A comparative risk analysis on CyberShip system with STPA-Sec, STRIDE and CORAS

Sahay, Rishikesh; Estay, D. A. Sepulveda; Meng, Weizhi; Jensen, Christian D.; Barfod, Michael Bruhn

Published in:
Computers and Security

Link to article, DOI:
10.1016/j.cose.2023.103179

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Computers & Security 128 (2023) 103179

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

A comparative risk analysis on CyberShip system with STPA-Sec, STRIDE and CORAS

Rishikesh Sahay[a], D.A. Sepulveda Estay[b], Weizhi Meng[c,*], Christian D. Jensen[c], Michael Bruhn Barfod[d]

[a] Business Management Department, Oregon Institute of Technology, Klamath Falls, OR 97601, United states
[b] Digitalization Group, Rigshospitalet, Denmark
[c] Department of Applied Mathematics & Computer Science, Technical University of Denmark, Kgs. Lyngby, DK-2800, Denmark
[d] DTU Management Engineering, Technical University of Denmark, DK-2800, Kgs. Lyngby, Denmark

ARTICLE INFO

Article history:
Received 15 December 2022
Revised 13 February 2023
Accepted 8 March 2023
Available online 11 March 2023

Keywords:
Cyber ship
Cyber physical systems (CPS)
STPA
Cyber risk assessment
STRIDE
CORAS

ABSTRACT

The widespread use of software-intensive cyber systems in critical infrastructures such as ships (CyberShips) has brought huge benefits, yet it has also opened new avenues for cyber attacks to potentially disrupt operations. Cyber risk assessment plays a vital role in identifying cyber threats and vulnerabilities that can be exploited to compromise cyber systems. Understanding the nature of cyber threats and their potential risks and impact is essential to improve the security and resilience of cyber systems, and to build systems that are secure by design and better prepared to detect and mitigate cyber attacks. A number of methodologies have been proposed to carry out these analyses. This paper evaluates and compares the application of three risk assessment methodologies: system theoretic process analysis (STPA-Sec), STRIDE and CORAS for identifying threats and vulnerabilities in a CyberShip system. We specifically selected these three methodologies because they identify threats not only at the component level, but also threats or hazards caused due to the interaction between components, resulting in sets of threats identified with each methodology and relevant differences. Moreover, STPA-Sec, which is a variant of the STPA, is widely used for safety and security analysis of cyber physical systems (CPS); CORAS offers a framework to perform cyber risk assessment in a top-down approach that aligns with STPA-Sec; and STRIDE (Spoofing, Tampering, Repudiation,Information disclosure, Denial of Service, Elevation of Privilege) considers threat at the component level as well as during the interaction that is similar to STPA-Sec. As a result of this analysis, this paper highlights the pros and cons of these methodologies, illustrates areas of special applicability, and suggests that their complementary use as threats identified through STRIDE can be used as an input to CORAS and STPA-Sec to make these methods more structured.

1. Introduction

Modern day ship systems are highly sophisticated, complex and dependent on the effectiveness of software-based systems for operation. Until 15–20 years ago, these ship systems were not connected to the Internet, security was restricted only to safeguarding the physical infrastructure (The Cyber Threat Against Operational Systems on Ships, 2020). The present internet connectivity in these systems, despite providing ease of operation, yet has also exposed them to cyber threats and vulnerabilities which are difficult to prevent with a strategy based only on physical infrastructure safety. Additionally, the advent of Industry 4.0 in the maritime industry is advancing the use of process digitalization, and the use of machine learning for data analysis allowing automated decision making and operation. This dramatically expands the attack surface for cyber attacks on ship systems.

In June 2017, A.P. Moller-Maersk was attacked by a malware known as Not-Petya that left its IT systems inoperable for several weeks (Capano, 2021). The estimated damage due to this attack on Maersk is $300 million (Shackelford, 2020). It caused the disruption in the global supply chain and impacted many other companies along with Maersk. Many reports suggest that the attack caused as much as $10 billion in damages in total (Tehrani, 2017; Wolff, 2021). Beyond the immediate effects that this attack had on Maersk's bottom line, this was another clear evidence that cyber

* Corresponding author.
E-mail addresses: rishikesh.sahay@oit.edu (R. Sahay), daniel.alberto.sepulveda.estay@regionh.dk (D.A.S. Estay), weme@dtu.dk (W. Meng), cdje@dtu.dk (C.D. Jensen), mbba@dtu.dk (M.B. Barfod).

https://doi.org/10.1016/j.cose.2023.103179

attacks that can go beyond the loss or corruption of data, to cause operational disruptions, are also a reality in the shipping industry.

Moreover, nowadays shipbuilders are also trying to innovate and build remote controlled automated ship. For instance, the Nippon foundation launched MEGURI2040 fully autonomous ship program in February 2020. On January 2022, a fully autonomous small boat successfully sailed in the waters around Sarushima in Japan (The Nippon Foundation MEGURI2040 Fully Autonomous Ship Program, 2022). These autonomous ships are equipped with Operational Technology (OT) systems, which are interconnected with each other to automatically navigate the ship. According to the Danish Centre for Cyber Security (CFCS), cyber threats against OT systems of ships is high (The Cyber Threat Against Operational Systems on Ships, 2020). CFCS assessed that the OT systems of ships can be attractive target for cyber criminals, since they are important for the shipping company, and they can leverage it for getting ransom. The CFCS's report highlighted that in 2017 a ransomware attack spread through administrative system to the ship's OT systems and disrupted the power supply (The Cyber Threat Against Operational Systems on Ships, 2020). The crew was not able to solve the issue and had to call help from the IT support (The Cyber Threat Against Operational Systems on Ships, 2020). Some works have highlighted many cyber security risks on current autonomous ships (Kavallieratos et al., 2019b; Tam and Jones, 2018).

Many other less visible attacks are happening to shipping operations every day, in a trend that is showing no signs of slowing down (The Cyber Threat Against Operational Systems on Ships, 2020). Companies in the shipping industry, formerly inclined to invest mainly in cyber security, have increasing evidence that failing to avert a cyber-attack is more and more likely. Cyber-resilience, the capacity to react to cyber-attacks, becomes thus desirable, through for example, designing a system with the ability to cope with a cyber attack already under way through DCRA resilience, namely Detection, Contention, Recovery and Adaptability.

In light of these findings, it is vital to improve the cyber security of ship systems. To strengthen the cyber security of these systems, it is important to have a holistic view of ship systems. Therefore, the first step is to prepare an architecture and identify System under Consideration (SuC) for cyber risk assessment. Second step is to perform cyber risk assessment and mitigation. Moreover, cyber risk assessment should not only consider the risk at the component level but also the way it can propagate to interconnected components and compromise the whole system. Components in these critical infrastructures such as ships are interconnected, therefore the disruption in one part of the system can trigger domino effect causing the damage to the whole system. So, having a holistic system-of-systems approach is important in cyber risk analysis of critical infrastructures.

As a result, this paper compares three different risk assessment methodologies namely STPA, STRIDE and CORAS, by applying these methodologies on the CyberShip framework (Sepulveda Estay et al., 2020), a framework for representing the cyber physical components on a ship. These methodologies follow a top down approach in analyzing risks, and they also consider the risk as a result of interactions between the different components of the system.

The main purpose of comparing these methodologies is to investigate how they perform in analyzing cyber and safety risks on the critical infrastructure CyberShip systems used as example in this paper, and to explore the utility of using some of these methods together.

The rest of the paper is organized as follows. Related extant literature on threat modelling through the use of systemic risk analyses is described in Section 2. Section 3 describes the CyberShip framework and its different components. Thereafter this CyberShip framework is analyzed using STPA-Sec in Section 4, by using the STRIDE method in Section 5, and through the use of the CORAS method in Section 6. Thereafter these analyses are compared in Section 7. Section 8 presents a discussion of the comparison of these risk analyses, and finally, Section 9 concludes the paper proposing areas of future work.

2. Related works

A number of guidelines have been developed aiming to address the growing concern of cyber attacks in the maritime industry (Guide for Cybersecurity Implementation for the Marine and Offshore Industries, 2021; Royce, 2016; The Guidelines on Cyber Security Onboard Ships, 2017). These guidelines provide a framework for securing ship systems and its operations. Moreover, a number of studies have been done on the cyber risk assessment of the ship systems and networks. MaCRA (Maritime Cyber Risk Assessment) model developed by Tam and Jones (2019) dynamically responds to the changes done within ship system and to threats. It considers the vulnerabilities in the system as well as how easily the system can be exploited by adversaries. The authors in Kavallieratos et al. (2019b) proposed a generic system architecture of autonomous ship and analysed threats and risks using STRIDE model. The AAWA (Advanced Autonomous Waterborne Applications) project led by Rolls Royce has also done an important work in highlighting cyber security and safety issues in autonomous ships (Autonomous Ships The Next Step, 2021). System Theoretic Process Analysis (STPA) have been used in analyzing cyber security risks in ship systems. For example, STPA has been used to derive verification objectives and hazardous scenarios in maritime systems (Rokseth et al., 2018), to identify causal scenarios and factors that drive maritime incidents and accidents (Puisa et al., 2018) and it has been advanced in the conceptual design autonomous vessels (Banda et al., 2019). STPA has been used to identify the conditions of risk for the case of remotely-controlled merchant vessels (Wróbel et al., 2018), their work focused mainly on the overall shipping operation, considering the vessel, the shore facilities, the environment and the organizational environment, all in an aggregated level. Recent research that describes the multiple control systems on board standard commercial ships (Hyra, 2019), reflects a need for greater detail in the systemic analysis of risks, a suggestion that is developed in this work. The authors in Omitola et al. (2018) used STPA-Sec, which is a variant of STPA method to analyse cyber attacks targeting navigation system of ships. STPA-Sec focuses on unsafe control actions (UCA), which occur due to cyber threats. In (Glomsrud, 2019), the authors suggested to use the STPA with attack tree for safety and security analysis of autonomous vessels. Kavallieratos et al. (2020) employed STPA method with security analysis to find the comprehensive list of security and safety requirements at the system design stage. In connection to cyber risks, authors have also proposed the combined use of STRIDE with STPA (Kaneko et al., 2018).

More recently a group led by Lim et al. (2018) has identified models and computational algorithms used in maritime risk analysis and mentioned that the development of models was the most common type of maritime risk analysis research. Akpan et al. (2022) highlighted the cyber risks in the various OT (Operational Technology) components of the ship. Grigoriadis et al. (2022) proposed an adaptive security framework to identify situational risks and deploy policies to mitigate them. SafeSec (Kavallieratos et al., 2020) offers a framework to identify security and safety objectives in the ship systems. This framework combines model based method for security requirements engineering with STPA method to identify safety risks. Wang et al. (2004) described control engineering techniques that could be used for the risk analysis in the marine industry, high-

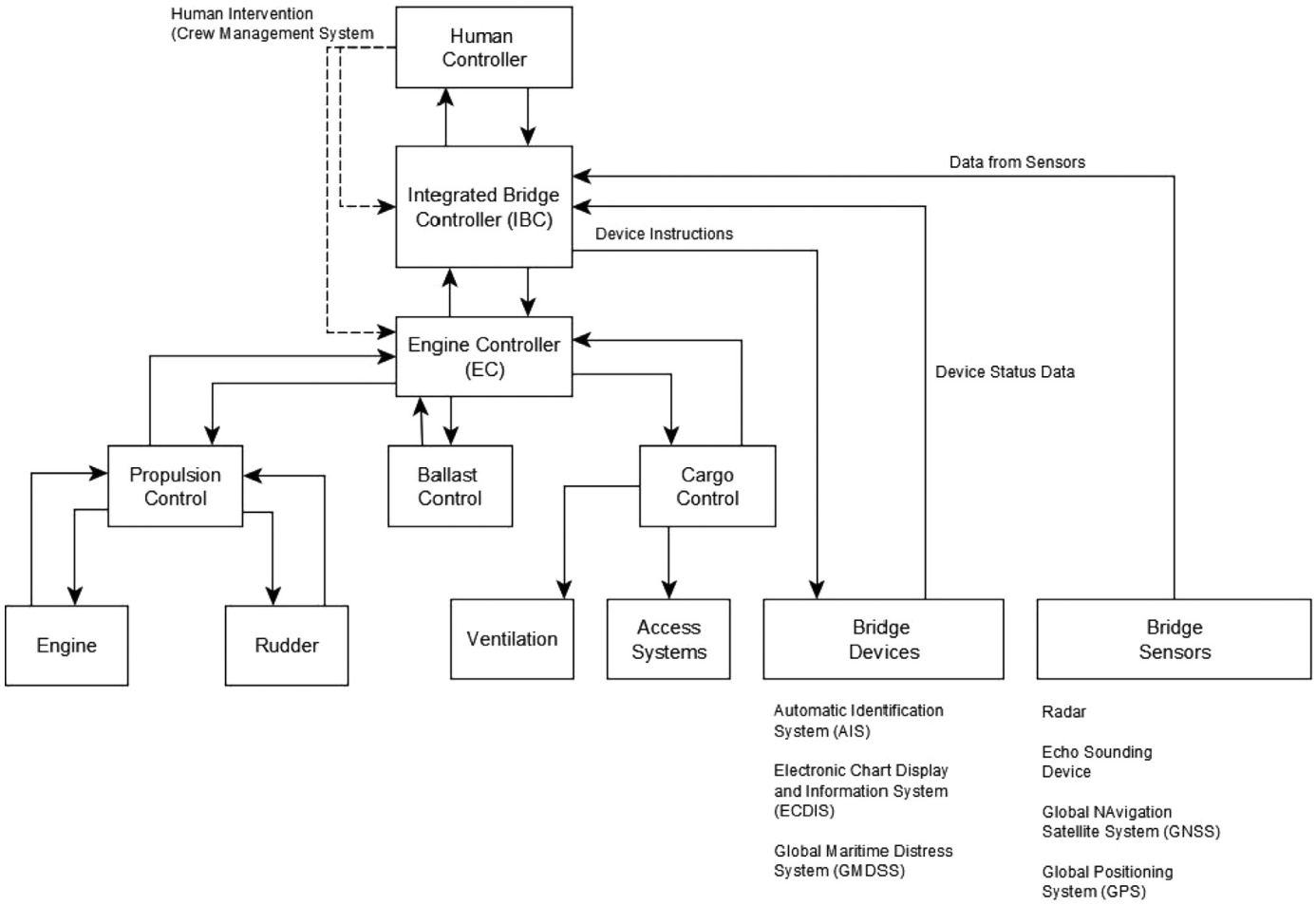

Fig. 1. CyberShip framework.

lighting that "*a framework with a holistic nature is desirable for risk assessment of large engineering systems*".

Most of the previous studies have been conducted to identify cyber and safety risks in the ship systems either with one method or by combining two methods. However, this paper identifies cyber and safety risks with STPA, CORAS, and STRIDE methods, as they can identify risks in a top down approach and offers a structured framework for risk assessment. The aim is to find out how these methods perform in terms of identifying security and safety risks when applied on the same framework CyberShip in this case.

3. The CyberShip model

CyberShip is a Cyber Physical System (Kayan et al., 2022) generated from the widespread adoption of Information and Communication Technologies (ICT) in maritime operations, and as a result increasing the attack surface of ships to cyber attacks (Hyra, 2019). The risk analyses presented in this paper are based on a CyberShip model of a system representing the interactions between infrastructure and information.

The CyberShip model used in this study is a hierarchical control structure as shown in Fig. 1. This representation is consistent with the current literature (Chaal et al., 2020; Hyra, 2019; Sahay et al., 2019; Sepulveda Estay et al., 2020) and includes several cyber-physical systems that are important for the safe operations of the ship. First, important components of the CyberShip framework are highlighted and then the working of the CyberShip framework is presented.

- **Bridge Devices:** These devices can sense the surrounding environment and provide the information to the ship controller for centralized process control and decision making. Bridge devices can be connected to shore-side networks for software updates, or be updated through removable media such as USB. Radar, Automatic Identification System (AIS), Electronic Chart Display System (ECDIS), Global Maritime Distress System (GMDSS), Global Navigation Satellite System (GNSS), and Echo Sounding are examples of bridge devices on the ship (The Guidelines on Cyber Security Onboard Ships, 2017).
- **Integrated Bridge Controller (IBC):** It supervises the operation of bridge devices (The Guidelines on Cyber Security Onboard Ships, 2017) by receiving data from sensors in these devices and providing a centralized interface to the crew on-board to access the data and to make decisions. The IBC also issues control commands to the engine controller, such as start/stop of the propulsion control system, rerouting the ship, and increase or decrease water level in the ballast, depending on the information from the bridge devices (The Guidelines on Cyber Security Onboard Ships, 2017).
- **Engine Controller:** It controls all the systems related to power generation and propulsion (Final Report:Autonomous Engine Room, 2015). It gathers data related to speed, rudder angle, and propeller, and it monitors the engine load, fuel consumption, and water level in the ballast compartment. Depending on the information from the integrated bridge controller, the engine controller commands and controls the propulsion control system to increase or decrease the speed of the ship. Furthermore,

it also sends the command to increase or decrease the level of water in the ballast compartment depending on the information from the bridge system.

- **Ballast Water Control:** It supervises the operation of the the ballast tank system in the ship used for draft and balance control (Process map for Autonomous Navigation, 2014). Ballast tanks are compartments within a ship that hold water, which is used to provide stability, by adjusting the water level in the tank. If the water in the ballast tanks is pumped out temporarily, this can reduce the draft of the vessel. Depending on the model of the ship, the ballast water control is independent of the engine system.
- **Propulsion Control:** It controls the propeller, rudder and steering of the ship. Propulsion control acts on the inputs from the engine control and provides the information to the engine controller such as speed of the ship, fuel level, engine load, etc (The Guidelines on Cyber Security Onboard Ships, 2017).
- **Cargo Management System:** Computer systems used for the management and control of cargo may interface with a variety of other systems ashore (The Guidelines on Cyber Security Onboard Ships, 2017). These systems may include shipment tracking details available to shippers via the Internet. Interfaces of this kind can make cargo management systems and data in cargo vulnerable to cyber attacks.
- **Human factors** also have to be considered in a cyber-ship model, as only in highly automated shipping systems, there is no expected interaction between human operators and the shipping systems. Examples of human factors that can have a disruptive effect through cyber-attacks include events such as unauthorized system entry (software level) or rewiring (hardware level).

3.1. Working of CyberShip model

CyberShip is an automated framework but humans (e.g., captain or crew) can also intervene when required. The operational workflow of CyberShip can be described as follows:

- As shown in Fig. 1, human controller (i.e., captain) can start the CyberShip by logging into the `Engine Controller` and providing the "start" control action.
- As we can see in Fig. 1, `Integrated Bridge Controller` receives data from bridge devices and issues control commands to the `Engine controller`. Control commands can be such as start/stop engine, increase/decrease the speed, reroute the ship, etc.
- Generally, the `Engine controller` resides in the separate room of the ship. As shown in Fig. 1, `Engine controller` receives the information from the `IBC` about the detailed position of the ship. For example, the IBC provides the information to the `Engine controller` about whether the CyberShip is sailing in the shallow water or deep water. Based on this information, the `Engine controller` issues a control command to the `Ballast control` to increase or decrease the water level in the ballast tank. Similarly, when the IBC receives the information about the congestion or problem in a particular route, it can provide information to the `Engine controller` to reroute the ship through another direction. For instance, the IBC can issue control commands like "reroute" the ship in case of a problem in the route that CyberShip is following. The IBC receives such information either from the shore center or from the bridge devices that are connected to it. For brevity purpose, we have not shown those details in Fig. 1.
- Controllers also receive the feedback once the control action has been performed. For instance, once the ballast control has performed the control action, such as increase/decrease the water level in the ballast tank, then it provides the feedback to the engine controller. In a similar way, `Engine controller` provides the feedback to the `IBC`.

3.2. Scenarios of human intervention

CyberShip is an autonomous vessel that relies on interconnected components for its operations. However, many times human interaction is required for the operation of the CyberShip. Identifying scenarios of human interaction is required for the safe and secure operation of the CyberShip. Some possible scenarios of interaction between crew members and CyberShip are mentioned below:

- Captain has privilege to access the engine controller to start, stop, increase, decrease and reroute the CyberShip.
- If ECDIS (Electronic Chart Display System) is not working because of malware attack or due to component failure then crew can use paper map to check the route.
- Captain and crew members from time to time check outside for any potential danger. Because there can be chances that some bridge devices are compromised or not working properly and that can cause accidents. If they find any potential danger, then captain can log into the engine controller to issue control commands to avoid any accidents. For example, if they will find another ship coming close then captain can log into the engine controller to divert the ship and make a safe distance.
- In case of other incidents like fire on the CyberShip, captain can log into the engine controller to stop the engine and avoid further damage.

Therefore, in case of any emergency due to component failure or because of cyber attacks captain has privilege to login into the system to issue commands to avoid accidents.

3.3. Classes of adversaries and threat vector

Identifying classes of adversaries and threat vector is an important step for risk assessment. For the CyberShip assessment, we have taken into consideration of internal and external adversaries.

- **Internal adversary:** Nowadays, how to protect cyber-physical systems from insiders is a major concern for many organizations. Because the disgruntled employee with a lot of privileges and technical knowledge may know how to cause severe damage to the ship infrastructure. Moreover, internal employees can also cause damage unintentionally. For instance, crew members may connect infected USB drive unintentionally or misconfigure the system that can damage the CyberShip system.
- **External adversary:** External adversaries are external to the organization and they try to either damage the system or steal confidential information about the organization that can impact the reputation of the company.

Below, we highlight the different types of threat vectors that can be used by internal or external adversaries.

- **Malware Infection through removable media:** Internal or external adversaries who have physical access to the system can connect USB containing malware that can infect and compromise the system. During maintenance and integration, USB drives are widely used. However, USB drive may have been infected through the office system or it may have been connected to an infected private laptop.
- **Malware infection through Internet/Intranet:** Nowadays, IT and OT (Operational Technology) systems are connected with each other. Malware can spread through the infected IT systems in the enterprise network, as they are connected with CyberShip.

- **Distributed Denial of Service Attack (DDoS):** DDoS or DoS (Denial of Service) can impact the operation of the CyberShip. If CyberShip systems are flooded with spurious packets, then control actions cannot be performed, which may cause severe damages.
- **Intrusion through remote access into CyberShip:** CyberShip can be accessed remotely from shore center, this has created another attack surface that is important to handle.
- **Social Engineering Attacks:** Social Engineering attacks are used to obtain unauthorized access into the IT systems. Generally, social engineering attacks rely on human traits such as curiosity, fear, etc. Through social engineering attacks such as phishing emails, adversaries try to obtain victim's credentials or distribute malware through malicious attachments. CyberShip system is also vulnerable to these attacks since they are connected with the IT systems. Moreover, crew members have a chance to open malicious attachments in their emails that can impact CyberShip.
- **Component failure:** Because of component failure, control actions may not be performed, which can impact the operations of the CyberShip.

3.4. Execution of cyber kill chain for adversaries

Cyber kill chain is a step-by-step approach required by an adversary to successfully compromise the system/network (Seven Stages of Cyber Kill Chain Supplementary Reading, 2017). It highlights the common stages in various cyber attacks and the stage at which attacks can be detected and prevented. Seven stages of cyber kill chain are highlighted below along with the way it works for each adversary against CyberShip.

- Reconnaissance: In the reconnaissance stage, adversaries explore the vulnerabilities in the system. Internal adversary can perform a scan to identify the vulnerabilities in the CyberShip system. Moreover, they can also collect credentials of critical systems through shoulder surfing. Generally, external attackers need to perform an active network and system scan to collect the information about the CyberShip systems.
- Weaponization: In this stage, attackers should prepare an attack vector to exploit the vulnerabilities that have been found in reconnaissance stage. For instance, internal adversary can use the collected credentials to log into the CyberShip components such as `Engine controller` to change the parameters that can damage the `Engine controller`. External adversaries can develop a malware or trojan to get access to the system to launch cyber attacks.
- Delivery: Generally, in this step, adversaries can deliver the malicious payload to the victim by email. In CyberShip, internal adversaries can deploy malicious payload using USB. Phishing emails can be used to deliver malicious payload to CyberShip system by external attackers. External attackers can send the phishing emails to captain or crew members with malicious attachments.
- Exploitation: In this stage, malicious code is executed by the victim. Internal adversaries can log into the CyberShip system such as `Engine Controller`, `Integrated Bridge Controller` or `Ballast control` and run the malicious code intentionally. However, external adversaries need to rely on crew members or captain to run the malicious code that can help them to get access to the CyberShip system. Crew members or captain may open their emails and run the malicious payload that can provide external attackers to obtain the access to CyberShip.
- Installation: The execution of malicious payload can trigger installation of additional malware or trojan in the CyberShip systems that can help adversaries to maintain the access in the CyberShip system.
- Command and Control: Once adversaries gain access to the CyberShip system by installing trojan or malware, they need to have a command and control connection to actually perform the attack and damage the system. For instance, trojan needs control command to perform any actions such as sniffing.
- Actions: At this stage, adversaries' mission is accomplished. They can encrypt critical control-related files, issue commands to shutdown the engine, change the parameters in the system that can damage the CyberShip system.

3.5. System under consideration (SuC)

Generally, organizations have multiple control systems, any of these control systems may be specified as an SuC (IEC, 2020). For the purpose of performing cyber risk analysis, SuCs must be identified. Architecture diagrams and dataflows can be used to highlight the components that are included in the SuC. For the cyber risk assessment, SuCs include:

- Integrated Bridge Controller
- Engine Controller
- Ballast Tank/Control

SuCs for STPA-Sec, STRIDE and CORAS assessment is shown with the architecture diagrams in Section 4.1, Section 5.2 and Section 6.

4. CyberShip analysis using STPA-Sec

This section first describes the STPA risk analysis methodology, and then presents the control structure of a `CyberShip` ballast control system. A high level control diagram of a `CyberShip` ballast control system is represented in Fig. 2. This section closes by identifying undesirable behaviours to be avoided and the control actions required to avoid hazardous situations.

4.1. STPA: systems theoretic process analysis

The STPA is a risk analysis methodology for safety and security, based on systems theory rather than traditional analytic reduction and reliability theories. It conceptualizes losses as a result of the inadequate interaction between components in the system due to a lack of adequate safety constraints. Consequently, safe and secure operation is seen as an emergent property resulting from the interactions between system components and the environment (Leveson, 2011).

In STPA, events that lead to a system failure are known as `accidents`, and these can occur through component failure, for example, condition that has been extensively analyzed through failure mode analysis methods such as FMEA or HAZOP (Foussard and Denis-Remis, 2014). Other more subtle system failure types are caused by unintended component interactions even when no component failure occurs.

Components are controlled through the enforcement of constraints and STPA assumes that inadequate constraints at different system levels may lead to system failures and accidents. Once a control structure and the interactions among components are represented, STPA suggests a safety and security analysis from a broad perspective including aspects such as physical, logic and information, social, operational, and managerial.

By analyzing the system's control structure, STPA represents how the interaction of different components can result in a safe and secure system, particularly by proposing requirements that prevent unsafe control actions, a result that is not possible with the traditional risk assessment mechanisms.

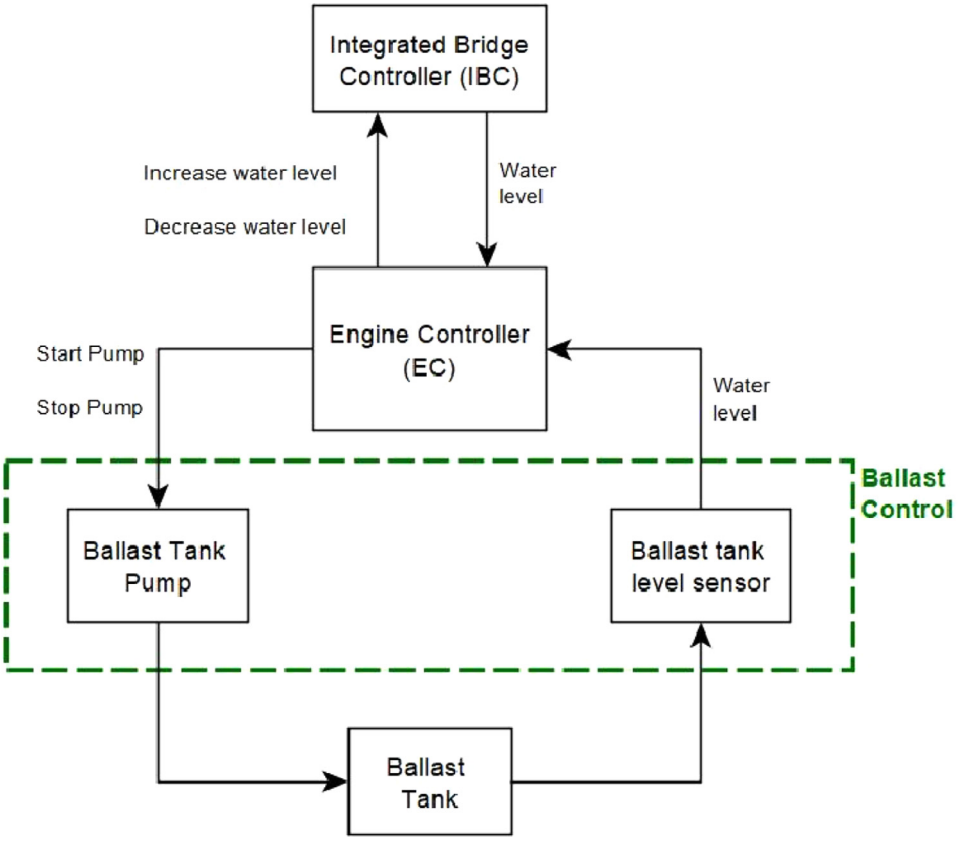

Fig. 2. High-level structure of the ballast control.

An STPA-Sec method identifies potentially unsafe control actions by analyzing how these can lead to an accident. Four ways in which a control action may be unsafe are (1) if a control action is executed, leading to a hazard, (2) if a control action is not executed, leading to a hazard, (3) if a control action is provided too early or too late, leading to a hazard, and (4) if a control action is executed for too long or stopped soon, leading to a hazard (Leveson, 2011). After the unsafe control actions have been identified, the next step involves investigating the system control structure to identify conditions, which can lead to these unsafe control actions happened.

First, the control loops are identified in the system under analysis. Second, the hazardous control actions are identified. Third, these hazardous control actions are used to define security requirements and constraints. Finally, causal scenarios are identified that can lead to a violation of either safety or security constraints.

By representing cyber and safety risks through a control system structure, it reflects that cyber-attacks are not events that happen from external sources, but rather events in which a system such as CyberShip are "*mis-designed*" to experience. In this approach, risky cyber-events are *unintended consequences* resulting from incomplete requirements at the time of system design.

A systemic analysis such as STPA-Sec follows a process to identify the "unintended" design that creates cyber-vulnerabilities, and suggests design changes through which cyber-vulnerable behaviour is less likely or no longer possible.

Assumptions. STPA-Sec uses base assumptions for its application, which are subject to analysis and control in order to validate its use. To begin, STPA requires the problem to be modeled as an analytic, causal system with a component hierarchy. This requires the identification of the system's components (analytic), their mutual interactions and influence (causal), and the level of control each component has over other components or subsystems (hierarchy).

Table 1
Undesirable behaviour or loss.

A1	Shipment late or non arriving
A2	Loss/harm to life of passengers/crew
A3	Wrong or non delivery to customers
A4	Damage to the ship
A5	Damage to the cargo
A6	Reputational loss

The system's dis-aggregation, or the level of aggregation of the components, is another explicit choice, argued by the problem objective for the model's creation. STPA is also founded on the identification and representation of a circular causality for regulation and control, on which a significant portion of its risk analysis is centered.

4.2. Unacceptable losses and hazards

In the first step, we identify all the losses that are considered unacceptable. In this stage, we identify what are the essential functions, which must be defended and how the disruption of these functions can lead to the undesirable outcomes. The analysis goes from top to down, from high-level to concrete. Table 1 shows losses that are unacceptable and must be avoided. These include service related losses such as late arrival of the shipment, wrong delivery to customers, or damage to the cargo, operational losses such as damage to the ship, human losses, and business losses from damage to the reputation of the shipping organization.

Based on the control structure as shown in Fig. 2, our work lists the hazards in Table 2, which can lead to the loss shown in Table 1. Hazard is defined as a condition, which can lead to high-level loss (Leveson, 2011).

Table 2
Hazards.

H1	Uncontrolled maneuvering of the ship
H2	Unidentified cargo items/wrong cargo data
H3	Incorrect functioning of ship components
H4	Uncontrolled transmission of data
H5	Uncontrolled data being transmitted

The Next step of analysis involves the identification of unsafe control actions (UCAs), which can lead to the hazards as shown in Table 2.

The CyberShip control structure, as shown in Fig. 2, combined with the identified high-level losses and hazards can set the foundation for STPA analysis. We consider the control actions `start pump` from Engine Controller (EC) to the ballast pump, and then identified the UCAs that can cause hazards, in any of the four types outlined by Leveson (2011).

Table 3 shows the list of unsafe control actions related to the actions performed by ballast pump based on the information provided by the Engine controller (EC). For instance, `start pump` action leads to increase or decrease of the level of water in the ballast tank. Increase or decrease of the water to a wrong level in the ballast tank can imbalance and damage the ship. The Engine controller might have received wrong parameters from the Integrated Bridge Controller, which can cause EC to initiate `start pump` action with the wrong parameters, resulting in a hazard.

The `Start pump` action can cause hazards in different conditions. For instance, if the ship is sailing through a shallow or deep water, then it may require to increase or decrease the level of water in the ballast tank to balance the ship depending on the scenario. However, if the `Engine Controller` is compromised by an external adversary or is not functioning properly because of the component failure, then it can damage the ship.

It should be noted that the attacker can modify the parameters instructing the ballast tank to increase or decrease the water to a wrong level, which can sink the ship.

This example highlights a main advantage of this methodology that the individual components are working well but the vulnerability lies in the interaction between different components. Table 3 highlights example of conditions when the control action CA1: `start pump` can become unsafe and lead to the undesirable losses as shown in Table 1.

4.3. System security constraint and security requirements

The analysis of unsafe control actions is used to suggest design requirements and constraints. These suggestions result from the conditions where the control actions become unsafe.

For example, the following constraint should be specified: the `start pump` action must not be provided if the water level information is not received from `Integrated Bridge Controller`. This responds directly to **UCA1.4**. The specific implementation of this constraint is not specified, as it can be achieved in different ways.

In the same way, a requirement (a risk boundary) should be set to avoid the increase or decrease of water level in the ballast tank to a dangerous level. This boundary will provide protection against commands issued from a compromised `Integrated Bridge Controller` to `Engine Controller` with wrong water level information, or avoid damage caused in case the `start pump` action is applied for too long or too short time.

Following this process, system constraints and requirements can be proposed for scenarios derived from the analysis of the other `Control Actions` designed in the system. Examples of other constraints and requirements are shown in Table 4.

5. CyberShip analysis using STRIDE

5.1. Methodology

STRIDE stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege. The method was developed by Loren Kohnfelder and Praerit Garg (1999), Kavallieratos et al. (2019a). Spoofing is the ability of the attacker to pretend as someone or something else. Generally, spoofing violates the authentication property of the system. The modification or disruption of network or data of the system is known as tampering, which violates the integrity of the system. Repudiation is a threat that refers to someone's allegation that didn't perform an action that impacts the system's operation. It violates the non-repudiation of the system. Information disclosure is a threat, which discloses confidential information to the people who are not supposed to have access to it. It violates the confidentiality of the system. The threat of Denial of Service (DoS) disrupts the availability of the system by consuming the resources required for the system to operate. STRIDE model helps in identifying the potential threats and vulnerabilities during the design phase of the system.

5.2. STRIDE application on CyberShip

In this Section, STRIDE analysis on the CyberShip framework is presented, considering components and their interactions with each other. It helps to get results which is valid regardless of its deployment in the framework. Due to space constraints, STRIDE is applied on critical components of the CyberShip framework. 'I', 'L' and 'R' denotes impact, likelihood and risk in theTables 7–9.

Figure 3 shows SuCs with data flows for the STRIDE analysis. Integrated bridge controller, engine controller and ballast tank are considered for the analysis. It analyzes threats against each component that could be exploited by an adversary to compromise the whole system. There are two ways to perform STRIDE analysis: 1) STRIDE-per-element; and 2) STRIDE-per-interaction. STRIDE-per-element focuses on a set of threats against each components, which makes it easier to enumerate threats. However, many times threats come up because of interactions among the components. STRIDE-per-interaction aims to find threats against interaction between the components.

As we can see, the above tables show the high, medium and low level threats per components. It can help in designing effective mitigation solutions for each threat according to the different components' requirements and risk levels.

Before delving into the STRIDE analysis of CyberShip, we highlight some assumptions for the analysis, which are described as follows:

- In this analysis, we assume that there are some countermeasures deployed but they are not enough.
- We also assume that the systems are not directly exposed to the Internet, i.e., they are protected by firewalls and other systems.
- Integrated Bridge Controller (IBC) has high exposure to the Internet as compared to the Engine Controller (EC) and computer system managing ballast tank.
- Captain and crew members can log into the CyberShip systems such as `Engine controller`, IBC, and computer system managing ballast tank to perform the actions such as start, stop, increase or decrease speed, reroute, etc. Moreover, captain can log into the system to change the system parameters.
- In the analysis, we considered external and internal adversaries.
- The risk analysis is carried out by considering the likelihood of the threat and its impact. For the STRIDE analysis, we used the criteria defined in Tables 5 and 6.

Table 3
Unsafe control actions for start pump action from engine controller (EC) to ballast tank pump.

Control action	Performed with hazard	Not performed with hazard	Performed too long or too short with hazard	Performed too early or too late with hazard
CA1: Start Pump	**UCA1.1**: when EC has provided wrong parameter (Velocity, Level) to Pump.			
UCA1.2: when EC receives the wrong parameters from IBC				
UCA1.3: when Ballast tank Pump is not functioning.				
UCA1.4: when Due to network failure control action is not received by Ballast tank.				
UCA1.5: when EC is compromised because of human in the loop such as unintentionally connecting infected USB in the CyberShip system.				
UCA1.6: when EC is compromised because of component failure.				
UCA1.7: when EC is compromised because of external hacker.				
UCA1.8: when it was not required.	**UCA1.9**: when EC is compromised because of human in the loop such as unintentionally connecting infected USB in the CyberShip system.			
UCA1.10: when EC is compromised because of component failure.				
UCA1.11: when EC is compromised because of external hacker.				
UCA1.12: when EC did not receive command from IBC.	**UCA1.13**: when `requirement` was for a shorter period and the pump acted for too long.			
UCA1.14: when `requirement` was for a longer period and the pump acted for too short.	**UCA1.15**: when there are communication channel congestion.			
UCA1.16: when there is a feedback delay between Actuator to Ballast tank.				
UCA1.17: when EC action was performed too early or too late.				

Table 4
Requirement and constraint examples.

Constraints	
C1	`start pump` action must not be provided if the water level information is not received from `Integrated Bridge Controller`
C2	Parameters communicated for `action` needed before execution
C3	User interface limited to required `actions`
C4	`Action` requirement confirmation must be defined and included
C5	A receipt confirmation must be sent of required `actions`
Requirements	
R1	A risk boundary should be set to avoid the increase or decrease of water level in the ballast tank to a dangerous level.
R2	A risk boundary should be set to define and confirm channel integrity.

Table 5
Impact criteria.

High	1. Loss of propulsion and endangering life of crew members.
	2. It can damage the system.
	3. It can result in the financial and customer loss.
	4. It can cause system malfunction.
	5. It can affect the availability of the system.
Medium	1. It can impact the integrity of the system.
	2. It can cause information disclosure.
	3. It can cause procedure disruption in real time.
Low	1. It can cause the operational delay in non-critical procedures.
	2. It may cause information disclosure of non-sensitive data.

Tables 7 –9 highlights STRIDE threats. As we can see in Table 7, spoofing, tampering and denial of service are the most critical threats for the engine controller that can directly damage the CyberShip. In the analysis we assume that the engine controller has limited exposure to the Internet. If engine controller is spoofed by an adversary, then a command that is issued from it can be executed. For example, if engine controller issues control commands (e.g., start or stop pump), then ballast tank can start or stop pumping water. Moreover, if the water-level feedback information from ballast tank to the engine controller is tampered in this process by adversary, then water can increase to a dangerous level in the tank, which can imbalance the ship. Denial of Service (DoS) attack on the engine controller can make it unavailable for operation, which can damage the CyberShip system, as it cannot perform required control actions.

Table 8 shows the STRIDE threat to ballast tank. It is seen that resultant risks are not very high in the case of ballast tank because it is not connected to the Internet and there are multiple layers of systems above it. So, the likelihood of direct attack is low in the case of ballast tank. Moreover, out of six STRIDE threats, four can have high impact on the system. Those four threats are spoofing, tampering, Denial of Service (DoS) and Elevation of Privilege (EoP). For example, tampering of feedback information from ballast tank to engine controller can damage the system, as based on this information, engine controller can issue commands to either increase or decrease the water level in the ballast tank. Moreover, computer system managing ballast tank can be spoofed and it can provide wrong information such as water level feedback to the engine controller that can damage the CyberShip. Denial of Service can happen on the computer system of ballast tank from the engine controller side, once the engine controller is compromised. It can also

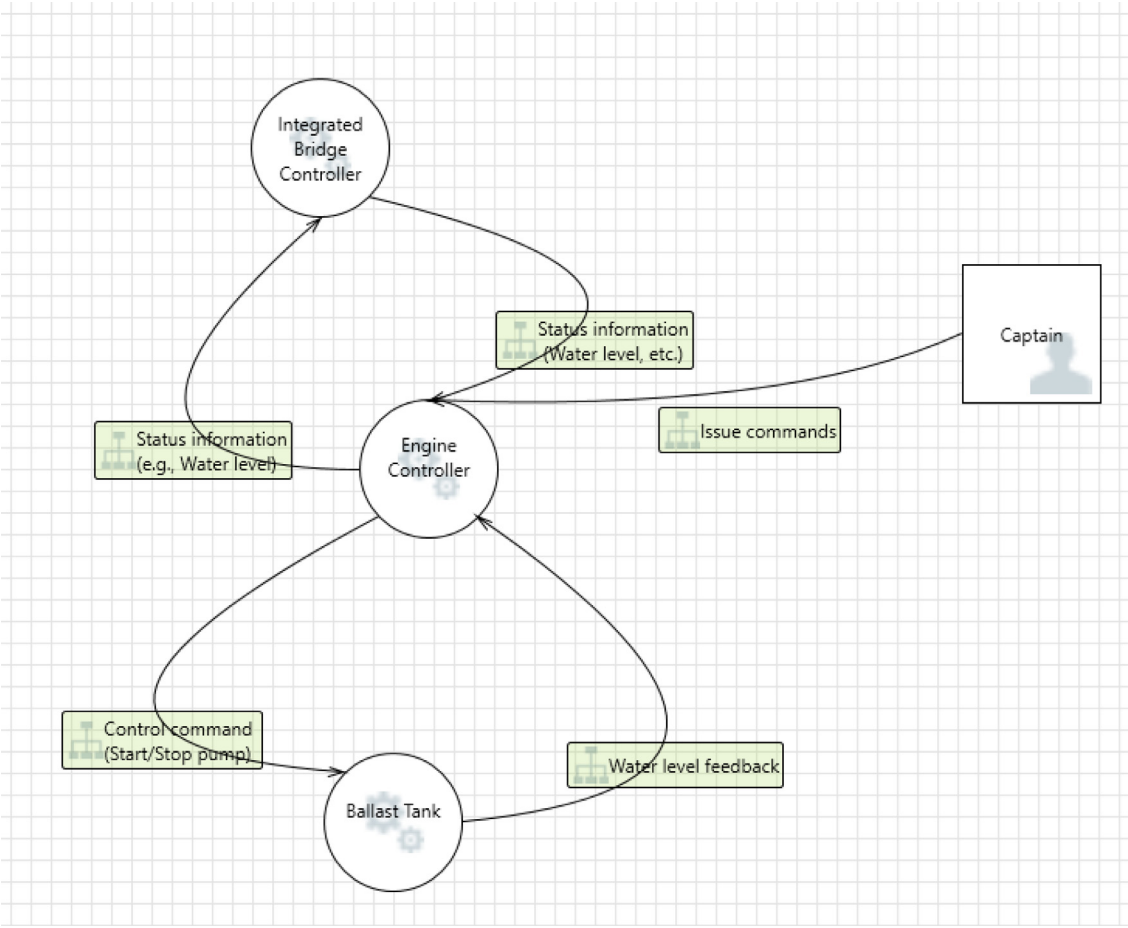

Fig. 3. Asset diagram for CyberShip.

Table 6
Likelihood criteria.

Very Likely	1. The adversary is highly motivated and capable, and there are no deployed countermeasures. 2. Existing popular exploits which can be executed remotely. 3. System is directly exposed to the Internet. 4. It can damage the system and provide benefit to attacker. 5. Up to once in 6 months.
Medium	1.The adversary is highly motivated and capable, while the system countermeasures are not enough to prevent the attack. 2. The system's vulnerability is widely known, but it requires physical presence to launch the attack. to exploit the vulnerability in the system. 3. Systems are not directly exposed to the Internet. 4. It can damage the system and provide some benefit to attacker. 5. Multiple layers of systems need to be compromised. 6. Once in 6 months up to once in 1 year.
Low	1. The attacker is not highly motivated or does not have the necessary knowledge to perform an attack, or deployed countermeasures are sufficient. 2. It requires administrative rights to launch the attack. 3. The system is not connected with external networks or systems. It requires physical presence. 4. It can not damage the system and provide much benefit to attacker. 5. Once in 2 years or less.

happen if the service engineer goes for some maintenance work and plugs the infected USB in the computer system of the ballast tank. The virus and malware spreading through the infected USB can cause cyber attack on the computer system managing the ballast tank. However, as computer system managing ballast tank is not connected to the Internet, it will not have much impact. While if mis-configuration happens, and these systems get connected to the Internet, then it can damage the CyberShip system. Similarly, elevation of privilege can only happen if the attacker is physically present or due to some mis-configuration, it gets connected to the Internet and malware or viruses are present in the computer system of the ballast tank.

Table 9 highlights the STRIDE threats to the IBC. Spoofing, Tampering, Denial of Service, and Elevation of Privilege are the most critical for the IBC, which can impact the operations of the CyberShip. For instance, if the information coming from the shore center is spoofed and tampered by an adversary, then it can impact the operation of the CyberShip. Moreover, in case navigational details provided by the IBC to the engine controller is tampered, it can damage the CyberShip system. Because of the wrong informa-

Table 7
Engine controller.

Threat (T)	Engine controller	I	L	R
Spoofing (S)	Engine controller can be spoofed to the ballast tank pump and to the IBC by external adversary because of malware or phishing attack. However, it requires physical presence or multiple layers of systems need to be compromised. Internal adversaries can acquire the credentials of captain to issue a control command. However, likelihood of spoofing from internal adversary is low because there are cameras in the engine control room where the critical control systems are installed. Crew members can unintentionally connect an infected USB that can cause external adversary to spoof.	H	M	H
Tampering (T)	Control command from the EC to the ballast tank pump can be tampered. It can cause damage to the ship. But, physical presence of external or internal adversary is needed or multiple layers of systems and firewalls should be compromised for the tampering it remotely. Crew members can change some parameters that can damage the system.	H	M	H
Repudiation (R)	EC can claim that it did not perform the command or received the data. It can impact the root cause analysis of an incident. Internal adversary can perform malicious action using login details of captain and deny performing the action. However, due to logging of events and presence of cameras reduce the likelihood of threat from internal adversary.	L	L	L
Information Disclosure (I)	It will not negatively impact the operation of the ship. Moreover, it will not provide much benefit to attackers.	L	L	L
Denial of Service (D)	The availability of the controller is very important. It can cause human safety issue. By deploying malware in the CyberShip system adversaries can launch DoS attack on the controllers. External adversary can use phishing emails to deploy malicious payload. However, internal adversary can use USB stick to deploy malicious payload in the CyberShip systems intentionally or unintentionally.	H	M	H
Elevation of Privilege (E)	If adversaries get administrative rights they can execute commands which can damage the ship. However, for external adversary it is difficult since the engine controller has limited exposure exposed to the Internet, and they need to deploy malicious payload somehow.	H	L	M

Table 8
Ballast tank.

Threats	Ballast tank pump	I	L	R
Spoofing (S)	Computer system managing ballast tank can be spoofed by external adversary because of malware or phishing attack. But, it requires multiple systems such as engine controller, Integrated Bridge Controller and firewalls to be compromised. Internal adversaries can acquire the credentials of captain to issue a control command. However, likelihood of spoofing from internal adversary is also low because there are cameras in the engine control room where the critical control systems are installed. Moreover, if crew members unintentionally connect an infected USB then also malware can spread and cause spoofing.	H	L	M
Tampering (T)	Tampering with this system and data in transit can cause critical damage and imbalance the ship. But, it is difficult to tamper with the hardwired signals in ballast tank. It requires physical presence. For internal adversary also it will be difficult to tamper because it requires high skills and because of cameras it will be difficult to tamper with.	H	L	M
Repudiation (R)	It can claim that it did not perform the command or received the data. But, it is very difficult as every action happens due to some events. It can impact the root cause analysis of an incident. Internal adversary can perform malicious action using login details of captain and deny performing the action. However, due to logging of events and presence of cameras reduce the likelihood of threat from internal adversary.	L	L	L
Information Disclosure (I)	Disclosure of information will not negatively impact the operation of the ship. Moreover, it will not provide much benefit to attackers.	L	L	L
Denial of Service (D)	The availability of the system is very important. It can impact the operation of the ship. By deploying malware in the CyberShip system, adversaries can launch DoS attack on the controllers. External adversary can use phishing emails to deploy malicious payload. However, internal adversary can use USB stick to deploy malicious payload in the CyberShip systems intentionally or unintentionally.	H	L	M
Elevation of Privilege (E)	If attackers get administrative rights they can issue commands which can cause imbalance and damage to ship. But, it is difficult for external adversaries to get the administrative privileges as it is not directly exposed to the Internet.	H	L	M

Table 9
Integrated bridge controller

Threats	Integrated bridge controller (IBC)	I	L	R
Spoofing (S)	Spoofing can cause damage to the system and to crew members. It can be due to phishing attack or through use of infected USB by crew members. The IBC's exposure to the Internet is high. Internal adversaries can acquire the credentials of captain to issue a control command. However, likelihood of spoofing from internal adversary is also low because cameras are installed where critical control systems are deployed. Crew members can unintentionally connect an infected USB that can cause external adversary to spoof.	H	M	H
Tampering (T)	Tampering with this system and data in transit can cause operational problem to other connected components and damage the ship. For external adversaries some backdoor access is required for tampering with these systems and parameters. Crew members can change some parameters that can damage the system.	H	M	H
Repudiation (R)	It can claim that it did not send the information or received the data. But, it is very difficult as every action happens due to some events. It can impact the root cause analysis of an incident. Internal adversary can perform malicious action using login details of captain and deny performing the action. However, due to logging of events and presence of cameras reduce the likelihood of threat from internal adversary.	H	L	M
Information Disclosure (I)	Disclosure of information will not negatively impact the operation of the ship. Moreover, it will not provide much benefit to attackers.	L	L	L
Denial of Service (D)	The availability of the system is very important. It can negatively impact the operation of the ship. By deploying malware in the CyberShip system, adversaries can launch DoS attack on the controllers. External adversary can use phishing emails to deploy malicious payload. However, internal adversary can use USB stick to deploy malicious payload in the CyberShip systems intentionally or unintentionally.	H	M	H
Elevation of Privilege (E)	If attackers get administrative rights they can forward wrong information which can cause operational problem to ship such as delay in arrival. But, it is difficult to get the administrative privileges for external adversaries as they need to deploy malicious. payload that can provide admin rights.	H	M	H

tion, CyberShip can be rerouted or water level in the ballast tank can be increased or decreased to damage the CyberShip. To prevent this, we need to encrypt information in-transit and implement input validation at the engine controller and ballast tank to prevent the damage.

6. CyberShip analysis using CORAS

CORAS is a model based risk analysis methodology (Lund et al., 2010). It uses Unified Modelling Language (UML) for threat and risk modelling. The UML is used to model the system under consideration and the context. It also offers a tool to support documenting, maintaining and reporting risk analysis results. CORAS is not only focused on identifying security requirements but also oriented towards performing holistic risk assessment. The CORAS method is comprised of eight steps. The authors in Lund et al. (2010) have described all the detailed stages of CORAS methodology in identifying the risks. For brevity purpose, here the main focus is on identifying threats and how they can impact the availability of the CyberShip framework.

The first step in the CORAS methodology is to identify System under Consideration (SuC) or assets. An asset could be anything such as hardware, software, people, reputation, etc. Figure 4 demonstrates the SuC for the threat identification and risk assessment. It also shows how the different assets or components interact with each other on a high-level. Three main assets or components that are selected for the risk assessment are: Integrated Bridge Controller, Engine Controller, and Ballast tank. The CyberShip availability in Fig. 4 is directly impacted by the compromise of these assets. Reputation of the organization or financial situation is indirectly affected due to the impact on the availability of the CyberShip system.

The main step after asset identification is risk identification using threat diagrams. Threat diagrams represent possible scenarios that must be considered threats. The purpose of this stage is to identify threats for the CyberShip framework. We have considered threats because of adversary/hacker (external), crew (internal), and component failure. Generally, threats trying to exploit the vulnerabilities in the CyberShip can harm the system. In the analysis, we mainly considered threats initiated by adversary (deliberately), crew (accidental), and component failure.

Before going into the CORAS risk assessment, we highlight some assumptions that are necessary for the analysis, as follows:

- We assume that there are not enough countermeasures to protect the system.
- Crew members are not well aware of cyber security issues. During interactions with the system, they can spread malicious code or modify some parameters that can damage the CyberShip system unintentionally.
- In the CORAS assessment, we will also consider the unwanted incidents because of component failure.
- To begin, CORAS first requires that the threats on individual components should be identified, and then it can be used to analyse how it can compromise and damage the whole system.

As we can see in Fig. 5, unwanted incidents that can damage the ballast tank is due to the wrong water level, which can imbalance the ship. Changes in system parameters, Denial of Service (DoS), and component or system malfunction can cause the wrong water level, which can damage in the ballast tank. For example, because of the DoS attack, ballast tank might not receive the com-

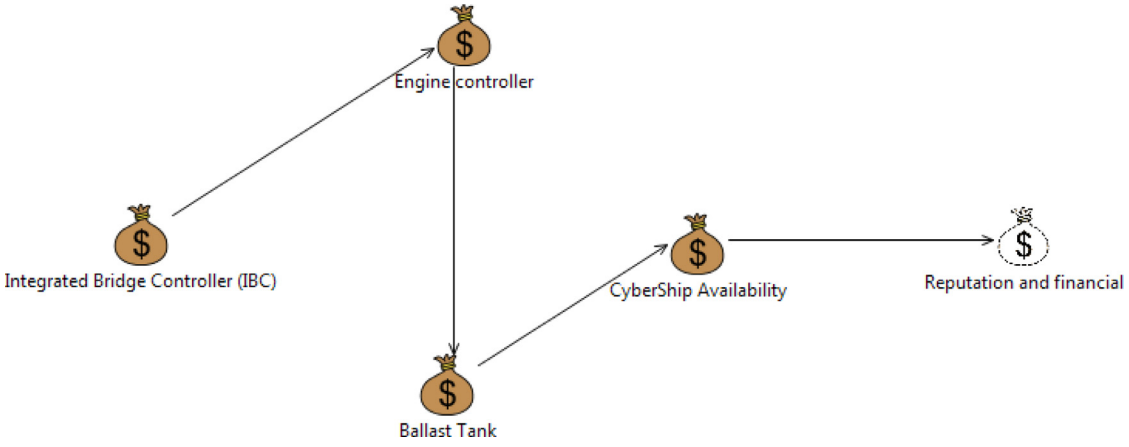

Fig. 4. Asset diagram for CyberShip.

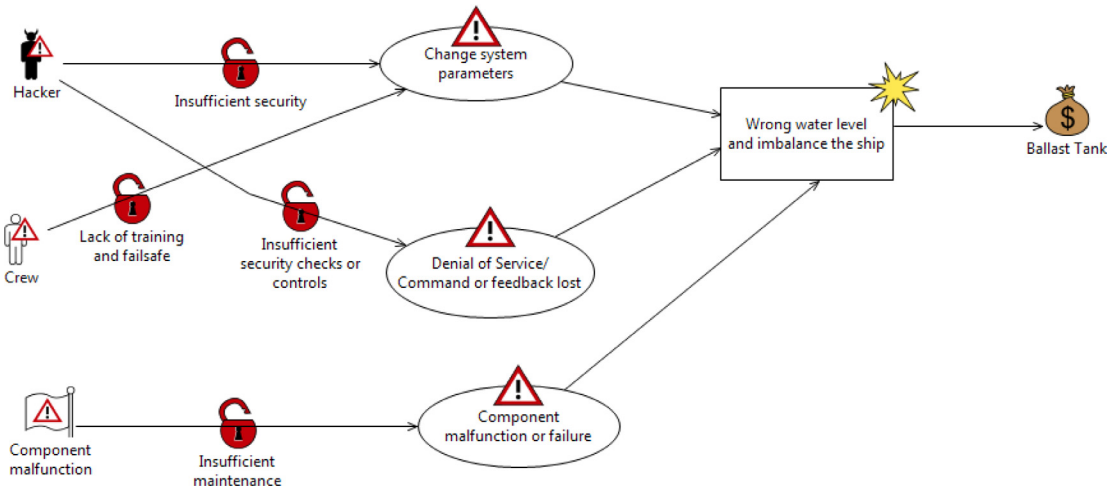

Fig. 5. Threat diagram for ballast control system.

mand from the engine controller or the feedback from the ballast tank to the engine controller might be lost that can lead to wrong water level and imbalance the ship. Moreover, the lack of proper maintenance can cause system malfunction or component failure due to that ballast tank cannot receive or operate on the commands, which can also lead to unwanted incidents.

As shown in Fig. 6, threat scenarios comprising of wrong system parameters and component failure can lead to the compromise of integrity and availability of the system, which can damage the engine controller. Adversary can get the advantage of the lack of sufficient security measures and input validation to provide wrong system parameters to damage the engine controller. For instance, adversary can input wrong parameters so the engine can consume more fuel or get heated up, which can damage the engine. Parameters can be changed to a wrong value by insufficiently trained crew members and service engineers, which can also damage the engine. Wrong parameters can cause improper functioning of the engine controller that can damage it. Moreover, the lack of proper maintenance can also cause the component failure, which can cause safety and security issues.

Figure 7 shows the threat scenarios that can lead to compromise of the Integrated Bridge Controller. Many bridge devices (e.g., AIS, ECDIS) are vulnerable to cyber attacks because of lack of authenticity and integrity checks. For example, due to the compromise of ECDIS, IBC will display wrong route information. Because of the lack of application whitelisting and anti-virus protection, ECDIS is vulnerable to malware attacks. It can be exploited by adversaries to feed wrong information to the IBC, which can result in displaying wrong information. All these can result in the compromise of integrity and availability of the IBC.

Figure 8 illustrates different threat scenarios and likelihood that can damage the CyberShip. Here, we assume that there is not much security has been implemented. More specifically, we expect that it is an initial level risk assessment to identify different threat scenarios and unwanted incidents, which can damage CyberShip. So, in most cases the likelihood of threat scenarios are assigned as “medium”. It can be seen in the Fig. 8 that wrong or fake parameters can lead to a threat scenario of displaying wrong information. As a result of this, unwanted incidents like reroute of the ship, increase/decrease of speed, and start or stop of propulsion control can occur. For instance, wrong or fake parameters provided to the engine controller can result in the increase or decrease of speed of the ship. Similarly, IBC can display wrong information because of malicious code, which can result in rerouting of the ship. Denial of Service (DoS) attack can lead to the loss of control commands and feedback messages to damage the CyberShip. All these unwanted incidents can damage the CyberShip and lead to the compromise of its availability. Incidents such as reroute and collision of the ship are assigned “low” likelihood because crew or captain of the ship can keep monitoring the ship and surrounding through the window. Moreover, these incidents can be easily noticed by crew members. Incidents such as “increase or decrease of speed” and “increase/decrease of water level in ballast” can get unnoticed for sometimes, thus “medium” likelihood is assigned.

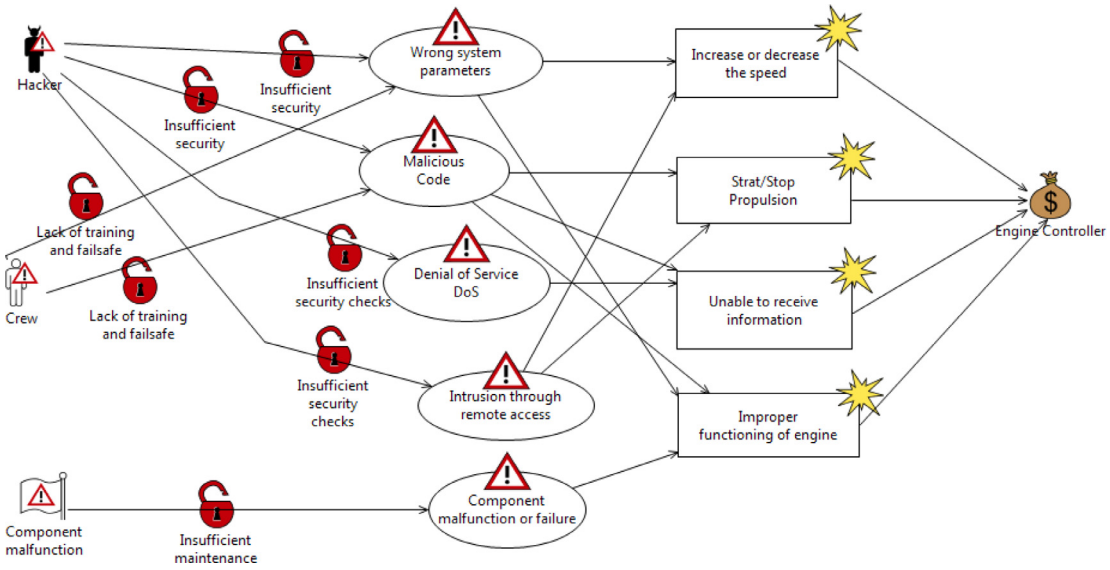

Fig. 6. Threat diagram of engine controller.

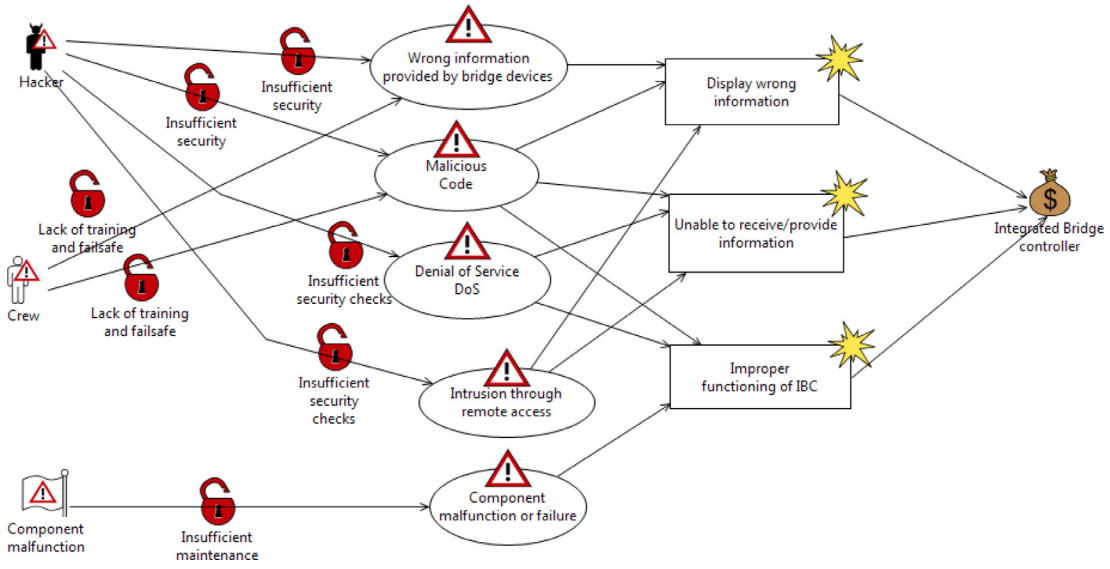

Fig. 7. Threat diagram for integrated bridge controller.

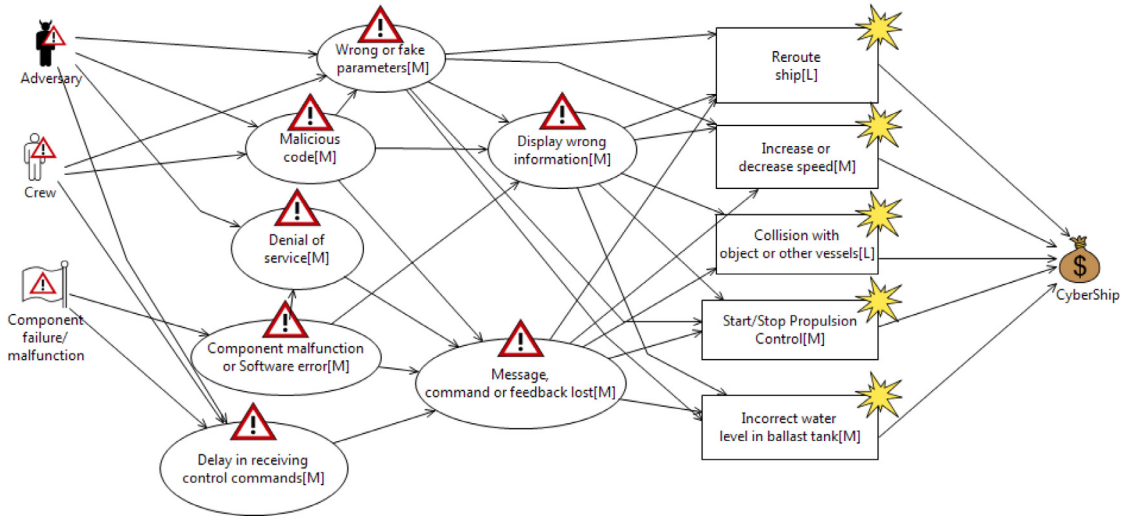

Fig. 8. Threat diagram for CyberShip.

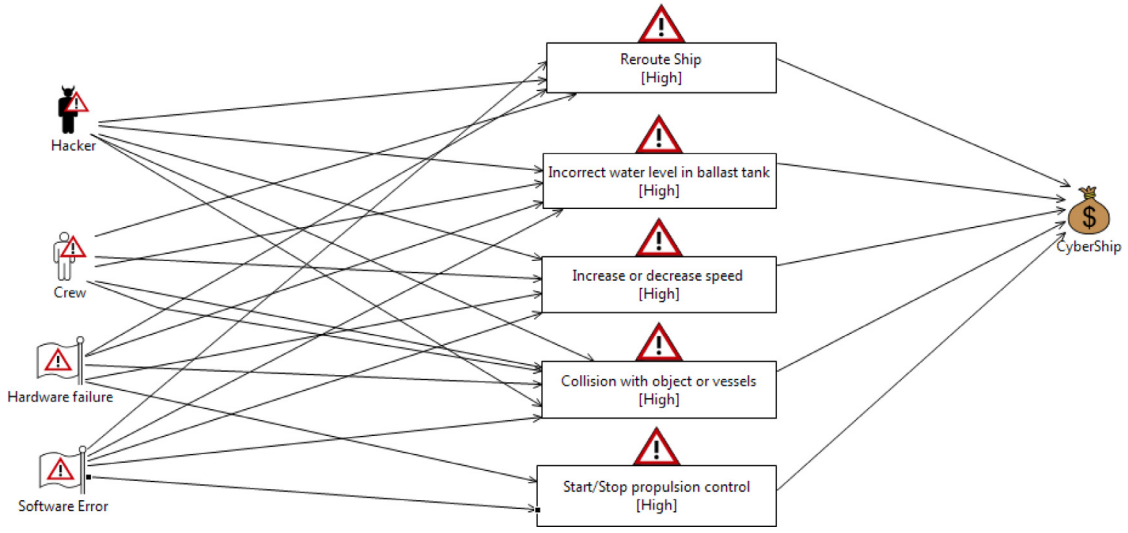

Fig. 9. Risk diagram for propagating risk in CyberShip.

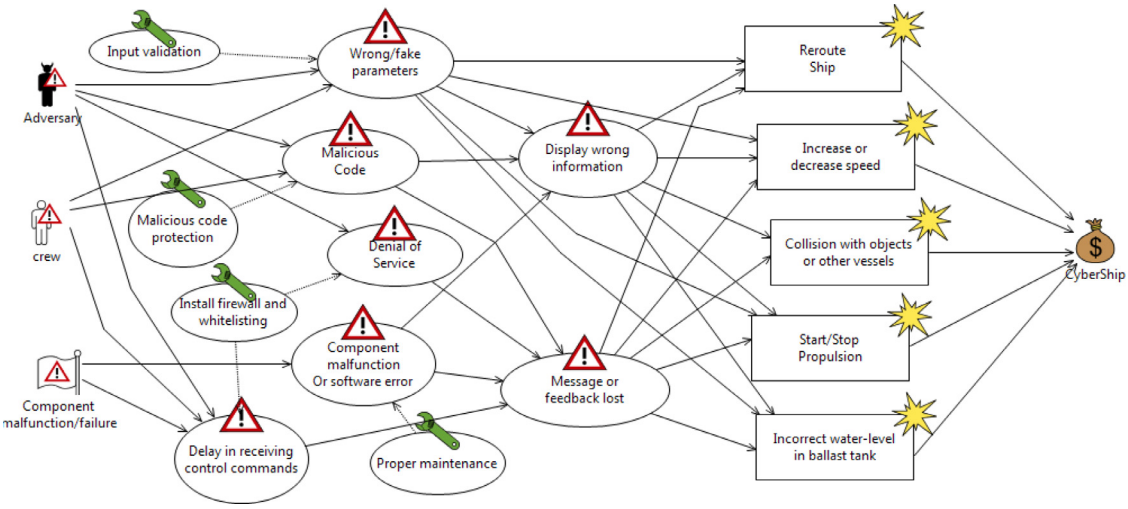

Fig. 10. Threat mitigation diagram for CyberShip.

Start/Stop propulsion control incident can occur quickly if engine controller is compromised, hence it is assigned the likelihood of "medium".

Figure 9 shows different risk scenarios impacting the CyberShip, which can result in the compromise of its availability. Wrong water-level in the ballast tank can imbalance the ship, which can result in accidents. Similarly, the increase or decrease of speed because of wrong parameters can damage the CyberShip's availability. Since, it can cause congestion on the particular route and delay the arrival of the ship at the port.

Figure 10 illustrates threat mitigation methods that can be deployed to mitigate the threats and protect the CyberShip system. The threat of wrong or fake parameters can be minimized with the deployment of `input validation`. This will prevent the change of the parameters in the controllers to beyond a threshold value. Denial of Service (DoS) threat can be mitigated with whitelisting and the deployment of proper firewall rules. Because of brevity purpose, only few mitigation mechanisms are shown in Fig. 10. It is noted that other mitigation mechanisms can be deployed as well. For example, to mitigate the threat of malicious code, authenticity and integrity checks can also be deployed along with malicious code protection. To protect CyberShip from component malfunction or software error, fail-safe and error handling should also be in place along with timely and proper maintenance. Moreover, if a system fails, it should not disclose any sensitive information.

It is seen that CORAS helps in identifying the threats, vulnerabilities and risks in a structured way through threat and risk diagrams. For example, it helps in visualizing threats and risks, and analysing the way it can propagate and damage the system. In the next section, we compare three risk assessment methodologies of STRIDE, STPA and CORAS.

7. Analysis comparison between STPA, STRIDE and CORAS

Risk assessment methodologies have some common stages in identifying and estimating threats and risks. These common stages in risk and threat assessment methodologies are:

- **Establish the System under Consideration and context:** It is important to identify system under consideration, scope, likelihood and impact criteria before delving into the risk assessment.
- **Model of the System:** An explicit model of the system helps in identifying the threats, vulnerabilities, impact and resultant risk without getting bogged down in too many details. There are many different types of diagrams that are helpful to model the system. For instance, data flow diagrams (DFDs), UML, and

Table 10
Comparison between STPA-Sec, STRIDE and CORAS methodologies.

Stages	STPA-Sec	STRIDE	CORAS
System Under Consideration and Context	System objective: Identify the goal for which the system is designed and system boundaries	System is identified on which analysis has to be performed.	Asset identification is the first step in CORAS.
Model of the System	Identify System Structure: list the controls, process models, processes and operators and their connections, including hierarchy.	System model is created using DFD.	Model of the system is created using threat diagram.
Identify Risks	Identify Hazards: list system state or conditions that with environmental worst-case scenario, lead to an unacceptable loss Identify requirements and constraints: list the controls by presence (passive) and the controls by action (active) through detection, measurement, diagnosis or response.	Identify threats: spoofing, tampering, repudiation, information disclosure, elevation of privilege, DoS are identified.	Threats, vulnerabilities and unwanted incidents are identified using threat diagrams.
Risk Estimation	Describe UCA: list conditions when each control action or their lack creates a hazard. List hazards through degradation over time.	Determine likelihood and impact in a brainstorming session.	Risk diagram helps in identifying and estimating risks.
Risk Evaluation	Evaluate UCA: list causal scenarios and additional constraints from the UCA analysis	Tolerable and unacceptable risks are identified which need further investigation.	Tolerable and unacceptable risks can be identified through risk diagram.
Risk Mitigation	List the additional design requirements to implement the additional constraints	List countermeasures to mitigate the threats.	Through threat mitigation diagram countermeasures are shown to mitigate threats and vulnerabilities.

state diagrams are frequently used to design the model of the system. The main purpose of these diagrams is to present the high-level architecture of the system along with the information flow, so that all the stakeholders involved in the risk assessment can have the same understanding. Among these diagrams, DFDs should be the most frequently used diagrams.

- **Identify Threats or Risks:** It is very important to identify what aspects of the system can go wrong and what are the possible threats and hazards. There are many methods to find threats and hazards in the CyberShip system. Generally, these are identified by organizing a workshop with the system's expert.
- **Risk Estimation:** Risk estimation is to understand the nature of threat or risk and to find out the likelihood.
- **Risk Evaluation:** In this stage, risk estimation results are mainly compared with the pre-defined criteria to determine whether the risk is acceptable or tolerable.
- **Risk Mitigation:** This stage is about planning and implementing countermeasures to mitigate or minimize the impact of risk.

7.1. Analysis of STPA, STRIDE and CORAS

In this section, a comparison between STPA, STRIDE and CORAS is presented. The aim is to highlight the key features of these methodologies and analyze the context in which they can perform better than the others. Table 10 highlights some key stages of STPA-Sec, STRIDE and CORAS methodologies.

The detailed control structure of the system is established for the hazard analysis using STPA-Sec methodology. Table 3 shows unsafe control actions or events that could cause hazardous state. As we can see, 17 unsafe control actions or commands have been identified in the framework using STPA-Sec. It is evident that one hazard can be triggered by more than one unsafe control action. For example, uncontrolled maneuvering of the ship can be caused by **UCA1.1,UCA1.2, UCA1.11, UCA1.12**.

Hazards identified through STPA-Sec method can be put into the following five categories, as suggested by Leveson (2011):

- Component failure
- Component interaction
- Software fault
- Human error
- System error

These five categories cover all the system components. Nowadays, cyber-physical-systems (CPS) are comprised of software. hardware, human and interacting components. Therefore, these five categories cover all the components of the CyberShip framework. It should be noted that the STPA-Sec method identifies unsecure control actions along with problems caused by design errors, software faults, component interaction, human in the loop, and cyber attacks. Moreover, it also identifies security constraints that can be enforced to prevent system from entering into vulnerable state, which leads to damage. The main aim of STPA-Sec is to identify unsafe and unsecure control actions that can cause dam-

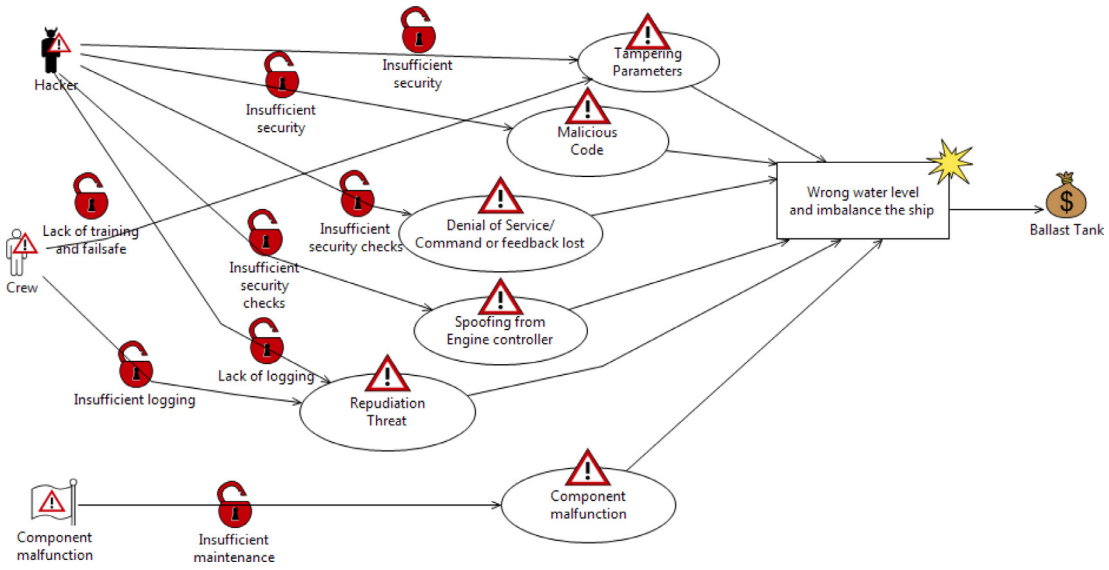

Fig. 11. Threat diagram of ballast tank considering STRIDE threats.

age to the system. More specifically, it identifies hazards or conditions that can lead the control actions to unsafe or unsecure state, which can compromise or damage the system. However, many well known cyber threats and vulnerabilities might not be covered by this. For example, as shown in Table 3, unsafe control actions such as **UCA1.1, UCA1.2, UCA1.4** and **UCA1.5** can occur because `Engine Controller` is compromised. However, threats and vulnerabilities that can compromise the `Engine controller` cannot be identified at this stage. A detailed threat modelling or cyber risk analysis at the component level is required in identifying threats and vulnerabilities that can lead to the compromise of `Engine controller`. It is important to include security-related accidents into the definition of accidents while performing STPA-Sec that uses a set of causal factors for security. However, these causal factors focus mainly on integrity and availability, confidentiality is not considered. Moreover, security related losses and hazards that can cause these losses must be considered in detail by integrating security requirements in the STPA-Sec analysis. Furthermore, STPA-Sec does not employ a threat model to consider new causal factors.

It is clear from the STRIDE analysis that it offers a systematic approach to identify cyber threats against each component of the system. It analyzes cyber threats corresponding to security properties such as authentication, authorization, system integrity, confidentiality, repudiation and availability. As shown in the analysis of STRIDE, the impact, the likelihood and the resultant risks are described in terms of qualitative values and very much subjective. As we can see, the STRIDE analysis helps in identifying different types of cyber threats. The STRIDE methodology is very good in enumerating the list of cyber threats that can compromise the system. However, it is important to identify safety as well as security threats in the cyber-physical systems (CPS) such as CyberShip. Therefore, cyber threats identified through STRIDE mechanism can be used as an input to other mechanisms such as CORAS and STPA-Sec. It can assist in identifying more threats in a structured way.

CORAS is also a top-down and asset-based approach of identifying risks. It relies on asset, threat, risk and treatment diagrams to identify threats, vulnerabilities, and risks. This method is very helpful in visualizing the way how threats can propagate and lead to unwanted incidents that can damage the system. Generally, CORAS method identifies threats in a brainstorming session. As shown in the CORAS analysis, it can also identify safety threats and risks such as component malfunction or software error and the way how these threats can lead to incidents that can damage the system. Moreover, it helps in identifying the nature of vulnerabilities that can cause threats as compared to STPA-Sec and STRIDE approach. However, CORAS lacks the structured process of finding out threats that can compromise the system. It relies on the system expert to identify threats and vulnerabilities and depends on experience. However, threats identified through the STRIDE method can be used as input while performing CORAS analysis, it can make CORAS more structured and will help in identifying more threats and vulnerabilities that are not easy to identify.

8. Discussion

The methodology comparison process involves the systematic collection of data and critical analysis, and thus the findings are based on evidence rather than opinions or personal biases. This objectivity leads to more accurate and reliable results that are replicable by other researchers. Comparing risk analysis methodologies is a scientific contribution that provides a systematic and objective approach to the analysis of risk. The findings of these comparisons can provide new insights, improve decision-making, and lead to the development of more effective and efficient risk analysis methodologies.

STPA-Sec uncovers more hazardous situations at the design level. Also, by focusing the analysis on the system structure, STPA-Sec analysis results in design recommendations that can secure shipping system against cyber attacks, which are independent of the source of the attacks. The main challenge with STPA-Sec is finding a method to identify cyber threats both at the component level and from mis-interactions between the components, which is consistent, thorough and not overly dependent on experience. STPA-Sec, on the other hand, can be extended with the STRIDE threats to identify new loss scenarios and requirements due to a lack of security.

CORAS offers a model-based framework for identifying threats, vulnerabilities and risks. Threat and risk diagrams are helpful in visualizing the way how threats and risks can compromise and damage the CyberShip. Countermeasures to mitigate threats are visualized through the threat mitigation diagram, which can help in eliciting security requirements for the system under consideration. Moreover, safety risks can also be identified using CORAS approach. However, the challenge in CORAS methodology is to find or enumerate different threats that can exploit vulnerabilities, which can damage the system. Because of the complexity of the system, it

is very common to miss threat scenarios. However, STRIDE threats can be considered during the CORAS based risk analysis. Moreover, these threats can be mapped to confidentiality, integrity and availability features, i.e., threats that can compromise the confidentiality, availability and integrity of the system. It can assist in performing a risk assessment in a structured way by considering the main objectives of the system. For instance, availability is the major concern in the case of CyberShip.

STRIDE approach is very good at enumerating threat lists. The disadvantage with STRIDE model is that it does not consider safety risks, which are a major concern in the cyber-physical systems such as CyberShip. STRIDE method is very good for identifying computer security threats; however, it does not consider the physical aspects, which are very important in cyber-physical systems. However, the list of threats found using STRIDE can be used in CORAS and STPA-Sec analysis to perform a detailed risk assessment. Furthermore, the identification of STRIDE threats depends on the experience of the person facilitating the risk assessment. For instance, it is not easy to directly identify how spoofing can happen in the case of engine controller. Below we use an example of using STRIDE threats as an input in the threat diagram of CORAS framework.

As we can see in Fig. 11, STRIDE threats such as tampering, DoS, spoofing and repudiation are all targeting ballast tank, which can lead to unwanted incidents such as wrong water level in the ballast tank to imbalance the ship. An adversary can spoof as an engine controller and issue control commands to either increase or decrease the water level in the ballast tank to a wrong level. It can happen because of insufficient security checks, lack of proper firewall rules & segmentation, lack of input validation, and whitelisting application, etc. In this way, by using STRIDE threats during STPA-Sec and CORAS analysis, we can identify more threat scenarios in a structured way.

However, all these three methodologies do not consider exploitability while performing risk assessment. Exploitability specifies how easy it is to target the particular component in the system. Moreover, during risk assessment, it is also recommended to consider the level of attackers, i.e., whether they are script kiddie or skilled attackers, which has not been considered during STPA-Sec, STRIDE and CORAS assessment. It depends on the facilitator to include these criteria during risk assessment.

9. Conclusion and future work

In this article, we applied STPA-Sec, STRIDE and CORAS methodologies on CyberShip framework to perform risk assessment. By applying these three methodologies, we identified the threats and hazard scenarios along with security requirements for the CyberShip framework. In the use case, we focused the analysis on safety and security; but other concerns such as privacy can be considered. The analysis helped in identifying the pros and cons of these methodologies when applied in the same framework. We found that STPA-Sec is good at identifying hazards by looking at the control actions and the structure of the CyberShip system.

CORAS provided a good framework for visualizing and identifying threats and unwanted incident scenarios. However, it relied on the system expert in identifying that, which would be difficult for somebody who is not an expert in those systems. STRIDE offers a structured approach for highlighting threats to the system, which can be coupled with CORAS and STPA-Sec and make them more effective.

Also, STRIDE could benefit from the control structure provided by STPA-Sec and threat mitigation diagrams provided by CORAS in identifying risks. We also extended CORAS method with STRIDE approach to identify more threats in CyberShip frameworks. As a future work, we intend to extend these methodologies in detail to identify the threats, risks and security requirements of CyberShip. Moreover, we will also study other methods such as PASTA, OCTAVE, DREAD, VAST and TRIKE for identifying threats and risks in CyberShip framework and perform a comparative assessment (Luo et al., 2021; Sivula, 2015).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Rishikesh Sahay: Conceptualization, Methodology, Software, Writing – original draft. **D.A. Sepulveda Estay:** Methodology, Software. **Weizhi Meng:** Methodology, Software, Supervision, Writing – review & editing. **Christian D. Jensen:** Resources, Writing – review & editing. **Michael Bruhn Barfod:** Resources, Writing – review & editing.

Data availability

No data was used for the research described in the article.

Acknowledgement

The authors would like to acknowledge the funding provided by the Orients Fund by the Danish Maritime Fund (DMF) to the project CyberShip - “Cyber resilience for the shipping industry” at the Technical University of Denmark, DTU, for the period 2017-2020.

References

Akpan, F., Bendiab, G., Shiaeles, S., Karamperidis, S., Michaloliakos, M., 2022. Cybersecurity challenges in the maritime sector. Network 2 (1), 123–138. doi:10.3390/network2010009. https://www.mdpi.com/2673-8732/2/1/9.

Autonomous Ships The Next Step. 2021.

Banda, O.A.V., Kannos, S., Goerlandt, F., van Gelder, P.H., Bergström, M., Kujala, P., 2019. A systemic hazard analysis and management process for the concept design phase of an autonomous vessel. Reliab. Eng. Syst. Saf. 191, 106584.

Capano, D. E., 2021. Throwback Attack: How NotPetya accidentally took down global shipping giant Maersk. https://www.industrialcybersecuritypulse.com/throwback-attack-how-notpetya-accidentally-took-down-global-shipping-giant-maersk/.

Chaal, M., Banda, O.A.V., Glomsrud, J.A., Basnet, S., Hirdaris, S., Kujala, P., 2020. A framework to model the STPA hierarchical control structure of an autonomous ship. Saf. Sci. 132, 104939.

Final Report: Autonomous Engine Room, 2015. Technical Report. MUNIN: Maritime Unmanned Navigation through Intelligence in Network.

Foussard, C., Denis-Remis, C.N., 2014. Risk assessment: methods on purpose? Int. J. Process Syst. Eng. 2 (4), 337–352. doi:10.1504/IJPSE.2014.070090. https://hal.archives-ouvertes.fr/hal-02305851.

Glomsrud, J.A., 2019. A structured STPA safety and security co-analysis framework for autonomous ships. In: Proceedings of the 29th European Safety and Reliability Conference (ESREL).

Grigoriadis, C., Laborde, R., Verdier, A., Kotzanikolaou, P., 2022. An adaptive, situation-based risk assessment and security enforcement framework for the maritime sector. Sensors 22 (1). doi:10.3390/s22010238. https://www.mdpi.com/1424-8220/22/1/238.

Guide for Cybersecurity Implementation for the Marine and Offshore Industries. 2021.

Hyra, B., 2019. Analyzing the Attack surface of ships.

IEC 62443-3-2:2020, Security for industrial automation and control systems - Part 3-2: Security risk assessment for system design. 2020. https://webstore.iec.ch/publication/30727.

Kaneko, T., Takahashi, Y., Okubo, T., Sasaki, R., 2018. Threat analysis using stride with stamp/STPA. The international workshop on evidence-based security and privacy in the wild.

Kavallieratos, G., Katsikas, S., Gkioulos, V., 2019. Cyber-attacks against the autonomous ship. In: Katsikas, S.K., Cuppens, F., Cuppens, N., Lambrinoudakis, C., Antón, A., Gritzalis, S., Mylopoulos, J., Kalloniatis, C. (Eds.), Computer Security. Springer International Publishing, Cham, pp. 20–36.

Kavallieratos, G., Katsikas, S., Gkioulos, V., 2020. Safesec tropos: joint security and safety requirements elicitation. Comput. Stand. Interfaces 70, 103429.

doi:10.1016/j.csi.2020.103429. https://www.sciencedirect.com/science/article/pii/S0920548919304982.
Kavallieratos, G., Katsikas, S., Gkioulos, V., Katsikas, S.K., Cuppens, F., Cuppens, N., Lambrinoudakis, C., Antón, A., Gritzalis, S., Mylopoulos, J., Kalloniatis, C., 2019. Cyber-attacks against the autonomous ship. In: Computer Security. Springer International Publishing, Cham, pp. 20–36.
Kayan, H., Nunes, M., Rana, O., Burnap, P., Perera, C., 2022. Cybersecurity of industrial cyber-physical systems: a review. ACM Comput. Surv. (CSUR) 54 (11s) 229:1-229:35.
Leveson, N., 2011. Engineering a Safer World: Systems Thinking Applied to Safety. MIT Press.
Lim, G.J., Cho, J., Bora, S., Biobaku, T., Parsaei, H., 2018. Models and computational algorithms for maritime risk analysis: a review. Ann. Oper. Res. 271 (2), 765–786.
Kohnfelder, L., Garg P., (April 1, 1999). *The threats to our products*. Microsoft Interface. Retrieved 13 April 2021.
Lund, M.S., Solhaug, B., Stlen, K., 2010. Model-Driven Risk Analysis: The CORAS Approach, first ed. Springer Publishing Company, Incorporated.
Luo, F., Jiang, Y., Zhang, Z., Ren, Y., Hou, S., 2021. Threat analysis and risk assessment for connected vehicles: a survey. Secur. Commun. Netw. 1263820. doi:10.1155/2021/1263820.
Omitola, T., Downes, J., Wills, G., Zwolinski, M., Butler, M., 2018. Securing navigation of unmanned maritime systems. In: Schillai, S.M., Townsend, N.C. (Eds.), Proceedings of the 11th International Robotic Sailing Conference: Southampton, United Kingdom, August 31st - September 1st, 2018. CEUR-WS, pp. 53–62. https://eprints.soton.ac.uk/430295/.
Process map for Autonomous Navigation, 2014. Technical Report. MUNIN:Maritime Unmanned Navigation through Intelligence in Network.
Puisa, R., Lin, L., Bolbot, V., Vassalos, D., 2018. Unravelling causal factors of maritime incidents and accidents. Saf. Sci. 110, 124–141.
Rokseth, B., Utne, I.B., Vinnem, J.E., 2018. Deriving verification objectives and scenarios for maritime systems using the systems-theoretic process analysis. Reliab. Eng. Syst. Saf. 169, 18–31.
Royce, R., 2016. Cyber security resilience management for ships and mobile offshore units in operation.
Sahay, R., Meng, W., Sepúlveda Estay, D., Jensen, C., Barfod, M., 2019. Cybership-IoT: a dynamic and adaptive SDN-based security policy enforcement framework for ships. Future Gener. Comput. Syst. 100, 736–750. doi:10.1016/j.future.2019.05.049.
Sepulveda Estay, D., Sahay, R., Barfod, M., Jensen, C., 2020. Exploring Cybership Vulnerabilities Through a Systems Theoretic Process Approach. https://ssrn.com/abstract=3753663.
7 Stages of Cyber Kill Chain Supplementary Reading. 2017. https://www2.deloitte.com/content/dam/Deloitte/sg/Documents/risk/sea-risk-cyber-101-july2017.pdf.
Shackelford, S., 2020. Cyber War and Peace: Toward Cyber Peace. Cambridge University Press. https://books.google.dk/books?id=dm_IDwAAQBAJ.
Sivula, A., 2015. Security risk and threat models for health care product development processes.
Tam, K., Jones, K., 2018. Cyber-risk assessment for autonomous ships. In: 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), pp. 1–8. doi:10.1109/CyberSecPODS.2018.8560690.
Tam, K., Jones, K., 2019. Macra: a model-based framework for maritime cyber-risk assessment. WMU J. Marit. Aff. 18 (1), 129–163.
Tehrani, R., 2017. NotPetya: World's First $10 Billion Malware.
The cyber threat against operational systems on ships. 2020. https://www.cfcs.dk/en/cybertruslen/threat-assessments/the-cyber-threat-against-operational-systems-on-ships/.
The Guidelines on Cyber Security Onboard Ships, 2017. Technical Report. BIMCO.
The Nippon Foundation MEGURI2040 Fully Autonomous Ship Program. 2022. https://www.nippon-foundation.or.jp/en/news/articles/2022/20220111-67000.html.
Wang, J., Sii, H., Yang, J., Pillay, A., Yu, D., Liu, J., Maistralis, E., Saajedi, A., 2004. Use of advances in technology for maritime risk assessment. Risk Anal. 24 (4), 1041–1063.
Wolff, J., 2021. How the NotPetya attack is reshaping cyber insurance. https://www.brookings.edu/techstream/how-the-notpetya-attack-is-reshaping-cyber-insurance/.
Wróbel, K., Montewka, J., Kujala, P., 2018. System-theoretic approach to safety of remotely-controlled merchant vessel. Ocean Eng. 152, 334–345.

Rishikesh Sahay is an Assistant Professor at Oregon Institute of Technology, USA. Before, he was a Postdoctoral researcher at Technical University of Denmark. He completed his PhD from Télécom SudParis and University of Pierre and Marie Curie (UPMC) in France. His Ph.D. thesis focused on autonomic cyber defense using software-defined networking. His research interests include autonomic cyber defense, policy-based network management, cyber resilience, software-defined networking, and network security.

Daniel Alberto Sepúlveda is now working at Digitalization Group, Rigshospitalet. Before, he was a researcher at the Department of Management Engineering with the project CyberShip, "Cyber-Resilience for the shipping industry". He finished my Ph.D. at DTU in 2018 in the topic of "Cyber risk and security in the global supply chain". He has over 10 years of experience working in the supply chain and operations departments of multinational companies. He is interested in understanding complex systems and decision making under uncertainty through Quantitative Analysis, System Dynamics and Real Options evaluation models.

Weizhi Meng is currently an Associate Professor in the Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), Denmark. He obtained his Ph.D. degree in Computer Science from the City University of Hong Kong (CityU), Hong Kong. Prior to joining DTU, he worked as a research scientist at the Institute for Infocomm Research, A*STAR, Singapore. His primary research interests are cyber security and intelligent technology in security including intrusion detection, smartphone security, biometric authentication, HCI security, cloud security, trust management, malware detection, blockchain in security, cyber-physical system security and IoT security. He is currently directing the SPTAGE Lab at DTU Compute, DTU, and received the IEEE MGA Young Professionals Achievement Award in 2020 for his contributions to leading activities in Denmark and Region 8.

Christian Damsgaard Jensen is an Associate Professor and the Head of the Cyber Security section at the Department of Applied Mathematics and Computer Science, Technical University of Denmark. He holds an M.Sc. in Computer Science from the University of Copenhagen and a Ph.D. in Computer Science from Université Joseph Fourier (Grenoble I, France). He held a position as Lecturer in Computer science at Trinity College Dublin from 1998 to 2002, where he was appointed to his current position. He conducts research in the area of security in distributed systems, where he is particularly interested in the development of models, policies and mechanisms that support secure collaboration in open distributed systems, such as pervasive computing, mobile computing and sensor networks. He has published more than 60 peer-reviewed papers in international journals, conferences and workshops.

Michael Bruhn Barfod is currently an Associate Professor at Technical University of Denmark (DTU), Denmark. His main activities lie within customised decision analysis, decision support systems, group decision making and applied risk assessments. He holds in-depth knowledge about appraisal methods and methodologies concerning transport infrastructure projects - both in respect to research and teaching. His research in particular applies theory in practice, and has often been carried out using real case data in various projects.