point O that has the

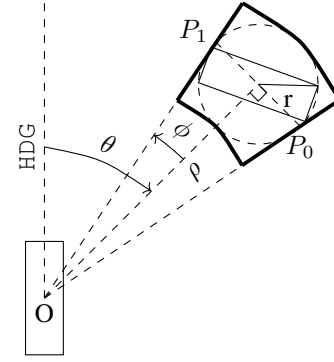


Fig. 7: An example annulus section and related quantities.

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
3:     return  $\langle 0, 360, 0, \infty \rangle$ 
4:   end if
5:    $r \leftarrow \frac{\sqrt{w^2+h^2}}{2}$ 
6:    $r^* \leftarrow r(1 + sm\%)$ 
7:    $\phi = \text{atan2}(r^*, \rho)$ 
8:    $a_{min/max} = C_{360}(\theta \pm \phi)$ 
9:    $d_{min} = \max\{0, \rho - r^*\}$ 
10:   $d_{max} = d + r^*$ 
11:  return  $\langle a_{min}, a_{max}, d_{min}, d_{max} \rangle$ 
12: end function

```

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 $\rho = 0$ (Line 2) for returning a zone delimiting the entire radar image (Line 3). 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 (Line 5). The bounding circle can also be expanded by a percentage factor $sm\%$, obtaining the final radius r^* (Line 6). In this way, the buffer zone allows compensating position inaccuracies at the expense of a less precise find zone determination. To calculate the angle span ϕ (Line 7), 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 8) 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 9) and return the tuple for the zone (Line 11).

Example 1. After having received an AIS message (see Section II-C) 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 [42], 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 from NMEA sentences originated by compasses or gyroscopes, e.g., HDT and THS sentences. Finally, w and h can be obtained from AIS, i.e., *ship static and voyage related data* messages that contain the size of the target ship. \square

C. 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 than can modify how such echoes appears 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 (see Section II-C) with data reflecting the changes occurring on the radar system. Consequently, INS equipment using AIS will display information consistent with the attack. This task aims at making the attack more difficult to detect since maritime operators can not rely on cross-checking procedures [43]. We detail below the alter function.

Alter function: The alter function allows modifying echoes of an existing ASTERIX packet Pkt . Such an operation is performed by wrapping the execution of an user-provided variadic function $f : (Pkt \times F_o \times F_a) \rightarrow Pkt \cup \emptyset$ 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. Now, we want to copy its image into a different position, thus creating a *ghost ship*. We create a function `copy_ship` that can be used with the alter function.

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

```

Algorithm 2 represents the implementation of the `copy_ship` function. 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 begins by returning empty 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 2-4). Then, it calculates the cell index distance offset i_o (Line 5) and copies the original video cell contents in a support variable (Line 6). Following, for each cell in Pkt (Line 8), it calculates the minimum ρ_{min} (Line 9) and maximum ρ_{max} (Line 10) covered distances as detailed in Section II-E. 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 11), the algorithm copies the original cell value into the offset position (Line 12). Finally, the function modifies the packet Pkt azimuthal span returning it (Line 19) or empty if no modification happens (Line 21). \square

D. Delivery

The *delivery* step starts by receiving the weaponized NMEA and ASTERIX packets. The *traffic injector* task is in charge of injecting such packets into the navigation network. As the involved protocols do not support authentication, this task

forwards them to the multicast or broadcast addresses that INS equipment and the PPI use to communicate in the navigation network. Once the above equipment consume the weaponized packets, they display the hijacked image and data.

V. RADAR HIJACKING

In this section, we will discuss two novel classes of attacks for radar hijacking leveraging the previous techniques.

A. Denial of Service attack

A Denial-Of-Service (DoS) attacks aims at rendering the radar system unusable and leaving the ship without means of safe navigation. In this attack, the adversary overlays sectors or the entire azimuthal range of the radar image by filling them with echoes. Below, we detail how adversaries can implement the different steps introduced in Section IV.

Reconnaissance: We assume that a radar must continuously operate during the 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 the 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 $\rho = 0$ as detailed in Section IV-B.

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 for a DoS. 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-E) are the amplification factors we use in the alter function of DoS attacks, namely the *DoS* function.

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
3:   Pkt.end_az  $\leftarrow$  360
4:    $n \leftarrow \frac{32}{cell\_res}$ 
5:   Pkt.n_cells  $\leftarrow$   $n$ 
6:   Pkt.cell_dur  $\leftarrow \frac{Pkt.cell\_dur \cdot Pkt.n\_cells}{n}$ 
7:   Pkt.cells  $\leftarrow [2^{cell\_res}, \dots, 2^{cell\_res}]$ 
8:    $i \leftarrow i + 1$ 
9:   if  $i = k$  then
10:     $i \leftarrow 0$ 
11:    return Pkt
12:   end if
13:   return
14: end function

```

Algorithm 3 represents the implementation of the *DoS* function. i and k are parameters used for controlling the injection rate as explained below. The idea behind this function is to update the received packets with new ones containing

echoes at maximum strength and covering the entire angle and distance span.

For covering the entire angle span, we set the *start_az* to 0 and *end_az* to 360 (Lines 2-3). For the distance span, we use the minimum number n of cells w.r.t. the constraints set by the ASTERIX protocol. This solution creates a video block that is as small as possible. To calculate n , we use the minimum number of bits in a video block, i.e., 32, and the current cell resolution (Line 4-5). As we replaced the number of cells in the original packet with n , we recalculate their *cell_dur* (Line 6). Then, we set each of the n cells at the maximum strength (Line 7).

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 original image visible between each injection.

In the function, i refers to a persistent counter which increments after each call (Line 8). Once i equals k , the modified packet is returned and i is reset to 0 (Lines 10-11). Otherwise, a null value is returned, indicating that no injection has to take place (Line 13).

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

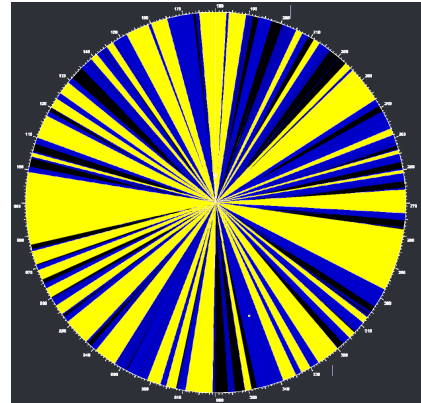


Fig. 8: Radar after a DoS attack with $k = 10$.

Delivery: Whenever the *DoS* function returns a non-null value, the *traffic injector* task transmits the modified packet to the multicast address allocated to the radar system. In Figure 8, we show how the PPI looks like after a DoS attack and using $k = 10$.

B. PPI poisoning attack

A PPI poisoning attack alters specific sections of the image shown on the PPI in real-time. The aim is to induce the crew to make wrong decisions or to fail carrying out the required actions while underway.

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 in accordance with COLREGs (see Section II-G). 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 system granting the delete capability (see Section III-C) to the attacker.

1) *Ghost ship*: A ghost ship is a fictitious target that this attack introduces into the radar image. Such a target 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.

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 9a. 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 .

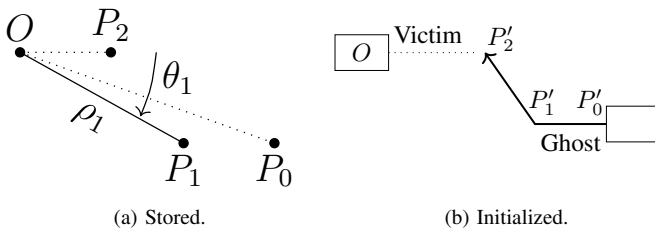


Fig. 9: An example trajectory comprised of three points.

We program our malware to use this trajectory to reproduce a COLREGs crossing condition (see Section II-G) 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 9b. It uses the victim position as the origin O and obtains their absolute coordinates (lat, lon) via the application of a geodesic formula (e.g., the one in [42]).

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.

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

$$\Delta C(t) = C(t) - COG(t) \quad (2)$$

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

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

$$a(t) = A \cdot sgn(S(t) - SOG(t)) \quad (5)$$

$$SOG(t + \Delta t) = SOG(t) + a(t) \cdot \Delta t \quad (6)$$

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-B. 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.

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 how we generate the source image for the ghost ship.

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

Delivery: The *traffic injector* injects weaponized AIS and ASTERIX packets. NMEA devices show the position of the ghost ship as real ones. PPIs display the video feed as in Figure 10b instead of the real one as in Figure 10a. We set the PPI in head-up mode, and we enable trails (see Section II-D). In both cases, the PPI displays two real targets, and in Figure 10b, 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 9b. Moreover, the radar system acquires the ghost ship as a valid target, and ARPA marks it as a dangerous one (see Section II-F).

The effects above show that the malware reproduced the conditions that lure operators to execute an evasive maneuver as desired.

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-G). The adversaries aim at modifying the trajectory of the vessel being overtaken so that no evasive maneuvers seem to be required.

The steps of the attack can be outlined as deleting the real target's echo and adding a ghost ship with the new trajectory, as described below.

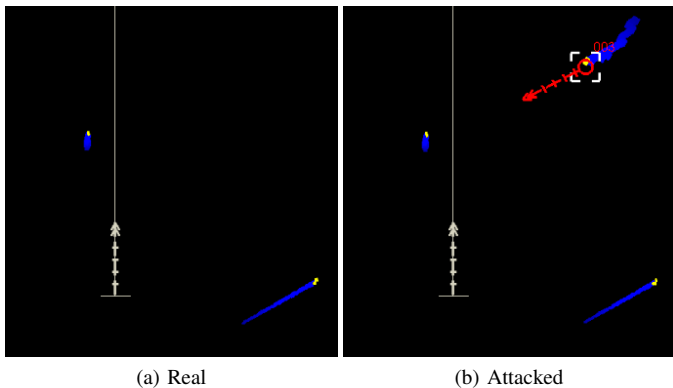


Fig. 10: Comparison of video feeds during V-B1.

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 9, 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 as the one described in Example 1. The second follows the implementation as in the ghost ship attack and using T .

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-C). The second follows the implementation as in the ghost ship attack.

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

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 [16]).

Poisoned PPIs display the video feed as in Figure 11b instead of the real one as in Figure 11a. Although the ARPA marks the overtaken ship as dangerous, the victim appears out of a safe distance in the real scenario. 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

In this section, we detail the design and the implementation of a detection system for the previously described attacks.

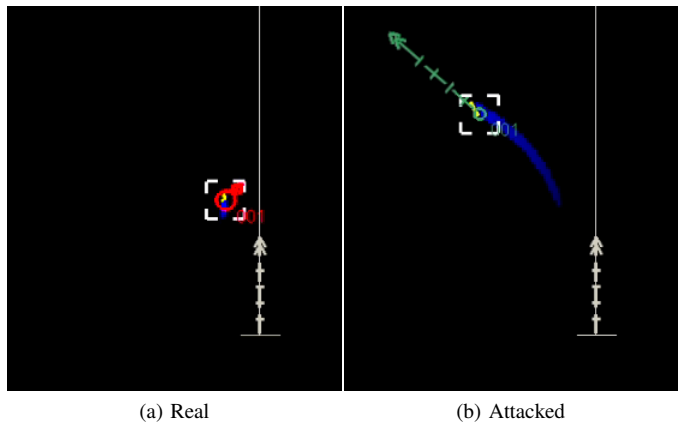


Fig. 11: Comparison of video feeds during V-B2.

A. Overview

As previously mentioned, performances 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 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-C). 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 the values of their variables after it receives a proper amount of ASTERIX traffic.

This implementation provides two main benefits: first, it can automatically tailor to every ship configuration; second, 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. For this reason, it does not require onboard systems redesign, standardization and certification.

In Figure 12, we depict the workflow of our detection solution. Next, we present each task in detail.

B. Collector

The *collector* implements packets capture and analysis as part of our system Policy Enforcement Point (PEP) functionality. It connects to the bridge network, receives the ASTERIX

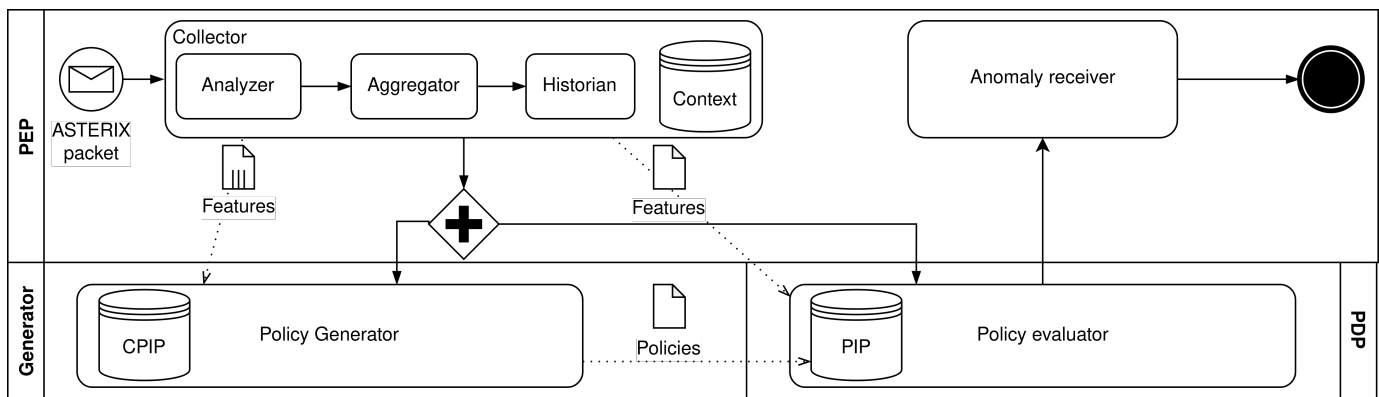


Fig. 12: The workflow of the detection system.

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 CAT-240 header (see Section II-E), 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 values of the aggregator 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*.

C. 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 $F_g \rightarrow \mathbb{B} \cup \{\text{undecided}\}$ 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 $P_f : F \rightarrow \mathbb{B} \cup \{\text{undecided}\}$. 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 result in the removal of the examined P_c from the CPIP.

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

Example 3. According to international standards [44], [28], an antenna should scan clockwise, continuous, and automatic 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

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

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 . \square

D. 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 operate by generating alerts targeted at the bridge alert management systems [45], or by feeding dedicated solutions as we proposed in our implementation.

E. Implementation

We realized our detection system using the Rust programming language [46] for implementing core tasks, Open Policy Agent (OPA) [47] for the policy engine, and Lua [48] as the scripting language for defining transformation functions of *candidate* policies.

The system’s configuration requires the list of unique IDs of transmitting antennas (i.e., the union of their SIC/SAC, IP address, and the network port used for communicating with PPI), the IP multicast address to bind, and where to forward anomalies.

The main thread starts capturing traffic received through the multicast. Then, it performs analysis on each packet to extract and calculate its features.

Values from features represent inputs for threads implementing the functionalities of the *policy generator* and *evaluator*. As detailed in Section VI-C, the *policy generator* receives a list of features related to the last packet and the last ones collected from the other antennas in the system.

An OPA instance evaluates *activation* policies from available *candidate* policies from a repository directory. Policies are expressed in Rego, i.e., the declarative language of OPA.

After an *activation* policy is admissible, the *policy generator* executes the embedded interpreter to evaluate the Lua script generating the entry for the PIP database.

The thread implementing the functionalities of the *policy evaluator* interacts with the OPA instance, evaluating each entry stored in the PIP against the features received from the collector. If the OPA instance returns an anomaly, it forwards its details to a web application that acts as our *anomaly receiver*.

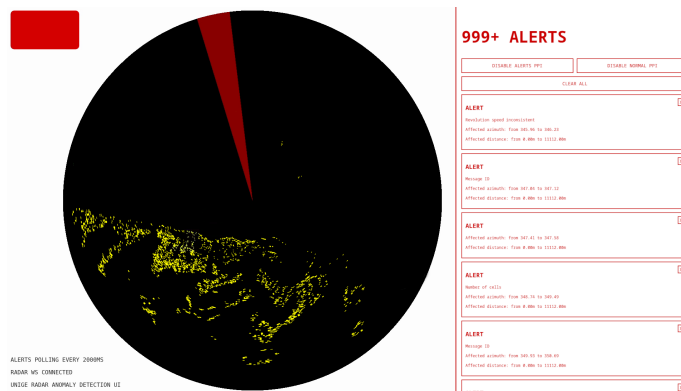


Fig. 13: Anomalies shown during V-B2.

The *anomaly receiver* appears 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. Figure 13 shows the anomalies produced by the attack in V-B2.

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 [49] that is integrated with the Shil [50] 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) [51] ship and radar simulator that implements an add-on for transmitting radar data using the ASTERIX CAT-240 protocol. Thus, we simulated sensors 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 for displaying the video data and tracking targets with ARPA. 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, amounts to 3761 lines of code, and supports cross-compiling to different architecture, e.g., x86 and ARM, and operating systems, e.g., Windows, Linux, and OS X. The version we used is a Linux executable file with a size of 1171KiB. Finally, the second virtual machine hosts our detection system.

B. Results

We generated on the BC simulator 25 instances of three scenarios that set the environment for executing the attacks we presented in Section V. Assuming that \mathcal{U} is the uniform random distribution, each instance features a number $\mathcal{U}_{\{2,8\}}$ of ships. We placed them at a distance of $\mathcal{U}_{\{3,5,5\}}$ nautical miles and $\pm \mathcal{U}_{\{10,80\}}$ degrees w.r.t. the bow of the victim, and moving 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 10. 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 the BC and tracked the surrounding vessels using ARPA during the test execution. We set our radar TCPA default alarm (see Section II-F) 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*.

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

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 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

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

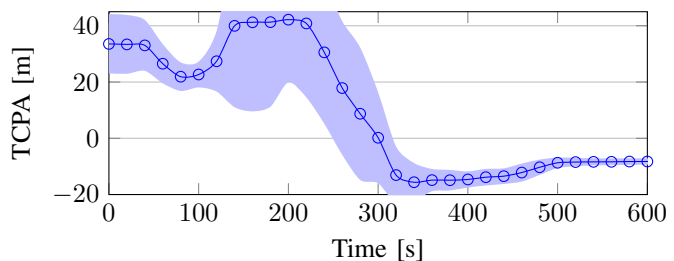
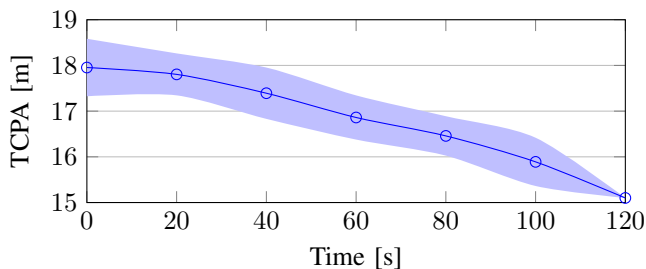


Fig. 14: Evolution of the TCPA during attacks V-B1 (left) and V-B2 (right).

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 the rotation speed to be constant within a revolution (P_5), and (iv) the number of aggregation in the historian within a fixed time period, i.e., enforcing the rotation speed to be constant across revolutions (P_6).

For each experiment, we recorded performance figures of the malware (see Table I) and the detection system (see Table II). We report the CPU and RAM usage for both attack and defense. In Table I, we also include statistics about ASTERIX traffic and the percentage of the malicious traffic compared to the legitimate traffic.

Then, we used information emitted by the ARPA system to evaluate the effectiveness of attacks.

For DoS attacks, we considered each experiment as a success if the radar image corruption caused at least one of the following two impacts on the ARPA system: (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 success conditions. We measured for each TTM the absolute error between the desired courses and speeds and the ones emitted by the ARPA system 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 14, 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. Figure 14 highlights that the two attacks had a success rate of 100% since the TCPA always complied with the expected trend.

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 the policies that each attack triggered during the experiments by considering the percentage of malicious packets that they matched.

C. Discussion

Concerning the techniques and requirements introduced in Section III, we put forward the following considerations related to our malware and the feasibility of the attacks. First,

the malware is a small and cross-compilable executable and can successfully carry out coarse-grained and fine-grained attacks using low computational and memory resources. These features allow an adversary to install it and execute the presented attacks on a wide range of INS configurations, including legacy and embedded systems. The experimental results indicate that the malware has a very limited footprint on both the usage of computational/memory resources and the amount malicious traffic w.r.t. the legitimate one. This fact confirms that the malware can act stealthy and execute PPI poisoning attacks without causing noticeable effects. For DoS attacks, we showed that it could leverage features of the ASTERIX protocol to obtain the above small footprint. Moreover, overhearing NMEA traffic allows the malware to acquire the information and requirements to operate independently and without communicating outside the INS.

Regarding attacks, the experiments executed in our cyber range with multi-ships scenarios and the commercial PPI proved their feasibility. Results also showed that the attacks could achieve a high degree of realism by precisely simulating the behavior of a vessel on a predetermined trajectory. Such realism and the capability of these attacks to inject AIS traffic for cheating the cross-checking with INS equipment show the high deception capability against maritime operators and the potential to cause catastrophic impacts.

Regarding 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 on-board configurations (P_1 , P_2) enabled the detection of all the attacks with high accuracy. The resulting performance highlighted that our system requires minimal resource footprint. Finally, it is worth noting that the detection operates only on packets header and can ensure similar performances on other antenna types, even with higher video resolutions.

VIII. CONCLUSION

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

Radar is an essential aid to ensure safe navigation, and the consequences of these attacks 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 a high level of 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 the awareness of the maritime operators also in response to these new types of attacks.

IX. ACKNOWLEDGMENTS

This work was partially supported by research funding from Fincantieri S.p.A. and a grant from the National Ph.D. Programme in Artificial Intelligence (Security and cybersecurity area). The research activities are carried out thank to the ShIL (Ship-In-the-Loop) research infrastructure, co-funded by Regione Liguria, University of Genova and DLTM under the program POR FESR LIGURIA 2014-2020 ASSE 1 "Research and Innovation (OT1)" Action 1.5.1 Notice "Support for research infrastructures considered critical / crucial for regional systems".

REFERENCES

- [1] International Maritime Organization, *SOLAS 2020 consolidated edition*, ser. IMO-publication, 2020.
- [2] "Strategy for the development and implementation of e-navigation," International Maritime Organization, MSC 85/26/Add.1 ANNEX 20. [Online]. Available: <https://u.garr.it/DSA6f>
- [3] "E-Navigation Strategy Implementation Plan - Update 1," International Maritime Organization, MSC.1/Circ.1595. [Online]. Available: <https://u.garr.it/OqShS>
- [4] "Adoption of the revised performance standards for integrated navigation systems (INS)," International Maritime Organization, RESOLUTION MSC.252(83). [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," *Journal of Transportation Safety & Security*, pp. 1–28, Apr. 2021. [Online]. Available: <https://doi.org/10.1080/19439962.2021.1909681>
- [6] "Performance standards for automatic radar plotting aids (arpas)," International Maritime Organization, RESOLUTION A.823(19).
- [7] *61162-1 Maritime Navigation and Radiocommunication Equipment and Systems—Digital Interfaces—Part 1: Single Talker and Multiple Listeners*, International Electrotechnical Commission Std., Rev. 2016.
- [8] *Specification for Surveillance Data Exchange – ASTERIX Category 240: Radar Video Transmission*, 1st ed., EUROCONTROL-SPEC-0149-240. [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, the International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 15, no. 3, pp. 519–530, 2021. [Online]. Available: <https://doi.org/10.12716/1001.15.03.04>
- [10] M. S. Lund, J. E. Gulland, O. S. Hareide, Ø. J. Rødseth, and K. O. C. Weum, "Integrity of integrated navigation systems," in *2018 IEEE Conference on Communications and Network Security (CNS)*, 2018, pp. 1–5. [Online]. Available: <https://doi.org/10.1109/CNS.2018.8433151>
- [11] O. S. Hareide, M. S. Jøsok, Øyvind and Lund, R. Ostnes, and K. Helkala, "Enhancing navigator competence by demonstrating maritime cyber security," *The Journal of Navigation*, vol. 71, no. 5, pp. 1025–1039, 2018.
- [12] "MITRE ATT&CK - Man in the Middle," <https://collaborate.mitre.org/attackics/index.php/Technique/T0830>, Mitre Corporation.
- [13] E. E. Casanovas, T. E. Buchailot, and F. Baigorria, "Vulnerability of radar protocol and proposed mitigation," in *2015 ITU Kaleidoscope: Trust in the Information Society (K-2015)*. IEEE, 2015, pp. 1–6. [Online]. Available: <https://doi.org/10.13052/jicts2245-800X.414>
- [14] G. C. Kessler, "Cybersecurity in the maritime domain," *USCG Proceedings of the Marine Safety & Security Council*, vol. 76, no. 1, p. 34, 2019. [Online]. Available: <https://commons.erau.edu/publication/1318>
- [15] B. Cain, D. S. E. Deering, B. Fenner, I. Kouvelas, and A. Thyagarajan, "Internet Group Management Protocol, Version 3," RFC 3376, Oct. 2002. [Online]. Available: <https://rfc-editor.org/rfc/rfc3376.txt>
- [16] *M.1371 Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band*, International Telecommunication Union Std., Rev. 5.

- [17] "Adoption of the Revised Performance Standards for Radar Equipment," International Maritime Organization, RESOLUTION MSC.192(79). [Online]. Available: <https://u.garr.it/zrLcN>
- [18] "Guidelines on annual testing of the automatic identification system (AIS)," International Maritime Organization, MSC.1/Circular.1252. [Online]. Available: https://www.imorules.com/MSCCIRC_1252.html
- [19] "Recommendation on performance standards for a universal shipborne automatic identification system (AIS)," International Maritime Organization, RESOLUTION MSC.74(69). [Online]. Available: <https://u.garr.it/pIRGf>
- [20] "Revised guidelines for the onboard operational use of shipborne Automatic Identification Systems (AIS)," International Maritime Organization, RESOLUTION A.1106(29). [Online]. Available: <https://u.garr.it/oNeQv>
- [21] A. Goudosis and S. Katsikas, "Secure AIS with Identity-Based Authentication and Encryption," *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 14, no. 2, 2020. [Online]. Available: <http://dx.doi.org/10.12716/1001.14.02.03>
- [22] S. Sciancalepore, P. Tedeschi, A. Aziz, and R. Di Pietro, "Auth-AIS: Secure, Flexible, and Backward-Compatible Authentication of Vessels AIS Broadcasts," *IEEE Transactions on Dependable and Secure Computing*, 2021. [Online]. Available: <https://doi.org/10.1109/TDSC.2021.3069428>
- [23] G. Kessler, "Protected AIS: a demonstration of capability scheme to provide authentication and message integrity," *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 14, no. 2, 2020. [Online]. Available: <http://dx.doi.org/10.12716/1001.14.02.02>
- [24] "Recommendation for the Protection of the AIS VHF Data Link," International Maritime Organization, RESOLUTION MSC.347(91). [Online]. Available: <https://u.garr.it/keE9d>
- [25] A. Dabrowski, S. Busch, and R. Stelzer, "A digital interface for imagery and control of a Navico/Lowrance broadband radar," in *Robotic Sailing*. Springer, 2011, pp. 169–181. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-22836-0_12
- [26] "Radar PI OpenCPN plugin," accessed on 5/11/2021. [Online]. Available: https://github.com/opencpn-radar-pi/radar_pi
- [27] O. LLC, *Radar Navigation and Maneuvering Board Manual*, 7th ed. OceanGrafix LLC, 2001. [Online]. Available: <https://msi.nga.mil/Publications/RNMB>
- [28] *IEC 62388 - Maritime navigation and radiocommunication equipment and systems - Shipborne radar - Performance requirements, methods of testing and required test results*, International Electrotechnical Commission Std., Rev. 2013.
- [29] *Specification for Surveillance Data Exchange - Part 1 All Purpose Structured EUROCONTROL Surveillance Information Exchange (ASTERIX)*, 3rd ed., 2020, EUROCONTROL-SPEC-0149. [Online]. Available: <https://www.eurocontrol.int/publication/eurocontrol-specification-surveillance-data-exchange-part-1>
- [30] D. G. Johnson and M. R. Warren, "Using ASTERIX CAT-240 for Radar Video Distribution—Practical Considerations from Deployed Applications," in *9th International Radar Symposium, 10Y14 December*, 2013. [Online]. Available: <https://www.cambridgepixel.com/site/assets/files/2485/asterix-cat240-for-video-distribution.pdf>
- [31] M. JANČÍK, D. Johannes, and P. JONÁŠ, "Security enhancements of the surveillance data exchange protocol "asterix"," *DEStech Transactions on Computer Science and Engineering*, no. cscbd, 2019. [Online]. Available: <http://dx.doi.org/10.12783/dtsc/cscbd2019/30009>
- [32] T. de Riberolles, J. Song, Y. Zou, G. Silvestre, and N. Larrieu, "Characterizing Radar Network Traffic: a first step towards spoofing attack detection," in *2020 IEEE Aerospace Conference*, 2020, pp. 1–8. [Online]. Available: <https://doi.org/10.1109/AERO47225.2020.9172292>
- [33] "Convention on the International Regulations for Preventing Collisions at Sea," International Maritime Organization, 1972. [Online]. Available: <https://www.imo.org/en/OurWork/Safety/Pages/Preventing-Collisions.aspx>
- [34] "MaCRA: a model-based framework for maritime cyber-risk assessment, author=Tam, Kimberly and Jones, Kevin," *WMU Journal of Maritime Affairs*, vol. 18, no. 1, pp. 129–163, 2019. [Online]. Available: <http://dx.doi.org/10.1007/s13437-019-00162-2>
- [35] "MITRE ATT&CK - Supply Chain Compromise," <https://attack.mitre.org/techniques/T1195/>, Mitre Corporation.
- [36] B. Svilicic, D. Brčić, S. Žuškin, and D. Kalebić, "Raising awareness on cyber security of ECDIS," *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 13, no. 1, 2019. [Online]. Available: <http://dx.doi.org/10.12716/1001.13.01.24>
- [37] Mass Soldal Lund, O. S. Hareide, and Ø. Jøsok, "An attack on an integrated navigation system," 2018. [Online]. Available: <https://brage.bibsys.no/xmlui/handle/11250/2568320>
- [38] Y. Dyravyy, "Preparing for Cyber Battleships – Electronic Chart Display and Information System Security," NCC Group, Tech. Rep., 2014. [Online]. Available: https://research.nccgroup.com/wp-content/uploads/2020/07/2014-03-03_-_ncc_group_-_whitepaper_-_cyber_battle_ship_v1-0.pdf
- [39] "The Guidelines on Cyber Security Onboard Ships," BIMCO, Tech. Rep., 2021, version 4. [Online]. Available: <https://www.bimco.org/about-us-and-our-members/publications/the-guidelines-on-cyber-security-onboard-ships>
- [40] The MITRE Corporation, "CAPEC-542: Targeted Malware," accessed on 14/02/2022. [Online]. Available: <https://capec.mitre.org/data/definitions/542.html>
- [41] 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," *Leading Issues in Information Warfare & Security Research*, vol. 1, 2011. [Online]. Available: <https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf>
- [42] T. Vincenty, "Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations," *Survey Review*, vol. 23, no. 176, pp. 88–93, 1975. [Online]. Available: <https://doi.org/10.1179/sre.1975.23.176.88>
- [43] M. Kristic, S. Žuškin, D. Brčić, and M. Car, "Overreliance on ECDIS technology: A challenge for safe navigation," *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 15, no. 2, pp. 277–287, 2021. [Online]. Available: <https://doi.org/10.12716/1001.15.02.02>
- [44] "Performance Standards for Radar Equipment," International Maritime Organization, RESOLUTION A.477(XII). [Online]. Available: <https://u.garr.it/G6iJP>
- [45] "Adoption of performance standards for bridge alert management," International Maritime Organization, MSC.302(87). [Online]. Available: <https://u.garr.it/Knbc7>
- [46] N. D. Matsakis and F. S. Klock, "The rust language," *ACM SIGAda Ada Letters*, vol. 34, no. 3, pp. 103–104, Nov. 2014. [Online]. Available: <https://doi.org/10.1145/2692956.2663188>
- [47] Cloud Native Computing Foundation, "Open policy agent," accessed on 29/01/2022. [Online]. Available: <https://www.openpolicyagent.org/>
- [48] R. Ierusalimsky, L. H. de Figueiredo, and W. C. Filho, "Lua—an extensible extension language," *Software: Practice and Experience*, vol. 26, no. 6, pp. 635–652, Jun. 1996. [Online]. Available: [https://doi.org/10.1002/\(sici\)1097-024x\(199606\)26:6\(635::aid-spe26\)3.0.co;2-p](https://doi.org/10.1002/(sici)1097-024x(199606)26:6(635::aid-spe26)3.0.co;2-p)
- [49] E. Russo, G. Costa, and A. Armando, "Building next generation cyber ranges with CRACK," *Computers & Security*, vol. 95, p. 101837, Aug. 2020. [Online]. Available: <https://doi.org/10.1016/j.cose.2020.101837>
- [50] 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 *2020 AEIT International Annual Conference (AEIT)*. IEEE, Sep. 2020. [Online]. Available: <https://doi.org/10.23919/aeit50178.2020.9241110>
- [51] J. Packer, "Bridge command," accessed on 29/01/2022. [Online]. Available: <https://www.bridgecommand.co.uk/>