Titre: Title:	Mise en place et expérimentation d'une infrastructure de test de la cybersécurité des technologies opérationnelles navales
Auteur: Author:	Basile Rulh
Date:	2025
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Rulh, B. (2025). Mise en place et expérimentation d'une infrastructure de test de la cybersécurité des technologies opérationnelles navales [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/67795/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/67795/
Directeurs de recherche: Advisors:	Nora Boulahia Cuppens, & Frédéric Cuppens
Programme: Program:	Génie informatique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Mise en place et expérimentation d'une infrastructure de test de la cybersécurité des technologies opérationnelles navales

BASILE RULH
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Août 2025

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Mise en place et expérimentation d'une infrastructure de test de la cybersécurité des technologies opérationnelles navales

présenté par **Basile RULH**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Martine BELLAÏCHE, présidente
Nora BOULAHIA-CUPPENS, membre et directrice de recherche
Frédéric CUPPENS, membre et codirecteur de recherche
Tarek OULD-BACHIR, membre

REMERCIEMENTS

Je remercie Nora et Frédéric Cuppens de m'avoir donné la chance de rejoindre le laboratoire et de mener cette recherche.

Merci à notre partenaire industriel, Chantier Davie de nous avoir permis de visiter leur entreprise, et de nous avoir accordé du temps pour répondre à nos questions.

Je remercie les ami.es du labo, notamment Amine, Vi, Flo, Erwan, Maxime, Seb, Vincent, Bastien, pour tous les bons moments partagés au cours de cette aventure. Un merci tout particulier à Marc-Antoine pour sa double casquette d'ami et d'encadrant.

Merci à Anna pour sa présence et son soutien infaillible et quotidien.

Merci Manue et Seba pour leur accueil, à notre arrivée au Québec et après.

Je souhaite aussi remercier ma famille pour tout le soutien dans les moments durs comme les plus faciles. Merci notamment à Léa et Louise d'avoir relu l'orthographe de ce mémoire.

Merci enfin au jury d'avoir accepté d'évaluer mon travail.

RÉSUMÉ

Le milieu maritime, au centre des enjeux économiques de notre société mondialisée, traverse une transformation numérique. En effet, les navires sont de plus en plus connectés aux côtes, pour des suivis de données, des suivis de l'état des machines, et des dépannages à distances. À terme, l'objectif semble être des vaisseaux autonomes, géré depuis des centres de données. Cette révolution entraîne une prise de conscience du manque de dispositifs de cybersécurité installés dans les navires.

La cybersécurité d'un navire apporte nombre d'enjeux spécifiques. Cela est lié à l'unicité de chaque navire, construit sur mesure, et à un mix de technologies généralistes et spécifiques au milieu naval, qu'elles soient opérationnelles ou informationnelles. L'augmentation du nombre d'attaques, ces dernières années, témoigne de l'écart entre le niveau actuel de la défense mise en place dans les navires et l'augmentation du risque engendrée par les évolutions technologiques.

Pour répondre à ces enjeux, notre recherche vise à mettre en place une plateforme adaptée aux expérimentations de cybersécurité. Cette dernière vise à représenter un navire grâce à l'émulation d'un certain nombre de composants. Nous avons choisi de nous concentrer sur les éléments de technologie opérationnels, car ce sont ceux pour lesquels il existe le moins de solutions actuellement.

Pour être le plus fidèle possible, nous avons inclus plusieurs protocoles présents dans les navires, comme NMEA-0183 ou Modbus-TCP. Notre plateforme représente le pont avec le système ECDIS, et une partie des systèmes de technologies opérationnelles autour du système de propulsion : capteurs et actionneurs, PLC, et serveur SCADA. Enfin, la plateforme intègre un simulateur pour certaines données qui ne peuvent pas être généré par émulation.

Au sein de ce système, nous avons déployé des attaques, pour déterminer quels en étaient les prérequis et les impacts potentiels. Nous avons mis en place des attaques qui visent un hôte, l'ordinateur qui héberge l'ECDIS, pour modifier les données présentées aux membres d'équipage. D'autres types d'attaques qui se concentrent sur le réseau ont aussi été déployées, pour mettre à mal la chaîne de commande de la propulsion.

Enfin, pour la partie défensive, nous évaluons différents logiciels de détection d'intrusion pensés pour les technologies informationnelles dans le contexte de nos attaques sur le réseau de technologies opérationnelles. L'objectif est d'évaluer l'aide que ces outils peuvent apporter dans ce nouveau contexte. Pour ce faire, nous avons aussi écrit des règles spécifiques pour

détecter la modification de certaines données physiques par un adversaire.

L'ensemble de ces éléments permet donc de montrer que nous avons mis en place une plateforme capable d'héberger des expérimentation adaptées à la cybersécurité du maritime. Nous l'avons utilisé pour démontrer les forts impacts qu'une cyberattaque pourrait avoir sur un navire, ainsi que pour évaluer l'utilité des méthodes de détection d'intrusion IT dans ce contexte.

ABSTRACT

Globalization has put sea freight at the center of our economies, and the maritime industry is going through a digital transformation. Ship-to-shore connectivity is getting more and more frequent, as it is used for maintenance, troubleshooting, or simply data transfer. The long-term goal of this transformation seems to be autonomous vessels that could be driven from the shore. All of these adaptations are highlighting the low cybersecurity standards for this industry.

Ship cybersecurity is bringing a lot of specific challenges, since no two ships are the same. Moreover, there is a special mix of technologies, where some are specific to the maritime environment, and others are more generalist. They are also divided between informational technologies and operational technologies. There is a gap between the defense level of ships and the risk that comes from those technological evolutions. The increase in attack numbers shows it.

Our research tries to build a testbed that allow cybersecurity related experimentation. This testbed recreate a ship by emulating components, with a special attention to operationnal technologies.

To increase the fidelity of our testbed, we used protocols that you could find in a ship, such as NMEA-0183 or Modbus-TCP. We included multiple parts of the ship, such as the bridge (ECDIS system), and the propulsion system (sensors and actuators, PLC, and SCADA server). Finally, a simulator is used to generate data that the emulation can't create, such as meteo or positioning.

We deployed some attacks in the testbed to determine what made them possible and what their impact could be. Some attacks are targeted at a specific device, the ECDIS host, to falsify the data available to the crew member, and others are targeted at the network to impact the propulsion command chain.

Next, we tried different intrusion detection systems specialized for informational technologies to evaluate the impact they could have in our context. We also developed specific SNORT rules to detect data tampering by an adversary.

Overall, we developed a testbed that allows cybersecurity experimentations in the maritime field. We used this testbed to show the impacts a cyberattack could have on a ship, and to evaluate intrusion detection systems in this context.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

ECDIS	Electronic Chart Display and Information System
NMEA	National Marine Electronics Association
AIS	Automatic Identification System
TTP	Tactique Technique et Procédures
SCADA	Supervisory Control and Data Acquisition
CAN	Controller Area Network
IP	Internet Protocol
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
IPv6	Internet Protocol version 6
IPv4	Internet Protocol version 4
LAN	Local Area Network
ASCII	American Standard Code for Information Interchange
PLC	Programmable Logic Controller
DNV	Det Norske Veritas
TO	Technologies Opérationnelles
TI	Technologies Informationnelles
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
IDPS	Intrusion Detection and Prevention System
IHM	Interface Humain-Machine
OSI	Open Systems Interconnection
IA	Inteligence Artificielle
ML	Machine Learning
SIEM	Security Information and Event Managment
C2	Commander et Contrôler
USB	Universal Serial Bus
DDoS	Distributed Denial of Service
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
OC-SVM	One Class Support Vector Machine
VDR	Voyage Data Recorder
UTC	Coordinated Universal Time
TZ	TimeZero

GPS	Global Positioning System
MitM	Machine in the Middle
ARP	Address Resolution Protocol
MAC	Media Access Control
PCAP	Packet CAPture
RPM	Rotation Par Minutes
HTTP	HyperText Transfert Protocol
ID	IDentification

CHAPITRE 1 INTRODUCTION

En augmentation constante depuis les années 60, le transport maritime représente plus de 90 % des flux de marchandises en volume, et 80 % en valeur [2]. En 2023, c'était un total de 12,3 milliards de tonnes qui a été transporté, soit quelques 390 tonnes par seconde [3]. Ce mode de transport est particulièrement avantageux pour les transports à très longues distances, c'est donc un élément central de la mondialisation. En effet, sa capacité à transporter d'importantes quantités de marchandises sur de longues distances à des coûts relativement faibles, combinée à la standardisation des conteneurs depuis les années 1960, en fait le pilier logistique du commerce international moderne. Cette efficacité économique s'accompagne également d'un avantage environnemental : grâce aux volumes considérables transportés, l'empreinte carbone par tonne transportée reste inférieure à celle du transport aérien ou routier.

1.1 Digitalisation et risques induits

Cette industrie, comme toutes les autres, évolue avec les progrès techniques, et est notamment dans une phase de digitalisation. Les navires étaient préalablement considérés comme des systèmes fermés qui n'entraient pas en contact avec le reste du monde, mais ils se retrouvent de plus en plus connectés. C'est utile voire nécessaire, par exemple pour des dépannages à distance, de la transmission d'informations en temps réel, permettre un accès internet aux personnes embarquées, etc. S'il est naïf de considérer qu'un navire non connecté est à l'abri de toute cyberattaque, il est indéniable que l'augmentation de la surface d'attaque conduit à une augmentation du risque. D'autant plus que beaucoup des systèmes présents sur les navires ne sont pas du tout pensés avec ces enjeux en tête. Des protocoles comme l'AIS (Automatic Identification System, ou Système d'Identification Automatique), NMEA-0183 (National Maritime Electronics Association, ou l'Association d'Électronique Maritime Nationale des États-Unis), ou NMEA-2000, qui sont les standards du milieu maritime, ont été imaginés sans prendre en compte les enjeux de sécurité. Si même les organismes de normalisation n'ont historiquement pas intégré ces enjeux de sécurité dans leurs standards, il est peu surprenant que de nombreux autres acteurs aient initialement suivi la même approche. Néanmoins, depuis quelques années, on observe une prise de conscience progressive : les organismes de standardisation commencent à intégrer des considérations de cybersécurité dans leurs nouvelles normes, comme la NMEA-OneNet (qui se base sur l'IPv6 (Internet Protocol Version 6, ou sixième version du protocole internet) pour gagner des éléments de

sécurité), même si ces efforts restent encore limités face à l'ampleur du défi.

L'un des gros enjeux de cette digitalisation est l'âge de certains équipements. Certains navires qui sont exploités depuis plusieurs décennies se retrouvent ainsi exposés sans qu'il ait été possible de retravailler en profondeur leur architecture. En effet, le temps et l'expertise nécessaires pour faire des modifications en profondeur de machines aussi complexes qu'un navire rend difficilement envisageable le process pour des armateurs : les coûts sont démesurés, en plus de représenter un formidable manque à gagner. Il est important de prendre en compte que chaque navire est unique, construit pour des besoins spécifiques, ce qui rend la tâche d'autant plus ardue.

Ainsi, on se retrouve avec une défense relativement superficielle. Idéalement, on voudrait une défense dite en profondeur (concept présenté à la fin de la partie 1.3), c'est-à-dire qui cumule les mesures de protection : depuis la sécurisation physique des équipements jusqu'à la surveillance du réseau, en passant par l'authentification des utilisateurs et le chiffrement des communications. Or, la situation actuelle est bien différente : si les pare-feux semblent être implémentés et sécurisés, nécessitant même parfois une clef physique pour les ouvrir, le reste de l'architecture est terriblement démunie. Comme précisé précédemment, les protocoles n'embarquent pas de sécurité intrinsèque, ce qui signifie qu'un attaquant ayant réussi à pénétrer le réseau peut intercepter et modifier les communications sans rencontrer de résistance supplémentaire. De plus, il n'y a pas d'IDS (Intrusion Detection Systems, ou systèmes de détections d'intrusions) spécialisés qui ont pu être testés, privant ainsi les navires d'une couche cruciale de détection qui pourrait identifier les comportements suspects avant qu'ils ne causent des dommages.

Et les adversaires ne s'y trompent pas : entre 2017 et 2020, le nombre de cyberattaques subies par les systèmes maritimes a augmenté de 900 % [4] ! Malgré cela, cette tendance à la digitalisation va s'amplifier comme en témoigne la volonté d'aller vers des bâtiments autonomes [5–7].

Une enquête de l'entreprise DNV (Det Norske Veritas) sur l'année 2024 [8] interroge un échantillon significatif de 500 professionnels du milieu maritime. Elle présente un certain nombre de chiffres marquants sur le lien entre la cybersécurité et le milieu maritime. La taille de l'échantillon et le fait qu'il soit composé de spécialistes du maritime rend l'enquête particulièrement précieuse pour nous.

Le premier chiffre révèle que 61 % des répondants considèrent que leurs institutions doivent accepter le risque cyber qui accompagne la digitalisation des systèmes. Cette statistique est particulièrement préoccupante, car elle suggère une forme de résignation face aux menaces cybernétiques. Cette attitude pourrait s'expliquer de plusieurs manières : soit par une per-

ception que les bénéfices de la digitalisation surpassent largement les risques, soit par un sentiment d'impuissance face à ces menaces, ou encore par une sous-estimation des conséquences potentielles d'une cyberattaque. Néanmoins, elle montre que les professionnels ont conscience que le risque cyber existe, même s'ils et elles semblent le sous-estimer.

Cette enquête nous apprend aussi que 68 % des professionnels considèrent que le risque est plus important pour les TO (Technologies Opérationnelles) que pour les TI (Technologies Informationnelles). En effet, si les TI peuvent être protégées par des solutions de sécurité standardisées et éprouvées dans d'autres industries, les TO présentent des défis uniques : elles contrôlent des systèmes physiques critiques, utilisent souvent des protocoles propriétaires ou datés, et une compromission peut avoir des conséquences directes sur la sécurité des personnes et des biens.

1.2 Technologies Opérationnelles

On appelle technologies opérationnelles l'ensemble des systèmes informatiques qui contrôlent directement les processus physiques du navire. Contrairement aux TI, qui gèrent les données administratives, les TO interagissent avec le monde réel à travers des capteurs et des actionneurs. Par exemple, dans le cas d'un méthanier, de nombreux composants de TO interdépendants sont présents : des capteurs de température surveillent en permanence l'état du gaz naturel liquéfié maintenu très froid, des dispositifs de mesure de pression contrôlent les conditions dans les cuves, des détecteurs de niveau suivent le volume de la cargaison, et des systèmes de compression gèrent la récupération des vapeurs de gaz. Tous ces éléments sont orchestrés par des automates programmables logiques (PLC, Programable Logic Controler) qui doivent maintenir des conditions extrêmement précises. Le moindre dysfonctionnement dans ce système pourrait non seulement endommager la cargaison d'une valeur de plusieurs millions d'euros, mais aussi créer des situations dangereuses, comme des explosions, des accidents, des bris de matériels, etc. Les données de tous ces capteurs convergent vers une salle de contrôle, où des opérateurs surveillent en permanence les paramètres critiques et peuvent intervenir si nécessaire via des interfaces hommes-machines spécialisées.

Les TO représentent un enjeu bien particulier : leurs architectures et les protocoles utilisés reprennent certains éléments d'autres industries, mais restent très spécifiques. Il est donc nécessaire de développer des solutions de sécurité tout aussi spécifiques. Pourtant, les éléments actuellement déployés ne sont majoritairement pas pensés avec la sécurité informatique en tête, à l'image des protocoles AIS et NMEA-0183. Beaucoup de ces systèmes sont vieux, beaucoup aussi sont propriétaires.

Un autre enjeu spécifique des TO est le temps réel : lorsque le pilote cherche à faire tourner le navire, le gouvernail doit immédiatement se tourner. De même, lorsqu'une cuve devient légèrement trop chaude, il faut immédiatement la refroidir pour éviter tout risque. Les cycles de décision de ces systèmes se mesurent donc en millisecondes. Ainsi, de nombreuses méthodes habituelles de cybersécurité ne peuvent être utilisées à cause de la latence qu'elles provoquent. Il n'est pas envisageable non plus d'interrompre certaines activités en attendant une intervention humaine, ou de bloquer certains comportements, même très suspects, de façon automatique.

Dans l'écosystème des technologies opérationnelles maritimes, les capteurs et actionneurs constituent les éléments fondamentaux de la chaîne de perception et d'action. Les capteurs, véritables systèmes nerveux du navire, collectent en permanence des données physiques : position du gouvernail, niveau de carburant, vitesse du navire, température des cuves, pression hydraulique. Ces informations numériques sont converties et transmises aux PLC qui les analysent instantanément. Les actionneurs, en réponse, transforment les ordres des PLC en actions mécaniques concrètes : activation de pompes, orientation des gouvernails, déclenchement de systèmes de refroidissement ou de sécurité. Cette interaction ultra-rapide entre capteurs, automates et actionneurs garantit la réactivité et la précision des systèmes maritimes modernes.

Ces appareils peuvent être d'une complexité informatique très variable : certains sont globalement de gros interrupteurs avec une seule commande, d'autres ont une complexité bien plus avancée, jusqu'à des systèmes d'exploitation temps réels. De même, certains n'obéissent qu'à une impulsion électrique, et d'autres reçoivent des messages complexes à travers des protocoles comme IP. Une fois encore, leur durée de vie, leur diversité et le fait qu'ils soient principalement propriétaires rend leur étude extrêmement difficile.

Les PLC constituent l'épine dorsale des systèmes de contrôle maritime, servant d'intermédiaires entre les capteurs, les actionneurs, et les IHM qui permettent aux opérateurs et opératrices d'interagir avec le système. Ces automates programmables sont conçus spécifiquement pour résister aux conditions difficiles rencontrées en mer, comme les vibrations, l'humidité et les variations de température. Leur architecture robuste et leur fonctionnement par cycle pour remplir des tâches en temps réel leur permettent d'assurer une fiabilité exceptionnelle, avec des taux de disponibilité bien supérieurs à des ordinateurs ou des serveurs classiques, ce qui est crucial pour maintenir la sécurité et l'efficacité opérationnelle du navire.

La complexité croissante des systèmes maritimes modernes a considérablement élargi le rôle des PLC, qui ne se limitent plus à de simples tâches de contrôle binaire. Ces dispositifs gèrent désormais des processus sophistiqués impliquant des calculs complexes, des communi-

cations en réseau et des interactions avec d'autres systèmes automatisés comme d'autres PLC ou des serveurs SCADA. Cette évolution, bien qu'améliorant significativement les capacités opérationnelles, introduit évidemment de nouvelles vulnérabilités cyber.

Les systèmes SCADA représentent une couche supérieure de contrôle ou de supervision qui s'intègre aux PLC pour offrir une vue d'ensemble cohérente des opérations maritimes. Ces services tournent sur du hardware et des systèmes d'exploitation proche des systèmes de TI, mais qui se différencient par leurs usages. Ces systèmes sophistiqués agrègent les données provenant de multiples automates et capteurs répartis sur l'ensemble du navire, permettant ainsi aux opérateurs de surveiller et de contrôler l'intégralité des processus depuis des interfaces centralisées. Cette centralisation de l'information facilite non seulement la prise de décision opérationnelle, mais permet également l'analyse des tendances à long terme et l'optimisation des performances du navire.

Les serveurs SCADA communiquent avec les PLC via divers protocoles industriels (MODBUS, Profinet, etc.), dont beaucoup ont été conçus à une époque où la cybersécurité n'était pas une préoccupation majeure, et sont propriétaires, donc difficilement analysables. Cette réalité historique, combinée à la nécessité croissante d'intégrer ces systèmes avec des technologies plus modernes et de les ouvrir sur Internet, crée un défi particulier en matière de cybersécurité. La protection de ces systèmes nécessite une approche équilibrée qui préserve leur capacité à fournir des contrôles en temps réel tout en implémentant des mesures de sécurité adaptées à l'environnement maritime.

Voici donc les trois niveaux principaux d'une architecture de TO d'un navire :

- Les capteurs et actionneurs sont le niveau le plus bas, qui interagit avec le monde réel, et présente ces données à au niveau intermédiaire.
- Les PLC sont le niveau intermédiaire. Ils agissent comme un réseau interconnecté de décision, et ordonnent aux actionneurs en fonction des décisions provennant des humains et des informations qui remontent des capteurs.
- Les IHM, et notamment les serveurs SCADA sont le niveau le plus haut, qui agrège les données de manière lisible pour des humains, et peut recevoir les ordres généraux, pour les communiquer aux PLC.

Chacun des éléments d'un navire est souvent présent en plusieurs exemplaires, selon un principe de redondance : dans l'hypothèse où un appareil venait à tomber en panne, un deuxième appareil capable d'effectuer le même travail devrait entrer en action pour ne pas interrompre le fonctionnement normal du navire.

Après avoir présenté les technologies opérationnelles dans le milieu maritime, il est temps de passer à des éléments de cybersécurité.

1.3 Éléments de cybersécurité pour le maritime

Dans cette partie, nous présentons différents éléments de cybersécurité indispensables à la compréhension de la suite de ce mémoire. Certains ont déjà été cités, mais seront maintenant détaillés. Nous commencerons par présenter certains des éléments de défense fréquemment utilisés en cybersécurité tels que des pare-feux ou des IDPS. Nous présenterons ensuite certaines théories et outils de réflexion que nous utiliserons dans ce mémoire, tels que les Kill Chains, le framework MITRE ATT&CK, ou la défense en profondeur.

Les pare-feux sont des éléments applicatifs qui ont pour objectif de segmenter un réseau en plusieurs parties, et de contrôler les communications entre ces parties. Ils agissent comme garde-barrière, autorisant certains messages ou certaines communications à passer, et bloquant certaines qui semblent suspectes. En fonction de leur complexité, les pare-feux peuvent prendre des décisions sur plus ou moins d'information : certains vont par exemple n'autoriser la communication que sur certains ports, où uniquement depuis certaines adresses IP, et les plus complexes peuvent analyser le contenu d'un message pour prendre une décision (on les appelle pare-feu applicatifs, en référence à la couche du système OSI (Open Systems Interconnection, ou interconnexion des systèmes ouverts) sur laquelle leur analyse se base). Évidemment, puisque chaque message doit être traité par le pare-feu avant qu'il ne soit transmis au sous-réseau protégé, il faut que la capacité computationnelle associée soit suffisante pour ne pas retarder les messages analysés.

Il est évidemment très fréquent de les placer à l'entrée d'une institution, différenciant "l'internet" (réseau extérieur) du réseau interne, ce dernier étant considéré comme plus sécurisé. De nombreux ordinateurs particuliers utilisent également des pare-feux pré-configuré par le système d'exploitation pour contrôler toute communication avec une autre machine, ajoutant ainsi une couche de défense. Évidemment, il est aussi envisageable de découper un réseau que l'on administre en deux avec un pare-feu, soit en considérant qu'une partie du réseau doit être plus sécurisée que l'autre, soit en considérant que les deux ont des rôles très distincts et donc que seuls certains type de communication doivent être transmis. C'est notamment souvent le cas entre des réseaux TI et des réseaux TO, ou entre plusieurs parties d'un réseau TO.

Dans un navire, les pare-feux semblent assez fréquemment utilisés. Jusqu'à récemment, ils bloquaient toute communication, et nécessitaient une intervention humaine (par exemple l'utilisation d'une clef physique) pour autoriser une connexion. Ainsi, le lien avec l'extérieur restait exceptionnel, et le pare-feu relativement simple à programmer. Cependant, la digitalisation présentée précédemment implique des communications constantes entre le navire et

l'extérieur, et donc une relaxation des règles du pare-feu. Des pare-feux internes semblent aussi fréquents, entre le réseau TI qui sert à l'équipage et les réseaux de TO.

Un autre système de sécurité très fréquent est l'IDS. Leur objectif est de détecter les intrusions. Ils peuvent être séparés en deux grandes catégories : ceux qui s'intéressent principalement aux réseaux (souvent simplement appelés IDS réseau), et ceux qui s'intéressent principalement à une machine (souvent simplement appelés IDS local, ou hôte).

Un IDS réseau va fonctionner un peu comme un pare-feu, mais avec moins de contraintes sur la latence. Il reçoit une copie de chaque communication, et va lever une alerte lorsqu'il détecte quelque chose de suspect. Il ne faut donc pas avoir plusieurs semaines de délai, car l'alerte serait inutile, mais la communication n'attend pas une décision de l'IDS à chaque message. L'IDS a donc aussi la capacité de traiter les messages par groupe, donc de prendre une décision avec plus d'informations.

Plusieurs méthodes peuvent être utilisées pour détecter une communication suspecte :

— Analyse par signature : Certaines attaques ont déjà été identifiées, et se déroulent toujours de la même manière. Elles sont donc entrées dans une base de données sous la forme de signature. Toute communication avec la même signature, donc identique, sera évidemment considérée comme une nouvelle attaque.
— Analyse statistique : Les réseaux ont des constantes statistiques. Par exemple, le nombre de commandes sur une boutique en ligne est plus important les jours de paye qu'en fin de mois. Ces constantes statistiques peuvent être observées en temps réel, et un changement sur l'une d'elles peut signifier quelque chose d'anormal (sortie d'une nouvelle collection ou cyber-attaque, après il faut analyser l'alerte).
— Analyse comportementale : Plutôt que de raisonner sur des grands nombres, une analyse comportementale va observer le comportement individuel d'une communication. Par exemple, une connexion à un compte sécurisé de travail à trois heures du matin depuis l'autre bout du monde est incontestablement un comportement inattendu qui lèvera une alerte. Est-ce un employé en vacances qui fait une insomnie ou une vraie cyber-attaque, encore une fois seule une analyse approfondie de l'alerte peut permettre de faire le tri.
— Analyse par intelligence artificielle : Il est aussi possible d'utiliser des méthodes de machine learning ou d'autre méthodes d'intelligence artificielle pour détecter des attaques, soit spécifiquement en les entraînant à travailler les éléments présentés précédemment, soit d'une façon moins anthropomorphe, simplement en leur donnant les données et en leur demandant de les trier.

Un IDS local va plutôt analyser les exécutables sur un appareil, étudier leur code avant de

les démarrer et observer leur comportement une fois qu'ils tournent.

- Analyse par signature : Comme dans un réseau, il est possible d'étudier chaque exécutable pour savoir s'il correspond à un maliciel déjà enregistré dans une base de données existante, qui serait réutilisé tel quel.
- Analyse statique du code : Il est aussi possible d'analyser le code assembleur avant de faire tourner le logiciel, afin d'essayer de détecter des parties de code qui correspondent à un comportement suspect, ou du moins incohérent avec le comportement attendu du logiciel.
- Analyse en bac à sable : Il s'agit de démarrer le logiciel étudié dans un environnement contrôlé par l'IDS, comme une machine virtuelle. Il est donc possible d'étudier très finement son comportement, puisque l'environnement est entièrement conçu pour, mais ce n'est malheureusement pas envisageable de faire durer ces analyses trop longtemps pour chaque exécutable.
- Analyse de comportement : Durant l'exécution standard du programme, il est toujours possible d'étudier certains comportements, puis de vérifier s'ils correspondent aux comportements attendus du logiciel, même si l'on a moins d'informations que dans un environnement entièrement contrôlé par l'IDS.

Il faut ensuite décider de la réaction suite à une alerte. Comme son nom l'indique, le rôle d'un IDS est de détecter un risque, donc il se contentera de prévenir des experts qui devront enquêter et prendre les mesures adéquates. Cependant, certains outils vont plus loin et prennent directement des mesures pour prévenir les conséquences d'une attaque détectée : ce sont des IPS (Intrusion Prevention System ou système de prévention d'intrusion). Ces systèmes peuvent prendre des actions automatiquement avant intervention humaine, notamment pour limiter la propagation d'une attaque au sein du réseau, ou d'autre actions pour lesquelles le temps de réaction est très important. Il est évidemment possible d'envisager un système hybride, où le système va lever une alerte dans tous les cas, et prendra des actions en fonction de l'évolution et de son niveau de certitude dans l'alerte. Évidemment, une fausse alerte sur un IPS est bien plus dangereuse qu'une fausse alerte sur un IDS, et certains ou certaines attaquantes pourraient les exploiter pour générer intentionnellement des alertes et donc les protocoles automatisés de l'IPS. Ce sont donc des outils précieux, mais à utiliser avec précaution.

Toutes les alertes qui sont remontées pour traitement humain sont renvoyées vers un autre logiciel, nommé le SIEM (Security Information and Event Management, gestion des informations et événements de sécurité). Cet outil va centraliser toutes les données utiles aux défenseurs pour pouvoir enquêter. Il contient évidemment les alertes générées par l'IDS en

temps réel, mais permet aussi d'accéder aux fichiers de logs des pare-feux, des ordinateurs et des serveurs. Il peut contenir des informations sur les comportements des utilisateurs, des observations sur le réseau, etc. Toutes ces informations sont présentées de façon à pouvoir facilement les corréler et les lier, avec pour objectif de permettre aux agents de défense de vite trouver la source d'une alerte (qu'elle soit fausse ou vraie).

Voici donc une présentation des principaux outils de défense qui seront mobilisés dans ce mémoire. Il s'agit évidemment d'une introduction à ces outils, et tous ne sont pas présentés.

Pour savoir comment bien les utiliser, il faut savoir comment fonctionne une cyber-attaque, afin de savoir où et quand il est opportun de bloquer l'adversaire. L'analyse des adversaires et de leurs comportements en fonction de leurs objectifs, de leurs moyens, et leur catégorisation est un champ de recherche à part entière dont ce mémoire ne fait pas partie, mais nous en mobiliserons tout de même certaines notions.

Chaque attaque correspond à une chaîne d'attaque (souvent appelée Kill Chain), c'est-à-dire à une suite d'actions, qui permettent à l'adversaire d'atteindre ses objectifs. En effet, il est rare qu'une seule action lui permette de les atteindre, il lui faut souvent enchaîner plusieurs actions, par exemple pour entrer dans le réseau de la victime, puis obtenir des privilèges administrateurs. Enfin seulement, l'adversaire atteindra sa cible, quelle qu'elle soit.

Il y a eu plusieurs tentatives pour mettre un cadre théorique autour de ce concept afin de mieux pouvoir appréhender les attaques et donc mieux les contrer. L'une des premières est la Cyber Kill Chain ®, développer par l'entreprise Lockheed-Martin en 2011 [9]. Cette solution propose sept étapes successives, qui caractériseraient toutes les attaques. C'est une vision très linéaire, puisque toutes les étapes doivent être présentes et exécutées dans un ordre précis. Cependant, de nombreux cas sortent de cette ligne pré-établie et empruntent des chemins plus courts, par exemple une attaque de Déni de Service Distribuée n'a que deux étapes : la reconnaissance et la surcharge. Dans certains cas, les attaques prennent des chemins plus longs et détournés, pour se déplacer plusieurs fois au sein du système de la cible, vers des zones coupées d'Internet, ou autre.

Pour dépasser ces critiques, Paul Pols à développé un nouvel outil nommé la Unified Kill Chain [10]. C'est une version qui reste linéaire sur trois grandes étapes, mais au sein de chaque étape, plusieurs actions sont présentés comme pouvant s'appliquer dans n'importe quel ordre. Les trois grandes étapes sont donc les suivantes :

1. L'entrée : Correspond au moment où l'adversaire va entrer dans le système de la victime. Il s'agit donc de la préparation, de la première brèche, et des premières actions une fois introduit.

2. À travers : Correspond au moment où l'adversaire se déplace à travers le système cible, afin de s'approcher de ses objectifs. Il va donc s'agir de mouvements latéraux au sein du système et d'augmentation des privilèges auxquels l'adversaire a accès.
3. La sortie : Correspond au moment où l'adversaire a atteint ses objectifs, et c'est ici que l'impact de l'attaque se ressent. Il dépend évidemment des objectifs de l'attaque, mais il peut s'agir notamment d'effets directs sur l'infrastructure ou de l'exfiltration de données.

Cette version est bien plus modulaire, et évite donc beaucoup des défauts qu'avait la première version. Néanmoins, elle reste relativement abstraite, ce qui ne permet pas forcément de réfléchir à toute la granularité des actions que l'adversaire peut déployer au niveau technique.

Pour cela, il existe un troisième outil, le Framwork MITRE ATT&CK [11], qui a été publié par la société à but non-lucratif étasunienne MITRE en 2013. Cette matrice cherche justement à répertorier toutes les actions utilisables par un adversaire. La méthode est de réfléchir à trois niveaux différents : les Tactiques, les Techniques, et les Procédures (souvent abrégé TTP). Bien que cette notion n'ait pas été inventée par la fondation MITRE, c'est le fondement de leur matrice ATT&CK.

Les tactiques sont les raisonnements stratégiques de grandes échelles. Il s'agit donc de ce que l'adversaire va vouloir accomplir pour atteindre son but. Cela correspond globalement aux étapes décrites par les chaînes de frappes.

Les techniques sont les outils à la disposition de l'adversaire. Chaque technique correspond à une tactique qu'elle permet. Par exemple, le scan actif est une technique permettant de faire de la reconnaissance, et l'usurpation d'identité numérique est une technique qui permet la discrétion vis-à-vis de la détection. Fin janvier 2025, il existe un total de 236 techniques réparties sur l'ensemble des tactiques, liste qui est régulièrement mise à jour en fonction des comportements qui sont observés lors de cyberattaques.

Les procédures, enfin, correspondent à la méthode exacte qui va être utilisée pour la technique. Certaines techniques n'ont pas de procédures associées, et sont à la fois techniques et procédures, mais la plupart ont un certain nombre de procédures associées. Pour reprendre les exemples précédents, un scan actif peut se faire en direction de ports et d'IP, mais aussi en utilisant des listes de mots courants pour vérifier si un site web ne contient pas des pages à ce nom (qui pourrait ne pas être accessible par hyperlien, mais quand même contenir des vulnérabilités). Dans les deux cas, c'est un scan actif : on envoie des requêtes et on déduit de la réponse des informations sur le système, mais la cible est l'information récupérée. Pour l'usurpation d'identité numérique, il est possible de l'appliquer à des comptes utilisateurs,

mais aussi à des fichiers (par exemple cacher un programme exécutable en simple fichier PDF).

Ce grand niveau de granularité permet d'observer très finement les cyberattaques et de communiquer très efficacement entre spécialistes. De plus, MITRE ATT&CK répertorie aussi les groupes connus pour utiliser chaque TTP, ce qui en fait un outil d'attribution très utile, même si ce n'est pas le sujet de ce mémoire.

En allant de plus en plus dans le détail de l'ensemble des actions disponibles aux adversaires, et le niveau de flexibilité avec lequel ils et elles les utilisent, on prend conscience de la diversité des attaques possibles sur un même système. Combinée à l'adage "le risque zéro n'existe pas", cette réalisation implique alors l'idée de défense en profondeur.

Cette notion s'oppose à l'idée d'une défense en périphérie, qui voudrait qu'on sécurise énormément les frontières du système, et donc que jamais l'adversaire n'ait accès au système. Dans ce cas, il est inutile de rajouter des mécanismes de sécurité à l'intérieur du système, car on le considère hors de danger. À l'inverse, la défense en profondeur suppose toujours que chaque élément du système sera compromis, et donc qu'il faut que le reste du système soit résiliant indépendamment de lui. C'est évidemment impossible à mettre en pratique, mais l'idée de démultiplier les mécanismes de défenses à l'intérieur d'un système a fait ses preuves. Par exemple, placer un pare-feu entre les systèmes de TI et de TO, c'est de la défense en profondeur, car ça n'empêcherait pas un adversaire d'entrer dans le système. Par contre, dans l'hypothèse d'une intrusion, ce pare-feu peut limiter l'impact de l'attaque, ou la rendre plus difficile.

Du point de vue de la Unified Kill Chain, la défense en périphérie correspondrait à éviter uniquement l'étape de *l'entrée*, alors que la défense en profondeur encourage à considérer chacune des étapes et à essayer de les rendre plus difficiles.

En prenant en compte les enjeux spécifiques du milieu maritime et les différents concepts de cybersécurité présentés, des éléments indispensables à la mise en place de solution défensive peuvent être mis en avant :

— Passivité : Il n'est pas envisageable de créer des systèmes de sécurité qui prennent des actions automatiques. Ce genre de système comporte des risques trop grands en cas de faux positifs. De plus, il serait possible pour un individu mal intentionné de les utiliser dans le cadre d'une attaque, en se comportant de manière à les déclencher et donc à impacter le comportement du navire. Ainsi, tous les systèmes de prévention d'intrusion (IPS) sont à éviter.
— Vitesse de traitement : Les communications entre les éléments industriels sont très

fréquentes, et impliquent des débits conséquents. Ainsi, une solution de détection d'anomalie qui analyserait ces messages se doit d'être capable d'analyser le débit qu'on lui soumet, même dans l'hypothèse où l'adversaire ajoute des messages. Il faut donc soit prévoir une grande capacité de calcul centralisé, soit de répartir la charge dans plusieurs systèmes.

— Facilité d'intégration : Il n'est pas envisageable de modifier l'architecture des navires de fond en comble pour ajouter une solution de sécurité, il faut qu'elle puisse s'intégrer facilement, et qu'elle ne nécessite pas trop de maintenance.

— Polyvalence : Puisque chaque navire est unique, il faut que notre solution puisse s'adapter aux spécificités métiers de ce navire, soit en fonctionnant sans les prendre en compte, soit en étant capable de les apprendre via une phase d'analyse d'un fonctionnement normal. La solution doit être capable de s'intégrer dans des architectures différentes, et de fonctionner avec une certaine variété de protocoles.

C'est à partir de ces éléments que l'on va maintenant définir les objectifs de recherche.

1.4 Objectifs de recherche

Il est très complexe de présenter une solution remplissant tous ces critères. Au vu de l'opacité de l'industrie, il est déjà difficile de vérifier si une solution fonctionne. C'est pour cela qu'il est important de mettre en place une plateforme de test, qui permet de se rapprocher au maximum de conditions réelles au sein d'un laboratoire. Ainsi, le premier objectif de recherche sera le suivant :

Comment concevoir une infrastructure de simulation, de test, et d'entraînement sur la cybersécurité des technologies opérationnelles maritime ?

Au sein de la plateforme, pour pouvoir étudier les risques liés à une cyberattaque et pouvoir évaluer une solution de détection, il est important de pouvoir mettre en place nous-même des attaques. Le second objectif de recherche est donc celui-ci :

Quels types de cyberattaques implémenter pour démontrer les impacts potentiels sur le système naval et son environnement opérationnel ?

Grâce à la plateforme et aux attaques qui y sont déployées, il devient alors possible d'analyser différentes solutions de sécurité. Étant donné le manque de maturité du domaine spécifique de la détection d'intrusion dans un navire, on essayera plutôt d'étudier des éléments connus et fonctionnels dans le milieu des TI. Le dernier objectif de recherche est donc formulé comme suit :

Quelles sont les limites et intérêts des solutions de détection d'intrusion prévues pour les

technologies informationnelles dans le contexte du domaine maritime ?

1.5 Plan du mémoire

Le deuxième chapitre présente une revue de la littérature de différentes plateformes de cybersécurité pour le maritime, dans l'industrie comme dans la recherche publique. Il inclut aussi des solutions de détection pour le maritime prises dans l'état de l'art. Le troisième chapitre décrira la mise en place de notre plateforme. Le quatrième chapitre sera dédié à la mise en place des attaques dans la plateforme, et le cinquième à l'étude d'éléments de détection d'intrusion. Enfin, la sixième partie conclut avec les limites de nos travaux, et des propositions de travaux futurs.

CHAPITRE 2 REVUE DE LITTÉRATURE

Pour mener à bien cet état de l'art sur la cybersécurité dans le domaine maritime, une méthodologie de recherche rigoureuse a été mise en place. L'ensemble des références bibliographiques a été géré et organisé à l'aide de l'outil Zotero [12], qui permet non seulement de stocker les références, mais également de les classer par thématiques et d'annoter les documents pour faciliter l'analyse ultérieure.

La recherche documentaire s'est principalement appuyée sur Engineering Village, une plateforme qui donne accès à des bases de données spécialisées dans les domaines de l'ingénierie et des sciences appliquées. Plus spécifiquement, deux bases de données complémentaires ont été exploitées : Compendex, qui constitue la base de données bibliographique la plus complète en ingénierie, et Inspec, qui se spécialise dans les domaines de la physique, de l'électronique, de l'informatique et des technologies de l'information, donc particulièrement adaptée dans notre cas. Ce fonctionnement sur deux bases de données a pu créer des doublons, mais ils ont été traités manuellement et cela a permis une recherche plus exhaustive.

Afin de cibler précisément les publications pertinentes, une stratégie de recherche par mots-clés a été élaborée. Cette stratégie s'est articulée autour de la combinaison des thématiques `"naval"` et `"cybersécurité"`, permettant ainsi d'identifier les travaux se situant à l'intersection de ces deux domaines. Cette approche a permis de filtrer efficacement la littérature scientifique pour ne retenir que les publications traitant spécifiquement des problématiques de cybersécurité dans le contexte maritime.

Le thème naval a été détaillé autour des notions de `'naval'`, `'maritime'`, et `'boat'`. Le terme `'ship'` a été expérimenté, mais finalement supprimé, car il apportait certains articles liés au domaine spatial.

Le thème cybersécurité a été détaillé autour de plusieurs formes (attachées, avec espace, et avec tiret), des mots `cybersécurité` et `cyberattaques`, ainsi que des outils spécifiques `'IDS'`, `'IPS'`, `'IDPS'`, et `'SIEM'`.

La méthode de recherche a également été enrichie par une approche en cascade. Lorsqu'un article particulièrement pertinent était identifié, une exploration systématique de ses références bibliographiques était réalisée afin de remonter aux sources originales des concepts présentés. De manière complémentaire, une recherche des publications citant cet article était effectuée, permettant ainsi d'identifier les développements plus récents s'appuyant sur ces travaux. Cette méthode, souvent désignée comme "citation mining" (ou minage de citations) dans la

littérature scientométrique, a permis d'élargir le corpus documentaire tout en garantissant sa cohérence thématique.

Cette méthodologie bidirectionnelle, associant la recherche par mots-clés dans des bases de données spécialisées et l'exploration des réseaux de citations, a permis de constituer un corpus documentaire exhaustif reflétant l'état actuel de la recherche et des développements en matière de cybersécurité maritime. L'ensemble des références identifiées a été systématiquement importé, classé et annoté dans Zotero, facilitant ainsi l'analyse critique présentée dans les sections suivantes.

Ce chapitre a pour objectif de présenter l'état de l'art de ce qui se fait en cybersécurité dans le domaine du maritime. Nous allons commencer par la présentation de différentes plateformes de recherche en cybersécurité pour le maritime, qu'elles soient produites par des laboratoires académiques ou par des entreprises privées. En deuxième partie, nous détaillons les différents articles qui proposent de mettre en place des éléments de détection d'attaques liés au domaine.

2.1 Plateformes de recherche de cybersécurité pour le maritime

Avant de présenter les plateformes en tant que telles, il faut savoir ce que l'on cherche dans une plateforme de recherche en cybersécurité pour le maritime. Pour cela, un article de revue de littérature par Uchenna Ani et al. [13] ayant pour objectif de déterminer les éléments de conception qui rendent crédibles des plateformes de sécurité dans le domaine industriel a fait ressortir 14 caractéristiques importantes. À chacune de ces caractéristiques est associée une proportion d'impact, qui permet de mesurer l'impact de ladite caractéristique sur la crédibilité de la plateforme. La proportion d'impact est tirée du nombre d'articles présentant une plateforme qui revendique viser cet objectif. Ces données peuvent être retrouvées dans le tableau 2.1.

La fidélité d'une plateforme de test représente à quel point la simulation reflète la réalité. Cette fidélité peut être évaluée selon le type d'approche de simulation adoptée : purement logicielle, purement physique, ou une combinaison des deux (hybride). En règle générale, une infrastructure privilégiant les simulations physiques atteint un niveau de fidélité supérieur à celle reposant principalement sur des simulations logicielles. La fidélité est également déterminée par le réalisme des simulations effectuées. Plus précisément, la capacité à reproduire des flux de données complexes se rapprochant de ceux observés sur un navire réel contribuera directement à améliorer la fidélité de l'infrastructure.

L'évolutivité est la caractéristique d'une plateforme qui définit sa capacité à s'adapter, à

Caractéristiques	Proportion d'impact (%)
Fidélité	41.46
Évolutivité	28.83
Flexibilité	19.51
Reproductibilité	19.51
Modularité	17.07
Coût-efficacité	9.76
Mesurabilité	9.76
Exécution sécuritaire	7.32
Diversité	4.88
Convivialité	4.88
Interopérabilité	2.44
Surveillance et journalisation	2.44
Complexité	2.44
Ouverture	2.44

TABLEAU 2.1 Tableau récapitulatif des caractéristiques impactant la crédibilité d'un banc de test et leurs proportions d'impact.

s'étendre. Une infrastructure démontre une bonne évolutivité lorsque son architecture permet d'intégrer facilement de nouveaux capteurs dans ses différents sous-systèmes, sans nécessiter une reconstruction complète de l'infrastructure. Cette flexibilité permet d'enrichir les fonctionnalités du système de manière progressive et efficace.

La flexibilité d'une plateforme représente sa capacité à s'adapter et à se reconfigurer pour répondre à divers scénarios d'utilisation. Par exemple, une bonne flexibilité permet aussi bien l'exécution de tests de différents scénarios d'attaques que l'analyse de solutions de détection. Cette polyvalence témoigne de sa capacité à modifier facilement sa fonction selon les besoins, sans compromettre son efficacité.

La reproductibilité d'une plateforme caractérise sa capacité à être recréée et à reproduire les scénarios qui y sont modélisés. Cette caractéristique revêt une importance particulière dans un contexte scientifique, car elle permet à d'autres chercheurs de recréer un environnement identique et de reproduire les mêmes scénarios, notamment pour tester et valider différentes solutions. C'est un élément essentiel pour la recherche, mais qui s'applique malheureusement moins à l'industrie privée.

La modularité d'une plateforme définit sa capacité à s'adapter aux nouvelles exigences qui émergent au fil du temps. Par exemple, elle se manifeste par la capacité à intégrer de nouveaux protocoles, permettant ainsi de rester en phase avec l'évolution des pratiques industrielles. Une architecture bien conçue, offrant la possibilité d'ajouter aussi bien des simulations de

réseaux que des composants physiques, garantit une plateforme véritablement modulable et adaptable aux besoins futurs.

La mesurabilité d'une plateforme représente sa capacité à collecter et à quantifier des données sur le processus de test, sans que cette collecte n'interfère avec les résultats obtenus. Les données collectées peuvent être des données de performances, les résultats de systèmes de détection, ou les impacts des attaques.

Le coût-efficacité d'une plateforme reflète sa capacité à atteindre les objectifs fixés et à exécuter les scénarios prévus tout en respectant les contraintes budgétaires, particulièrement dans un contexte de recherche. Dans notre cas de recherche universitaire où les ressources financières, spatiales et temporelles sont limitées, plusieurs décisions de conception impactent directement cette caractéristique. Par exemple, le choix d'utiliser des machines virtuelles pour la simulation d'environnements, plutôt que des équipements physiques, peut représenter une solution plus économique. Toutefois, il est important de noter que la recherche d'un meilleur coût-efficacité peut parfois compromettre la fidélité de la plateforme, notamment lorsque les outils moins coûteux ne peuvent pas offrir le même niveau de réalisme que leurs alternatives plus onéreuses.

L'exécution sécuritaire garantit que la simulation des différents scénarios au sein de la plateforme n'entraîne aucun impact sur la sécurité des systèmes réels. Autrement dit, il ne faut pas qu'un scénario puisse avoir un impact en dehors de la plateforme.

La diversité d'une plateforme représente sa capacité à intégrer une large gamme de composants tout en préservant son évolutivité, permettant ainsi de simuler un vaste éventail de configurations. C'est particulièrement important dans le cadre de notre recherche sur les technologies opérationnelles du maritime, un domaine où les architectures et les technologies sont particulièrement diverses. Cette polyvalence enrichit les possibilités de simulation sans compromettre la capacité d'évolution de la plateforme.

La convivialité d'une plateforme caractérise sa facilité d'utilisation par d'autres utilisateurs ou utilisatrices, avec pour objectif de minimiser les erreurs d'utilisation. Dépendamment du contexte, il peut s'agir d'autres chercheuses ou chercheurs en informatique, pour qui il faut donc faciliter la prise en main, mais aussi de personnes qui ne sont pas spécialisées en informatique, pour qui il faut proposer des interfaces adaptées.

L'interopérabilité d'une plateforme définit sa capacité à faire coexister et interagir des simulations logicielles avec des composants physiques. Cette interopérabilité peut notamment se manifester par la capacité d'une plateforme à établir une communication efficace entre une partie purement logicielle et une autre intégrant des capteurs physiques, permettant ainsi une

interaction harmonieuse entre ces deux types de simulation.

La surveillance et la journalisation d'une plateforme représentent sa capacité à monitorer et à enregistrer les informations relatives aux simulations effectuées, afin de les rendre disponibles pour l'analyse.

La complexité est liée au nombre de composants pris en compte par la plateforme, et à la façon dont ils interagissent. Elle peut aussi émerger de la complexité de chaque composant individuellement.

L'ouverture correspond au fait de rendre accessible l'intégralité du code source d'une plateforme, selon les règles d'une licence libre, comme par exemple GPLv3 [14]. C'est un gage de qualité puisque cela permet à la plateforme de progresser au-delà du cercle des premiers créateurs.

Chacune de ces caractéristiques représente une qualité qu'une plateforme de test en cybersécurité industrielle peut avoir ou non. Elles permettent d'évaluer la qualité de ces plateformes, même si ce n'est pas une science exacte. C'est aussi et surtout un outil utile pour voir quelles sont les priorités de ces plateformes, en fonction des métriques qu'elles mettent en avant et choisissent d'intégrer au maximum.

Présentons maintenant les plateformes de cybersécurité maritimes existantes.

2.1.1 Grace, par Fathom 5

L'entreprise Fathom 5 a développé une plateforme nommée Grace [15], d'après l'informaticienne et amirale de la marine Grace Hopper, inventrice du premier compilateur de l'histoire. Cette plateforme a des objectifs de recherche, mais aussi de formation. En effet, elle a été utilisée lors de la DEFCON [16] de 2021, afin de participer à augmenter le niveau de connaissance et d'intérêt des participants pour le domaine de la sécurité maritime. Enfin, cette plateforme a aussi pour vocation d'être utilisée dans le cadre de la formation de membres d'équipages afin de les sensibiliser aux enjeux de cybersécurité et de leur donner les bons réflexes.

Grace se présente sous la forme de six modules indépendants mais interconnectables, chacun se focalisant sur un sous-système d'un navire. Les modules sont les suivants :

- Propulsion et direction
- Navigation
- Sécurité incendie
- Technologies informationnelles
- Ballasts
- Aviation

Chacun de ces modules est indépendant, et permet donc d'étudier le sous-système de façon autonome. Cependant, ils peuvent aussi être reliés pour étudier le système à plus grande échelle, ce qui permet une très grande adaptabilité en fonction des cas d'usage. Les sources de cette plateforme sont entièrement fermées, et il est donc difficile de trouver des informations sur son fonctionnement interne.

Néanmoins, de photos publiques, on peut observer que les modules cherchent à reproduire très fidèlement les interfaces utilisateur d'un véritable navire, ce qui entre en cohérence avec l'objectif d'utilisation dans le cadre de la formation de professionnels du maritime.

Finalement, Fathom 5 semble mettre l'accent sur la fidélité, l'évolutivité, la flexibilité, la modularité, la convivialité, mais sans s'intéresser à la reproductibilité et à l'ouverture.

2.1.2 MaCySTe

Le projet MaCySTe (pour Maritime Cyber-Security Testbed), a été présenté pour la première fois en 2023 dans un article de SoftwareX [17]. C'est un projet de l'université de Gênes, porté par Stefano Musante, Giacomo Longo, Alessandro Orlich, et Enrico Russo. MaCySTe revendique deux objectifs principaux : tester des solutions de recherche, et être utilisé dans le cadre de formations pour les professionnels du milieu maritime.

Le projet entier est libre, et donc accessible sur GitHub [18]. Il s'agit purement d'un outil de simulation, qui ne repose sur aucun élément physique autre qu'un ordinateur.

La simulation du pont (visuel des autres navires, radar, et autres interfaces de pilotage du navire) est effectuée par une version modifiée d'un autre outil libre nommé Bridge Command [19]. L'ECDIS (Electronic Chart Display Information System, ou système électronique d'affichage de cartes et d'informations) utilisé est lui aussi open-source, il s'agit d'OpenCPN [20] (Open Chart Plotter Navigation). Un ensemble d'autres outils libres comme OpenPLC, OpenSearch, ou Docker, ainsi que du code développé pour l'occasion, permet de simuler les différents composants du navire, leurs interactions numériques, et leurs comportements physiques.

Le gouvernail est la partie la plus travaillée, en simulant un système de contrôle hydraulique à huile, avec les pompes, vannes, réservoirs, pour le côté physique, et plusieurs PLCs qui communiquent via le protocole MODBUS pour le côté données.

En conclusion, MaCySTe est un outil de recherche qui place l'ouverture, la reproductibilité et le coût-efficacité au cœur de sa stratégie.

2.1.3 Plateforme de Naval Group

Le Laboratoire Cyber Naval de Naval Group a développé un banc d'essai physique pour la recherche en cybersécurité maritime. Créée par les chercheurs Franck Sicard, Estelle Hotellier et Julien Francq, cette plateforme vise à fournir un environnement réaliste pour étudier les menaces de cybersécurité dans les systèmes navals. Elle a été présentée dans un article publié en 2022 [21].

Le banc d'essai est une représentation à échelle réduite d'un navire de guerre, implémentant quatre sous-systèmes principaux :

— Direction : Contrôle les gouvernails du navire
— Énergie : Gère l'approvisionnement en carburant et les opérations de réservoir
— Artillerie : Contrôle une tourelle de canon miniature
— Propulsion : Gère les hélices du navire

Cette plateforme se démarque par son utilisation extensive de composants industriels (PLC, IHM, alimentation, etc.) réels, issus de fabricants comme Siemens et Schneider Electric. Ce choix permet un réalisme accru, de par les failles potentiellement présentes sur ces composants, et de par les protocoles propriétaires qu'ils intègrent nativement.

Le réseau de la plateforme est découpé entre réseaux industriels et bus de terrain, avec des protocoles tels que Modbus RTU, Modbus TCP, Profinet ou Profibus. Cette grande variété n'est pas forcément réaliste, mais permet de tester de nombreux éléments sur une seule plateforme.

Comparée à d'autres bancs d'essai de cybersécurité maritime, cette plateforme se distingue par sa mise en œuvre physique permettant une fidélité, ainsi que par son utilisation de composants réels et divers plutôt que de la simulation. Ce choix se fait au détriment de l'ouverture, du coût-efficacité et de la reproductibilité.

2.2 Mécanismes de détection d'intrusions pour le maritime

Dans cette partie, nous présentons certains articles de recherche présentant des méthodes de détection d'intrusions testées spécifiquement sur certains protocoles dédiés au maritime.

A. Amro et al [22] ont présenté en 2022 un IDS qui analyse les messages NMEA-0183. Leur méthode repose à la fois sur des analyses fréquentielles et sur des règles. L'analyse fréquentielle évalue la quantité de messages transmis, et permet de détecter lorsque certains messages sont perdus, ou à l'inverse, lorsque des messages non prévus sont introduits dans le système. Ce fonctionnement est très efficace sur des systèmes industriels, car les différents

appareils communiquent de façon très régulière. Malheureusement, la fréquence est spécifique à chaque système, ce qui implique une phase d'apprentissage avant de pouvoir être activé. De plus, des attaques avancées peuvent remplacer des messages plutôt que de les supprimer ou d'en rajouter, ce qui n'est pas détectable par cette méthode. C'est pour cela que les auteurs ajoutent aussi un certain nombre de règles pré-établies, qui vérifient que le contenu des messages reste dans un certain cadre. Pour chaque valeur numérique transmise par le système, une plage de valeurs crédibles va être établie, et toute valeur sortant de ce cadre déclenchera une alerte. Certaines de ces plages de valeurs sont constantes, telles que pour des constantes physiques comme la pression, d'autres doivent être adaptées au système, comme la vitesse de rotation du moteur. Cette technique a comme désavantage de forcer à établir pour une bonne partie des données une limite spécifique au navire protégé, ainsi que le fait que des attaques avancées pourraient générer des données crédibles qui ne sortiraient pas des plages spécifiées.

En 2021, C. Boudehenn et al [23] ont proposé une méthode de *Machine Learning* pour détecter les anomalies dans les données GPS dans des trames NMEA-0183. La méthode utilisée est One-Class Support Vector Machine, (OC-SVM), qui est une méthode fréquemment utilisée en détection d'intrusion. Dans un hyperespace représentant les données transmises, OC-SVM va déterminer une zone à laquelle appartiennent les données légitimes. Toute nouvelle donnée qui n'appartiendra pas à cette zone sera donc considérée comme une attaque. L'un des grands avantages de cette méthode est qu'elle ne nécessite pas d'exemple d'attaque, puisqu'elle se contente "d'entourer" la zone des messages légitimes. De plus, c'est un algorithme très efficace qui ne demande pas énormément de données d'entrée, ni de puissance de calcul. Ainsi, les auteurs ont entraîné leur modèle sur seulement trois heures de données légitimes, et le font tourner sur une Raspberry Pi.

Deux des canaux de communication de l'extérieur vers le réseau intérieur d'un navire sont l'AIS et le radar. W. C. Leite Junior et al se sont donc intéressés à l'exploitation de ces canaux pour déclencher un maliciel précédemment installé dans le navire à la détection de ces commandes.

En 2023, W. Liu et al [24] ont développé un système de détection d'intrusion qui utilise de l'apprentissage fédéré pour assurer la confidentialité de chaque source de données, tout en permettant d'entraîner leur modèle sur de grandes quantités de données. Cependant, en l'absence de données spécifiques au domaine maritime, les auteurs ont fait leurs expériences sur le jeu de données NSL-KDD, qui est générique à toute la cybersécurité.

Avec l'idée de détecter des attaques, ainsi que leurs chemins de propagation, N. Pelissero et al utilisent la théorie des graphes pour modéliser et suivre les relations entre les différentes

variables physiques, systèmes physiques, et systèmes numériques. Ce travail a donné lieu à trois articles, [25], [26], et [27].

Une autre proposition est celle de Morten C. N. en 2021 [28]. Celle-ci repose sur une analyse de comportement en croisant les données de la position du navire, de sa vitesse, et de sa direction, pour s'assurer que toutes les données sont cohérentes. Les auteurs génèrent leurs propres données grâce à plusieurs Raspberry Pi qui simulent les différents capteurs nécessaires.

G. L. Babineau et al ont travaillé sur une façon d'utiliser la redondance déjà présente dans de nombreux systèmes critiques, dont les navires, pour détecter des anomalies et des attaques. Les auteurs présentent leurs travaux sur un système de votes entre les différentes redondances d'un switch dans un article [29] de 2012.

De cette revue de littérature, on peut conclure que la recherche sur des solutions de détection d'intrusions spécifiques au domaine maritime est relativement active. Cependant, ces études sont souvent limitées à un protocole spécifique de l'ensemble de l'écosystème d'un navire, et souffrent du manque de données publiquement disponibles (données réelles comme synthétiques). De plus, ces recherches restent aujourd'hui expérimentales, et il ne semble pas exister de système de détection prêt à l'usage, ni même de guide clair et faisant consensus sur comment mettre en place un système de détection dans un navire.

CHAPITRE 3 LA PLATEFORME

Dans cette partie, nous présenterons en détail les différents éléments de la plateforme, les différents appareils, les programmes exécutés dessus, l'architecture du réseau, et les protocoles qui sont utilisés.

3.1 Présentation générale

La plateforme peut être simplifiée en trois groupes d'éléments, comme présenté dans la figure 3.1 :

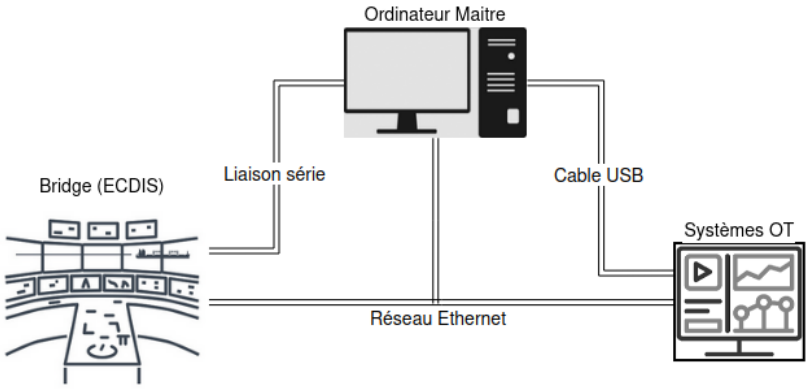

FIGURE 3.1 Schéma très simplifié de la plateforme

— Le système OT représente la partie émulée de la plateforme. Des trois, c'est la partie la plus complexe, puisque c'est celle qui contient le plus d'éléments différents. Les systèmes émulés sont au nombre de deux : la propulsion et le VDR (Voyage Data Recorder, où enregistreur de donnée de trajet, soit l'équivalent de la boite noire pour un navire).

— Le pont représente l'interface de contrôle des humains sur le navire. D'autres interfaces sont aussi présentes au sein du système OT, mais elles ont plus vocation à être utilisées par des techniciens ou lors d'évènements spécifiques. Le pont comprend un Electronic Chart Display and Information System (ECDIS), ainsi que des commandes de pilotages.

— L'ordinateur maître a plusieurs rôles. Il va simuler toutes les données que le système OT ne peux générer lui-même, c'est-à-dire la direction, la position GPS, et les données

météos. D'autre part, il va aussi servir de centre de gestion pour toute la plateforme, en étant le principal poste de travail du projet. Enfin, pour la partie attaque, il agit aussi comme serveur de commande et de contrôle des attaques.

Deux réseaux principaux relient ces éléments :

— Une liaison série (RS232) relie directement la machine maître et l'ECDIS. A travers le protocole NMEA-0183, elle transmet les données météos, GPS, et de navigation.

— L'autre est un réseau Ethernet (RJ45), qui passe par deux switchs. Ce réseau remplit plusieurs rôles et supporte différents protocoles, aussi nous détaillerons chacun d'entre eux ultérieurement.

Un câble USB permet aussi une communication entre l'ordinateur maître et les moteurs, au sein des systèmes OT. Cette communication est extra-diégétique.

3.2 Pont

3.2.1 L'ECDIS

L'ECDIS est une carte interactive qui affiche la position du navire, des navires environnants, ainsi que de nombreuses données (météorologiques, topographiques, de vitesse et de direction, etc.). Présente sur le pont, c'est un des éléments principaux d'aide à la navigation.

Pour notre plateforme, deux logiciels différents ont été utilisés : Le premier est OpenCPN [20], une version open-source. Comme nous l'avons évoqué dans la section de l'état de l'art, cet outil est régulièrement utilisé, notamment du fait de sa gratuité et de son caractère libre. Cependant, il souffre d'un certain nombre de limitations, notamment dans les types de données qu'il peut traiter, et n'est pas utilisé dans l'industrie.

C'est pour cette raison que nous avons préféré utiliser TimeZero, [30] un outil commercial. Cet outil est plus réaliste, et utilisé dans l'industrie. La figure 3.2 présente l'interface. Sur le bandeau de droite, on peut retrouver les différentes données, récupérées par les capteurs répartis sur le navire. Au centre, la figure de navire rouge représente la position actuelle du navire. Si d'autres navires sont communiquent par AIS, ils sont représentés sous la forme de les flèches bleues. Enfin, le chemin parcouru est représenté par une traînée rouge derrière le navire.

3.3 Ordinateur maître

La machine maître contient différents composants, pour ces différents rôles, chacun isolés dans des conteneurs dockers.

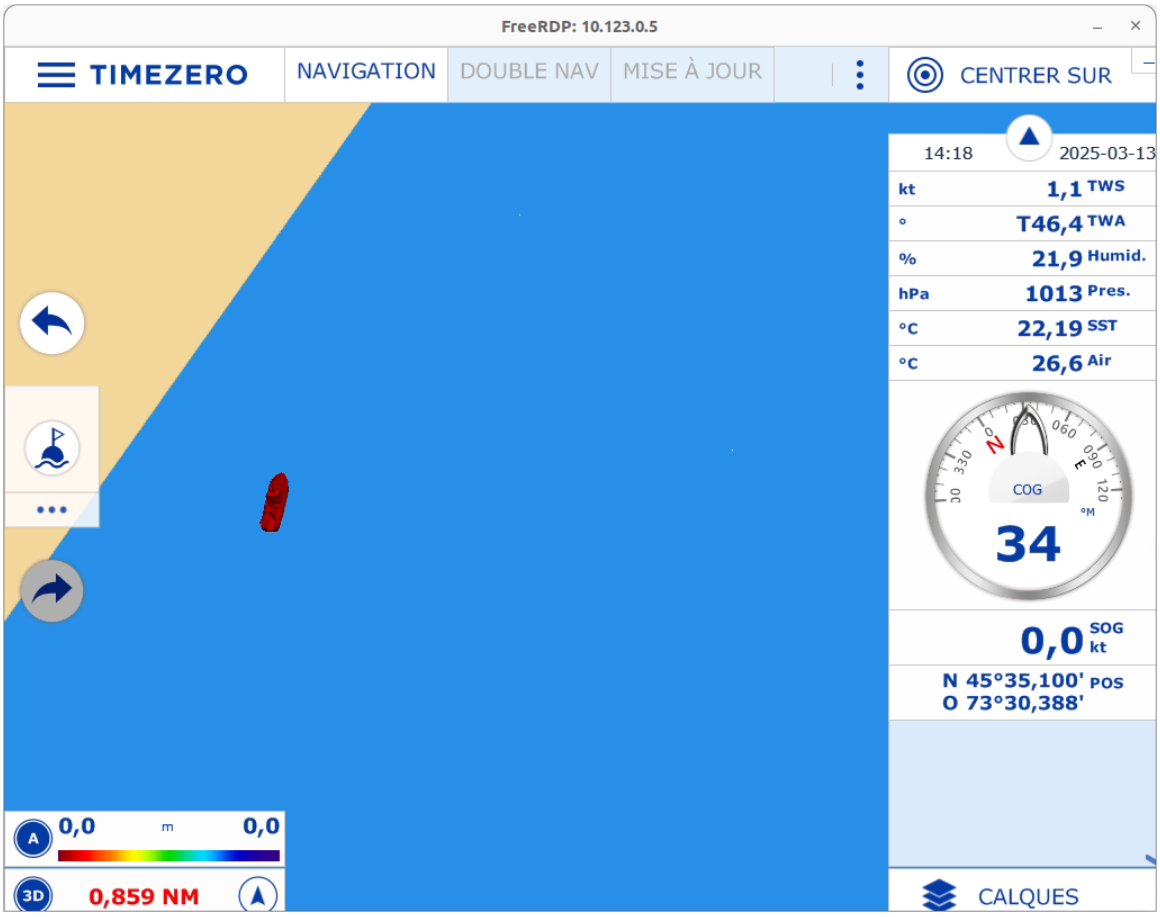

FIGURE 3.2 Présentation de l'interface de l'ECDIS Time Zero.

Docker [31] est un logiciel open source qui permet de conteneuriser une application, c'est a dire de générer un sous-système d'exploitation avec l'application et les dépendances dont elle a besoin. Ce système est relativement séparé du système principal de l'ordinateur dit hôte, ce qui permet d'assurer une grande stabilité pour l'application, indépendamment du système hôte. Docker permet aussi d'automatiser et de faciliter le partage de cette application.

Pour simplifier la plateforme, il a été choisi de rassembler les rôles de simulation, de gestion de la plateforme et de direction des attaques sur la même machine. Pour ce faire, Docker a été utilisé pour s'assurer qu'il n'y ait pas d'interférences entre les différents codes, et permettre de les sauvegarder et d'en faire de la gestion de version grâce à git [32].

Les éléments de cette machine qui concernent la gestion des attaques seront détaillés dans le quatrième chapitre, dans des parties spécifiques.

3.3.1 Administration

Nous voulions pouvoir gérer un maximum d'éléments de la plateforme de façon centralisée, pour faciliter son utilisation. Pour ce faire, nous avons conçu une interface web grâce à Flask, permettant de gérer les différents paramètres d'un scénario.

Ce site web contient trois fenêtres. La première contient un texte introductif sur le fonctionnement des deux autres. La seconde permet de gérer les scénarios d'attaques.

La dernière interface permet de paramétrer la simulation : il est notamment possible d'y choisir les caractéristiques physiques du navire (poids, moment d'inertie, coordonnées GPS initiales, puissance des moteurs...), et un certain nombre d'autres paramètres, comme le port série pour la communication NMEA-0183 avec l'ECDIS. Une fois ces valeurs entrées, cette fenêtre permet aussi de démarrer la simulation, ce qui aura pour effet d'ouvrir la fenêtre de contrôle du simulateur présenté en 3.3.2. Enfin, il est aussi possible de gérer en temps réel différents paramètres, tels que le risque de surchauffe des moteurs, les conditions météos, etc.

3.3.2 Simulateur

La simulation est faite par un code python qui est interprété dans son propre docker. Le code python est démarré par le site web d'administration. Suite à ça, les données commencent à être générées. Pour ce faire, lors du démarrage du docker, il ne démarre qu'un service SSH, qui va lui-même se charger de lancer la simulation lorsque réclamé par l'administration.

Cette application inclut aussi une interface graphique avec deux curseurs pour diriger le navire (puissance des moteurs et angle du gouvernail). Cette interface graphique, présentée par la figure 3.3, est accessible depuis un navigateur sur la machine hôte, ou depuis une autre machine sur le réseau Ethernet, comme celle contenant l'ECIDS, qui représente donc le pont.

Les données générées peuvent être rangées en deux catégories : les données correspondant au déplacement du navire - comme la direction ou la position GPS ; et les données météorologiques - comme le vent, la température et la pression.

Données du déplacement du navire.

Pour générer ces données, plusieurs paramètres sont déterminés avant la simulation, depuis le site web d'administration, comme présenté dans le tableau 3.1.

La seconde donnée d'entrée est la vitesse de rotation immédiate des moteurs. Cette vitesse est évidemment accessible sur le réseau Ethernet reliant les systèmes OT, mais ces données risquent d'être modifiées par des attaques. C'est pourquoi il fallait un canal de communication

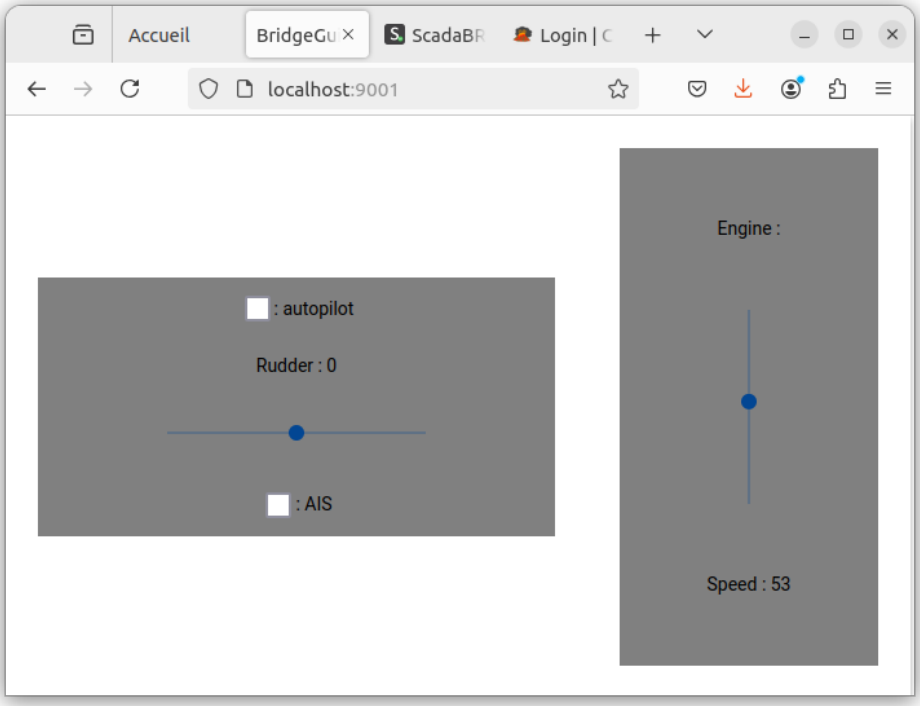

FIGURE 3.3 Interface de pilotage du navire

extra-diégétique entre le simulateur et les moteurs. Il aurait été possible de passer par le réseau Ethernet qui relie déjà ces différents éléments, mais cela aurait impliqué une communication qu'il aurait fallu ignorer, et ce à la fois dans toutes les attaques et dans tous les outils de détections d'intrusion. C'est pourquoi il a été choisi d'opter pour un autre support de communication, en branchant directement un câble USB entre les deux.

Pour la vitesse du navire, on considère que deux forces s'appliquent :

— La première est celle provoquée par les moteurs, que l'on modèlise comme le produit de la puissance maximale des moteurs (P) et le coefficient représentant le niveau d'activation des moteurs, noté c, et variant entre 0 et 100, et venant directement des moteurs via la connexion USB décrite précédemment. On suppose cette force toujours exactement alignée avec la direction du mouvement.

— La seconde est une force représentant les frottements, aussi bien avec l'air qu'avec l'eau. Elle est définie par le produit du coefficient de frottement dynamique α et de la vitesse du navire v. On la suppose toujours exactement opposée à la direction du mouvement.

Nom de la caractéristique	Valeur par défaut
Latitude initiale	45.585°
Longitude initiale	−73.50647°
Masse du navire (m)	200T
Coefficient d'inertie (J)	10
Coefficient de frottement dynamique (α)	30
Coefficient de frottement de rotation (β)	30
Puissance maximale des moteurs (P)	4

TABLEAU 3.1 Caractéristiques physiques paramétrables du navire.

En notant a l'accélération du navire, et m sa masse, on obtient alors le bilan suivant :

$$ma = Pc - \alpha v \tag{3.1}$$

Pour ce qui est de sa vitesse angulaire noté θ_v, on considère que deux moments s'appliquent :

1. Le premier est celui provoqué par le gouvernail, modélisé comme un seul facteur directement égal à la position du gouvernail, entre -50 et 50.
2. Le second est celui provoqué par les frottements, qui est définit par le produit du coefficient de frottement de rotation β et de la vitesse angulaire.

En notant θ_a l'accélération angulaire et J le moment d'inertie du navire, on obtient le bilan suivant :

$$J\theta_a = R - \beta\theta_v \tag{3.2}$$

En intégrant les équations précédentes, on peut donc obtenir l'approximation de la vitesse linéaire et vitesse angulaire après un temps dt considéré :

$$v(t + dt) = v(t) + \frac{Pc - \alpha v(t)}{m} dt \tag{3.3}$$

$$\theta_v(t + dt) = \theta_v(t) + \frac{R - \beta\theta_v(t)}{J} dt \tag{3.4}$$

Ces deux équations nous permettent donc de calculer la vitesse linéaire et angulaire de notre navire. Le temps dt est paramétrable a travers l'interface de configuration de la simulation, mais est considéré comme étant égal à $0,5s$ par défaut, pour assurer des temps de réaction adapté aux expérimentation en laboratoire ou aux démonstrations.

Données météorologiques

Les conditions météorologiques sont un grand facteur de risque pour des accidents, mais aussi pour les cyberattaques. En effet, de mauvaises conditions météorologiques peuvent grandement réduire le champ de vision de l'équipage, les rendant d'autant plus dépendants des systèmes informatiques. C'est donc dans ces conditions que l'impact potentiel d'une cyberattaque est le plus grand, c'est pourquoi nous avons décidé d'inclure deux scénarios au simulateur : temps clair et mauvaise météo.

Une partie du simulateur est donc dédiée a la génération de six types de données différentes, qui viendraient directement d'outils de mesures embarqués dans le navire. Ces données sont générées suivant une loi normale. Les paramètres de cette loi, pour chaque scénario et chaque type de données, sont données dans le tableau 3.2.

3.4 Liaison série - NMEA

Le protocole NMEA-0183 est un protocole encore largement utilisé dans le domaine maritime pour interconnecter les capteurs et les actionneurs. Il s'agit d'un protocole point-à-point, c'est-à-dire qu'il permet la communication entre deux machines. Ainsi, des agrégateurs sont souvent utilisés : ils possèdent plusieurs ports par lesquels ils se connectent aux différentes machines, et transmettent le tout vers l'ECDIS sur une seule connexion. Pour les messages de l'ECDIS vers les autres appareils, ils peuvent attribuer un canal en fonction du type de message.

Le protocole utilise un format de données en texte American Standard Code for Information Interchange (ASCII), ce qui le rend relativement simple à comprendre et à implémenter. Les messages sont structurés en phrases, chacune commençant par un symbole dollar ; un identifiant de la machine source ; un identifiant de type de message ; suivi de plusieurs champs de données séparés par des virgules. Chaque champ contient des informations spécifiques telles

	μ_{normal}	$\mu_{mauvais}$	σ	Unité
Humidité	20	78	2	Pourcentage
Température de surface de la mer	22	12	0,5	Degrés Celcius
Direction du vent	45	45	2	Degrés, par rapport au Nord
Force du vent	1	7	0,3	Nœuds
Pression	101 300	101 300	50	Pascal
Température de l'air	27	17	0.2	Degrès Celcius

Tableau 3.2 Données transmises par la sation météo interne.

que la latitude, la longitude, l'altitude, et l'heure (généralement en Coordinated Universal Time (UTC)). Cette structure standardisée permet aux systèmes de navigation de traiter et d'afficher les données de manière cohérente, garantissant ainsi une communication efficace et fiable entre les différents appareils.

Voici un exemple de trame NMEA-0183, où j'ai simplement réduit le nombre de chiffres significatifs pour la rendre plus lisible :

$$\$IIMWV, 47.7, T, 0.96, N, A * 0E\text{<cr><lf>}$$

Le tout premier caractères, $, sert uniquement a marquer le début d'une trame. Les deux suivants, II signifient *Integrated Instrumentation*, ce qui veut dire que les données viennent d'appareil situés directement dans le navire. Autrement dit, la source des données est une petite station météo intégrée au navire.

Ensuite, les caractères MWV décrivent le type de trame envoyée. Ce champ permet de déchiffrer tout le reste de la trame puisque les données sont structurées différement pour chaque type de trames. Une trame MWV est une trame contenant la vitesse du vent et son angle.

Enfin, les données se présentent, séparées par des virgules. Les deux premières (47,7 et le T) communiquent l'angle du vent. La lettre T, pour True, signifie que l'angle est donnée par rapport au nord, et pourrait être remplacer par un R (pour Relative), qui signifirait que l'angle donné l'est par rapport a la proue du navire. Enfin, la valeur numérique (47,7 ici), est l'angle du vent, donné en degrés entre 0 et 360.

Le deuxième groupe de donnée rassemble le 0,96 ainsi que la lettre suivante, N, qui donnent la vitesse du vent. Encore une fois, le N donne une information sur la façon de lire le nombre précédent, ici l'unité. Un N signifie que la vitesse est donnée en nœuds, tandis qu'un K signifirait des kilomètres, et un M des mètres.

Finalement, le A permet simplement de transporter un statut. S'il y a un A à cette emplacement, la trame est valide, mais tout autre lettre signifierait qu'il y a un problème. Enfin, on arrive à la fin du champ de données, donc on revient à des champs présents quel que soit le type de message. *0E est une somme de contrôle pour vérifier l'intégrité des données transmises, et les caractères *cariage return* ($< cr >$), et *line feed* ($< lf >$), ferment tout simplement la trame.

La National Marine Electronics Association a aussi proposé deux autres versions plus avancées de ce protocole, mais qui ont du mal à s'imposer sur le marché.

La première est NMEA-2000, qui a été publiée en 1997. Elle est souvent abrégée par les

acronymes NMEA2K, voir N2K. C'est une amélioration de la version 0183 sur de nombreux points, notamment le débit (250kb/s contre 38.4kb/s) et le nombre d'appareils sur un même lien. La norme a été pensée pour fonctionner sur un bus Controller Area Network (CAN), ce qui permet de passer d'une communication point-à-point (deux appareils), à un bus de communication reliant jusqu'à 252 appareils. Elle abandonne aussi les caractères ASCII pour un encodage binaire bien plus compact. Néanmoins, du point de vue des questions de sécurité, comme sa prédécesseuse, la norme NMEA-2000 n'apporte aucune protection d'authentification ou de confidentialité.

Si NMEA-2000 est devenue la norme dans les navires de plaisance, les bâtiments commerciaux ont une plus grande inertie d'adaptation et ont tendance à se contenter de la norme NMEA-0183, qui remplit très bien son rôle dans l'architecture actuelle.

La deuxième norme est NMEA OneNet. Annoncée en 2021, cette mise a jour est pensée pour fonctionner au dessus de l'Internet Protocol version 6 (IPv6), via un Local Area Network (LAN) Ethernet, ce qui permet une cohabitation avec d'autres protocoles Internet Protocol (IP) sur le même réseau. Le débit passe alors à un maximum de 10Gb/s, et le nombre d'adresses devient en pratique illimité, grâce à l'IPv6. Les messages restent transmis sous forme binaire, tout comme en NMEA-2000. OneNet introduit néanmoins une option pour chiffrer tous les messages, en plus de la sécurité apportée par l'IPv6 par rapport au bus CAN ou à une liaison série.

NMEA OneNet est une norme prometteuse, que la National Marine Electronics Association prévoit d'élargir, notamment pour inclure de la communication sans fil en sus de l'Ethernet et de la communication entre les navire et les côtes, ce qui est adapté aux enjeux de digitalisation, d'automatisation et de connexion au réseau qui traversent actuellement le milieu maritime. Néanmoins, cette version est bien trop récente pour qu'on puisse envisager qu'elle représente une part de marché importante, même a moyen terme.

Les différences entre les trois protocoles principaux de la National Marine Electronics Association sont résumées dans le tableau 3.3.

Parmi les trois normes, la norme NMEA-0183 reste la plus utilisée, c'est pourquoi c'est celle que nous avons choisi d'utiliser dans notre plateforme. Son rôle est donc de transmettre toutes les données générées par le simulateur à l'ECDIS.

3.5 Systèmes de technologies opérationnelles

Cette partie entre en détail dans le fonctionnement des systèmes de technologies opérationnelles qui ont été développés pour la plateforme. La figure 3.4 présente les différents appareils

Version	NMEA-0183	NMEA-2000	OneNet
Date	1983	1997	2021
Débit	38,4kb/s	250kb/s	10Gb/s
Support	Liaison série	CAN Bus	Ethernet/IPv6
Nombre d'appareils sur un réseau	2	252	Illimité
Sécurité	Absente	Absente	IPv6 + Chiffrage
Niveau d'adoption	Majoritaire	Minoritaire	Rare

Tableau 3.3 Comparaison des différentes normes NMEA

qui composent ce système.

Les liens composés de deux lignes correspondent aux connexions Modbus reliant les différents appareils. Elles représentent les communications effectives qui sont engagées, mais pas les connexions physiques. En effet, chacun de ces éléments est connecté au réseau Ethernet central. Si chaque élément était relié à un réseau différent, cela aurait impliqué un plus grand nombre d'interfaces réseaux pour le PLC et pour le SCADA, et plus de switchs pour gérer le réseau. De fait, cela aurait augmenté les coûts et la complexité de la plateforme. Le gain en réalisme ne nous a pas semblé suffisant pour justifier cet engagement.

Enfin, les deux flèches avec un trait simple représentent des connexions à travers des câbles USB, qui sont présents sur la plateforme mais pas dans le scénario, car il s'agit de connexions extra-diégétiques, correspondant à des données utilisées pour la simulation du déplacement physique du navire, comme présenté en section 3.3.2.

3.5.1 Modbus

Avant de parler des composants en eux-mêmes, présentons le protocole qui les relie tous, Modbus.

Le protocole Modbus a été développé en 1979 par la société Modicon (aujourd'hui partie de Schneider Electric) pour établir une communication entre leurs PLCs. Il s'agit de l'un des premiers protocoles de communication industriels standardisés et est devenu, au fil des décennies, l'un des protocoles les plus répandus dans l'environnement industriel. Sa popularité s'explique par sa simplicité technique, sa nature ouverte (il est devenu un standard libre en 2004, géré par l'organisation Modbus-IDA), et sa facilité d'implémentation. Le protocole Modbus a survécu à l'évolution technologique grâce à sa capacité d'adaptation, passant d'une communication série (Modbus RTU ou ASCII) à des implémentations sur Ethernet et TCP/IP (Modbus TCP).

Peu importe sa version, le protocole repose sur une architecture maître-esclave où un dispositif

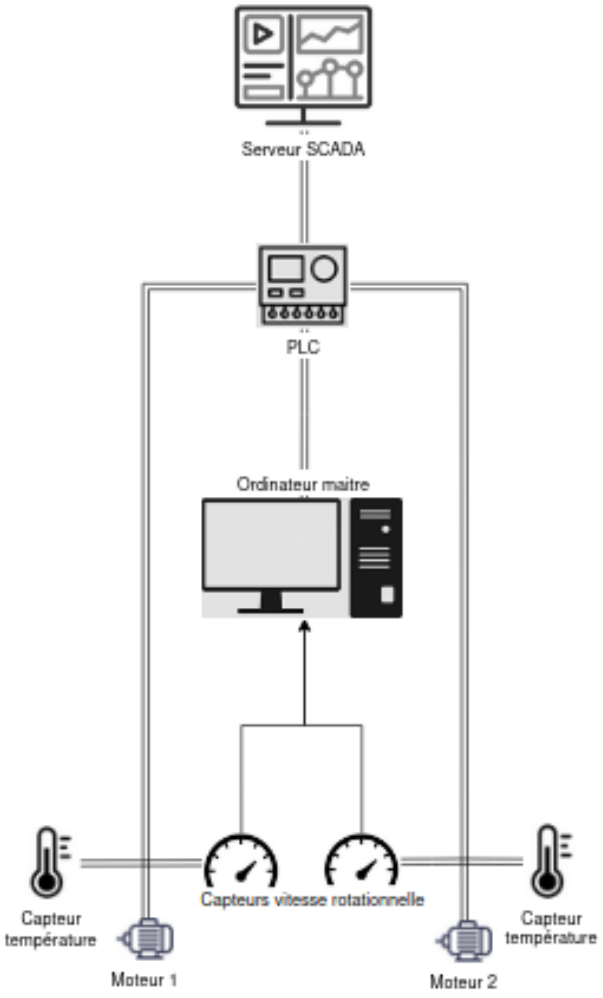

FIGURE 3.4 Schéma des différents composants du système d'OT.

maître communique avec un ou plusieurs esclaves. Les données sont séparées en fonction de leur mode d'accès (lecture seule, par exemple pour un capteur, ou lecture et écriture, par exemple pour une commande). Elles sont aussi séparées en fonction de leur taille, soit d'un seul bit, soit par mots de 16 bits. Il y donc un total de quatre tables de données.

Chaque requête du maître comprend l'adresse de l'esclave, un code de fonction qui spécifie l'opération à effectuer, les données éventuelles, et un mécanisme de vérification d'erreur. La réponse contient les mêmes champs, mais les données changent évidemment.

La différence entre le Modbus "classique" et le Modbus TCP est le support des données. Modbus RTU et Modbus ASCII fonctionnent sur un câble série, et sont donc des communi-

cations points à points. Modbus TCP, comme son nom l'indique, fonctionne via TCP, donc à travers le protocole IP. C'est donc un outil bien plus polyvalent, qui peut utiliser un réseau (Ethernet ou sans fil) pré-existant et cohabiter avec d'autres protocoles.

Techniquement, Modbus TCP encapsule des messages Modbus classique dans des trames TCP, en ajoutant un en-tête spécifique. Cet en-tête contient notamment un identifiant de transaction qui doit être répliqué dans la réponse, la taille du paquet Modbus encapsulé, et un identifiant pour le serveur.

Le protocole Modbus n'embarque aucune sécurité nativement. Une version exploitant le fonctionnement de TLS nommée Modbus TCP Security existe, mais son utilisation reste relativement réduite à cause des pertes de performance entraînées par le chiffrement. En effet, dans un écosystème ou des requêtes sont envoyées souvent plusieurs fois par seconde, la surcharge engendrée par le chiffrement est conséquente.

3.5.2 SCADA

En haut de la chaîne de commande se trouve le serveur SCADA. Il est simulé par le logiciel libre ScadaBR [33], qui tourne sur une Raspberry Pi 5. Ce serveur est directement relié au PLC maître des moteurs, et permet donc d'accéder aux informations qu'il contient.

Il rend aussi disponible une interface humain-machine (IHM) présenté en figure 3.5, qui présente en temps réel et de façon réaliste les informations importantes de ce système. Cette IHM présente deux côtés symétriques, un pour chaque moteurs, qui présentent les informations remontées depuis le capteur de vitesse et le capteur de température. De plus, une barre permet de baisser la charge sur un moteur, pour gérer individuellement leurs puissances. Enfin, au centre, une barre permet de décider de la vitesse du navire, ce qui court-circuite l'ordre donné par le système de pilotage principal, c'est à dire l'interface web du simulateur présenté en section 3.3.2.

3.5.3 PLC

Notre système PLC est fait avec l'outil libre OpenPLC [34], qui vise depuis 2014 à rendre possible la simulation de PLCs sur différents ordinateurs. Dans notre cas, nous utilisons une Raspberry Pi 4 comme support.

Ce PLC est connecté d'un côté au serveur SCADA, qui récupère toutes les données pertinentes pour leur analyse centralisée. De l'autre côté, il est connecté aux deux appareils représentant les capteurs et les actionneurs, à partir desquels il va lire des données et écrire des ordres pour la puissance des moteurs. Enfin, le PLC est aussi connecté au simulateur, puisque c'est

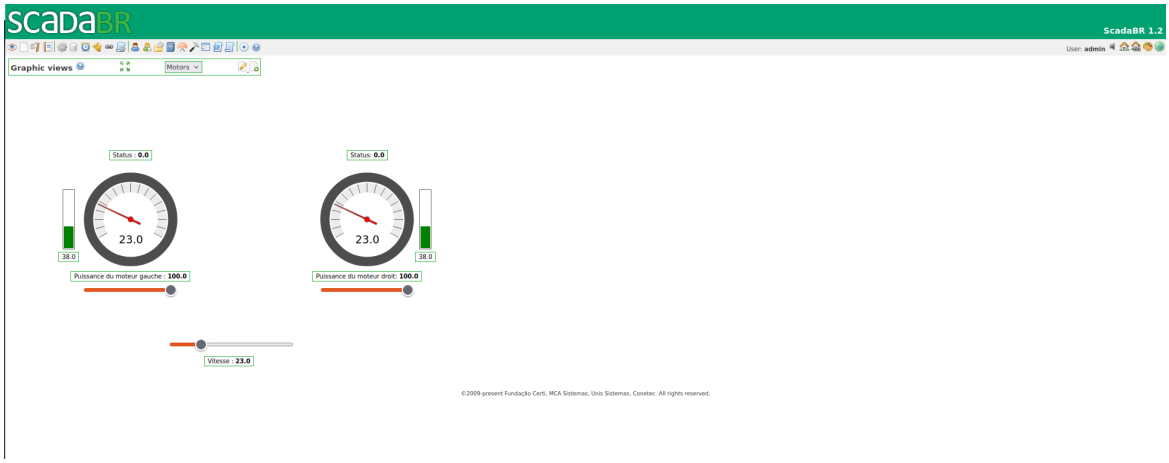

FIGURE 3.5 Capture d'écran de l'interface de présentation des systèmes de propulsion.

lui qui représente le pilotage du navire, donc qui détermine la vitesse demandée.

Le code de ce PLC est rédigé en Structured Text (ST), c'est à dire le mode de programmation le plus proche de la programmation classique. C'est ce dernier notamment qui s'occupe de la répartition de la charge entre les deux moteurs.

3.5.4 Capteurs et actionneurs

La ligne de contrôle d'un moteur est simplifiée à trois éléments : un capteur de température, un capteur de vitesse, et le moteur en lui même. Même si dans la réalité beaucoup plus de capteurs et d'actionneurs peuvent être présents, ces trois là couvrent relativement bien les différentes fonctions existantes.

Le moteur et le capteur de vitesse représentent la boucle de rétroaction principale, qui permet au système d'effectuer une tâche et de vérifier que celle-ci est bien effectuée. Le capteur de température, lui, correspond plutôt à un outil de sécurité, qui vise à limiter le risque d'accident, spécifiquement de surchauffe. Ainsi, même si la boucle de rétroaction principale pourrait être beaucoup plus complexe, et si la température est loin d'être la seule métrique à surveiller pour éviter les accidents, ces trois rôles principaux sont représentés dans notre plateforme.

Pour chaque ligne de moteurs les trois éléments sont représentés par le même appareil physique, une Raspberry Pi Pico. Ces dernières sont équipées d'un HAT (Hardware At the Top) leurs permettant de communiquer à travers un câble Ethernet. Elles sont programmées en MicroPython pour recevoir et répondre aux requêtes Modbus, et générer des données réalistes. Notamment, elles incluent une inertie qui fait qu'elles mettent du temps à atteindre la

vitesse demandée.

Le capteur renvoie la vitesse du moteur vers le PLC en Modbus, mais il la communique aussi au simulateur via un câble USB. En effet, il est important d'ajouter une communication extra-diégétique qui s'assure que le calcul du déplacement pour les coordonnées GPS simulées ne se basent pas sur l'une des données qui peut être attaquée. Cependant, il est possible de falsifier les données qui communiquent sur le réseau Modbus, ce qui entraîne des fausses données dans le PLC et dans le SCADA, comme nous le verrons dans le second thème. Il est donc important de s'assurer qu'il existe un canal de communication qui ne soit pas attaquable. Il serait possible de l'ajouter au sein du réseau Ethernet, en le filtrant pour pas qu'il ne soit vu par les outils d'attaque ou de détéction automatiques, mais il est plus simple de directement passer par un autre canal. Puisque la Raspberry Pico qui représente le moteur doit être alimentée par un câble USB, et qu'elle peut communiquer à travers nativement, le plus simple est donc de l'alimenter depuis l'unité centrale contenant les éléments de simulation présentés en section 3.3, et de faire communiquer ces informations extra-diégétiques directement par ce câble.

Enfin, chaque moteur a un certain pourcentage de chance de surchauffe. Ce pourcentage est paramétrable pour chaque simulation, depuis l'interface web d'administration présentée en section 3.3.1. Il y a trois modes de fonctionnement, qui peuvent être activés indépendamment pour chaque moteur :

— Sans panne : Ce mode de fonctionnement désactive complètement le risque de panne, permettant de faire des expérimentations sans risque d'éléments imprévus.
— Petit risque de panne : Dans ce cas, la probabilité est calibrée pour qu'une surchauffe arrive dans les 15 prochaines minutes. Ce mode de fonctionnement permet de capturer l'accident dans une expérimentation suivante.
— Grand risque de panne : Dans ce cas, une surchauffe arrivera dans les 30 prochaines secondes. Ce mode de fonctionnement permet de présenter les capacités de la plateforme dans le cadre d'une démonstration.

Lorsqu'un moteur surchauffe, ça température varie autour de 100°, et le moteur se met a tourner très lentement.

3.6 VDR

Le VDR (Virtual Data Recorder), est un système qui enregistre toutes les données possibles du navire afin de les rendre disponibles pour une enquête en cas d'accident. Finalement, le VDR joue un rôle équivalent à celui qu'une boite noire joue dans un avion.

Pour ce faire, il sauvegarde ces données dans deux mémoires différentes, et l'une d'entre elle est installée au sein d'un petit dispositif flottant qui devrait pouvoir être facilement récupéré si le navire sombre. La seconde mémoire reste toujours accrochée au navire.

Les enjeux de cybersécurité autour du VDR sont relativement différents de ceux qui encadrent le reste des appareils du navire, puisqu'il n'a techniquement aucun impact concret sur le navire. Cependant, des adversaires avancés et qui veulent cacher leur attaque devront évidement le prendre en compte. Mais un autre risque spécifique au VDR est la falsification des données par l'équipage et/ou l'armateur. En effet, les données issues du VDR peuvent être utilisées comme pièces a conviction lors de procès ou contre des assurances. Ainsi, les fabricants de VDR se doivent de sécuriser leur système contre leur propre propriétaire.

Les données enregistrées par le VDR sont diverses, et peuvent notamment inclure :

- Des journaux de toutes les communications entre les différents appareils OT
- Des données précises sur l'environnement et le navire (localisation, météo, statut des différents appareils)
- Des journaux des différentes communications avec l'extérieur à travers l'AIS
- Des captures d'écran régulières de l'ECDIS, du radar et/ou d'autres appareils du genre
- Une capture audio voire vidéo constante du pont

Dans le cas de notre plateforme, le VDR est représenté par une Raspberry Pi 5, et les données seront enregistrées dans une base de données SQL. Nous avons inclus un maximum de données possibles, à l'exception de la capture audio du pont, qui n'aurai pas grand intérêt pour nos applications de recherche et représentait un grand investissement de bande passante par rapport à notre réseau.

3.7 Conclusion

Notre plateforme est donc une plateforme axées sur les technologies opérationnelles, qui choisit l'émulation plutôt que la simulation virtuelle pour représenter au mieux les systèmes réels. Les principales caractéristiques que l'on met en avant sont les suivantes :

- La fidélité, car il s'agit de la caractéristique la plus importante pour s'assurer que les solutions que l'on étudie sont crédibles et réalistes.
- L'évolutivité, car la plateforme est pensée pour être utile dans la durée, et inclure de nouveaux systèmes sur lequel le laboratoire souhaite travailler.
- Le coût-efficacité, pour réussir à mettre en place le maximum avec les moyens à disposition.

CHAPITRE 4 DÉPLOIEMENT D'ATTAQUES DANS LA PLATEFORME

Ce chapitre se concentrera sur la mise en place de deux catégories d'attaques contre le navire simulé. La première vise l'ECDIS, et donc les informations disponibles à l'équipage mais sans impact direct sur les opérations du navire. La seconde catégorie vise la chaîne de commande des moteurs, impactant donc les informations qui en remontent et commandes qui leurs sont envoyées.

Ces deux types d'attaques représentent deux possibilités, et ont pour objectif d'être les plus génériques possibles, visant à présenter au maximum les types d'impacts que l'adversaire pourrait utiliser. De plus, la plateforme est pensée pour que l'on puisse y déployer d'autres attaques afin de contenir des scénario toujours plus diversifiés, en accord avec notre objectif d'évolutivité.

4.1 Caractérisation des adversaires

Cette section vise à caractériser le profil type des acteurs malveillants identifiés dans les scénarios d'attaques étudiés, en analysant leurs motivations, leurs connaissances et leurs capacités techniques.

Notre analyse se concentre sur un profil spécifique d'adversaire commun aux deux catégories d'attaques : un acteur possédant une connaissance approfondie du navire ciblé et de son écosystème humain, de compétences techniques en cybersécurité moyennes et d'une bonne connaissance du domaine maritime.

4.1.1 Compétences dans le domaine maritime

Les connaissances dans le domaine maritime peuvent provenir de différentes sources :

- Documentation technique (publiquement accessible ou précédemment volée), telle que les plans de construction ou les manuels d'utilisation des équipements
- Expérience professionnelle antérieure à bord du navire ou sur des navires similaires
- Informations obtenues par l'intermédiaire de contacts internes à l'équipage ou à la compagnie maritime

Cette connaissance approfondie permet aux adversaires de supprimer la phase de reconnaissance active au sein du navire, ce qui empêche de les détecter à cette étape. On considère notamment que les adversaires connaissent déjà les adresses IP et l'architecture réseau du navire.

Il est aussi très fréquent que des adversaires, quel que soit le domaine d'attaque, aient une très bonne connaissance des environnements humains de leur cible. Ceci inclut notamment la connaissance de l'organigramme et des rôles du personnel à bord, et une certaine familiarité avec les procédures opérationnelles standards et les routines de l'équipage. En effet, ces informations non techniques sont souvent perçues comme beaucoup moins critiques qu'elles ne le sont. Il est donc fréquent de les trouver publiquement sur des sites d'entreprise, ou au moins relativement facile de les avoir lors d'interactions sociales qui passeront inaperçues, par exemple lors de conventions de professionnels.

Cette connaissance de l'écosystème humain permet aux adversaires d'exploiter le facteur humain comme vecteur d'attaque privilégié, notamment par des techniques d'ingénierie sociale, comme vu par exemple dans les sections 4.3.1 et 4.4.1.

4.1.2 Compétences dans le domaine de la cybersécurité

Nos scénarios analysés nécessitent des adversaires disposant de compétences techniques moyennes en matière de cybersécurité. En général, ces acteurs :

- Maîtrisent des techniques d'attaque basiques mais éprouvées (hameçonnage, utilisation de malwares préfabriqués)
- Ne développent généralement pas leurs propres outils d'attaque sophistiqués, mais sont capables d'adapter des outils existants, ou de les paramétrer précisément
- S'appuient sur des vulnérabilités connues plutôt que sur la découverte de failles zero-day

Néanmoins, ces adversaires savent très bien cumuler ces compétences avec leurs connaissances du milieu maritime, afin d'adapter les stratégies et méthodes à un milieu de technologies opérationnelles relativement différentes des méthodes habituelles.

4.1.3 Objectifs de l'attaque

Objectif technique

Les deux catégories d'attaques visent un objectif technique commun : impacter le bon fonctionnement du navire. Cette perturbation opérationnelle peut prendre plusieurs formes :

- Provoquer des dysfonctionnements nécessitant un retour du navire en révision
- Générer des pannes intermittentes difficiles à diagnostiquer
- Altérer le fonctionnement de systèmes critiques pour compromettre la sécurité de navigation
- Dans les cas les plus graves, provoquer des incidents ou accidents potentiellement

dangereux

Ces perturbations peuvent entraîner des conséquences économiques significatives : retards de livraison, coûts de réparation, immobilisation prolongée du navire, atteinte à la réputation de l'armateur.

Motivations sous-jacentes

Ces motivations profondes ne sont pas forcément impactantes pour le déroulé de l'attaque, mais plusieurs raisons peuvent pousser quelqu'un à vouloir effectuer ces attaques. Par exemple, on pourrait penser à un concurrent du domaine qui souhaite mettre à mal l'armateur, espérant sortir lui-même renforcer de l'évènement. On peut aussi envisager des conflits géo-politiques qui pourraient entraîner une sorte de guerre cyber-commerciale. Enfin, un employé mécontent cherchant à se venger de son employeur peut aussi être à l'origine de ces attaques.

4.2 Coordination des attaques

Pour coordonner les attaques, nous utilisons Caldera comme serveur de Contrôle et de Commande (C^2), ainsi qu'une troisième page sur le site web d'administration, pour déclencher des scénarios en un clic.

Caldera est un outil de cybersécurité développé par le MITRE Corporation qui permet de simuler de nombreux scénarios d'attaques grâce à une approche en Tactique Technique et Procédures (TTP) pour décrire le déroulé de l'attaque.

Pour utiliser Caldera comme serveur C^2, il suffit de déployer sur un ordinateur un agent spécifique. Ces agents sont fournis par Caldera, sous la forme de programmes Python, de scripts Powershell ou Bash. Ces agents vont s'occuper de rester en contact avec le serveur central via des balises, et vont pouvoir dérouler un certain nombre d'actions prédéfinies ou des commandes arbitraires sur l'ordinateur infecté.

Les balises de Caldera représentent une évolution sophistiquée du mécanisme de communication agent-serveur. Contrairement aux agents classiques qui maintiennent une connexion persistante avec le serveur C^2, les balises établissent des connexions périodiques et temporaires selon un intervalle configurable, ce qui leur confère une discrétion significative face aux systèmes de détection. Cette approche présente plusieurs avantages stratégiques : d'abord, la réduction de la signature réseau rend leur détection plus complexe pour les outils de sécurité qui surveillent les connexions suspectes ; ensuite, leur faible consommation de ressources système permet de maintenir une présence plus discrète sur l'ordinateur compromis ; enfin, leur

mécanisme de communication asynchrone offre une meilleure résilience face aux instabilités réseau ou aux déconnexions temporaires. Les balises peuvent également être configurés pour utiliser différents protocoles de communication (HTTP, HTTPS, DNS) ou pour emprunter des chemins de communication indirects via des proxies, augmentant ainsi leur capacité à contourner les restrictions réseau.

Dans notre plateforme, Caldera est hébergée dans un docker sur la machine maître. De plus, le site web d'administration décrit dans la section 3.3.1 contient un wrapper qui permet de déclencher un certain nombre d'attaques en un clic.

Il est important de noter que dans notre architecture, chaque attaque pourrait se dérouler indépendamment des autres. Ainsi, utiliser Caldera comme déclencheur est un choix pratique qui nous permet de simplifier l'exécution de scénario, mais l'utilisation d'un C^2 centralisé n'est pas une condition nécessaire aux cyber-attaques présentées. Des déclencheurs plus discrets et plus stratégique sur les conditions permettant le succès de l'attaque peuvent tout à fait être imaginé, comme par exemple en fonction de l'horloge où d'autres conditions techniques.

4.3 Catégorie 1 - attaque sur l'ECDIS

Dans cette catégorie d'attaques, l'adversaire va prendre le contrôle de l'ECDIS du navire, ce qui lui permet de contrôler les informations techniques présentées à l'équipage.

4.3.1 Introduction initiale

L'ECDIS étant généralement isolé des réseaux externes pour des raisons de sécurité, son vecteur d'attaque principal réside dans les supports amovibles utilisés pour les mises à jour cartographiques. Deux scénarios d'exploitation se distinguent par leur approche d'ingénierie sociale.

Dans le premier cas, l'attaquant procède par hameçonnage ciblé (spear phishing) en envoyant un faux courriel officiel annonçant une mise à jour critique des cartes de navigation. Ce message, soigneusement élaboré pour imiter les communications légitimes des fournisseurs de données nautiques, contient en pièce jointe un fichier malveillant dissimulé dans des données cartographiques apparemment authentiques. Lorsque l'officier chargé de la navigation télécharge ces fichiers et les transfère sur une clé USB pour mise à jour de l'ECDIS, le malware s'installe silencieusement sur le système.

Dans le second scénario, l'attaquant opte pour une approche plus directe en ciblant les événements professionnels du secteur maritime (salons, conventions, formations) où il peut

rencontrer physiquement l'officier responsable des mises à jour. Sous couvert d'un échange professionnel, l'attaquant offre une clé USB de type "rubber ducky" - un dispositif qui se présente comme une clé USB ordinaire mais qui, une fois connectée, émule un clavier et exécute automatiquement des commandes préprogrammées pour déployer le malware sur l'ECDIS.

4.3.2 Prise de contrôle de l'ECDIS

Suite à cela, l'adversaire peut donc exécuter des commandes sur l'ordinateur hébergeant l'ECDIS. Il va notamment aller modifier le fichier de configuration de TimeZero, afin de prendre avantage de l'option définissant le port sur lequel sont reçus les messages NMEA-0183. Il suffit de redémarrer le logiciel pour que la nouvelle configuration soit prise en compte, ce qui est relativement crédible après une mise à jour des cartes.

Le changement dans le fichier de configuration modifie le port d'écoute de TimeZero, qui est au départ le port COM4, pour le remplacer par le port UDP-4000. Grâce à cela, il est possible d'installer un programme malveillant qui récupère les données entrantes sur le port COM4, et écrit les informations de son choix sur le port UDP-4000. Ces informations seront ensuite récupérées par TimeZero, et affichées comme légitimes.

Comme présenté dans la figure 4.1, cette attaque revient à une sorte d'attaque Machine au Milieu (MitM, pour Machine in the Middle en anglais), mais qui se déroulerait sur un hôte plutôt que dans un réseau. Dans cette figure, les traits noirs représentent les données légitimes, et les traits rouges les données contrôlées par l'adversaire. Les carrés collés aux bords de l'image représentent les ports de communication, et ceux au centre représentent les différents logiciels sur la machine.

Suite à cela, le programme a accès à toutes les données sensibles sur la liaison NMEA-0183, supprimant toute confidentialité puisque aucun chiffrement n'est mis en place. La disponibilité est aussi impactée, puisque c'est le maliciel qui choisit s'il transmet ou non les données du port COM4 au port UDP-4000. Autrement dit, c'est le maliciel qui choisit complètement si TimeZero a accès aux données ou non. Enfin, l'intégrité ne peux être garantie, puisque aucun mécanisme de signature n'empêche l'adversaire de modifier les messages qu'il reçoit pour envoyer de fausses informations à TimeZero.

4.3.3 Impacts

À ce point dans l'attaque, les possibilités sont très nombreuses pour l'adversaire. Il contrôle la liaison par laquelle passe les informations suivantes :

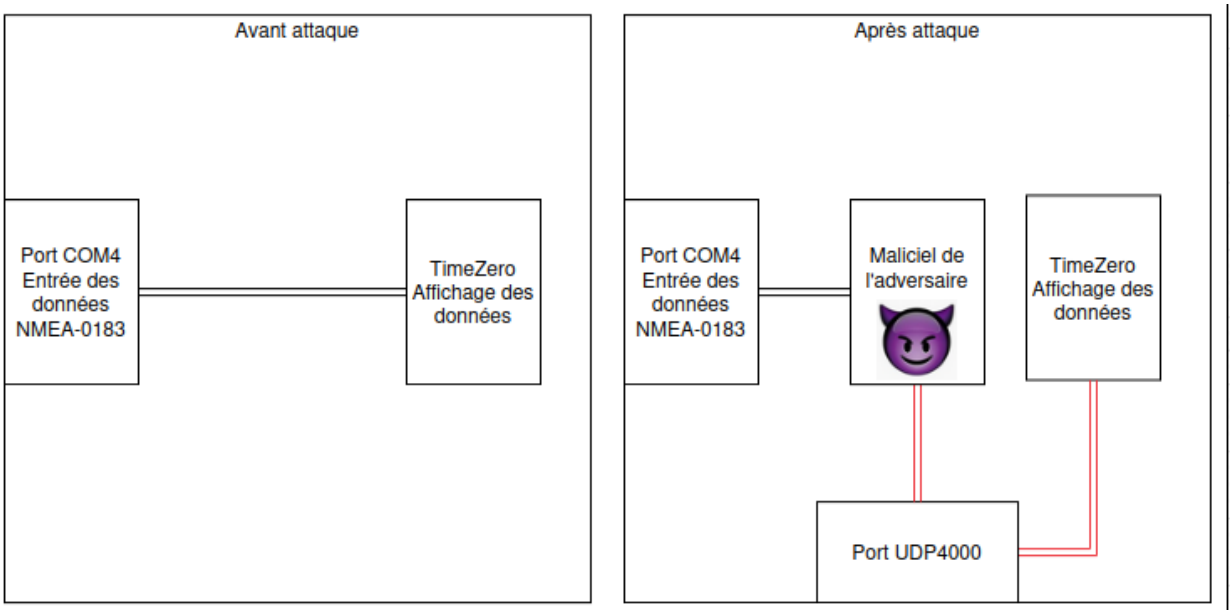

FIGURE 4.1 Comparaison avant/après de la prise de contrôle de l'ECDIS

— L'ensemble des données météorologiques
— La position GPS
— L'ensemble des informations sur d'autres navires arrivant par l'AIS
— Les communications de l'autopilote

Nous avons choisi de mettre en place quatre actions possibles pour l'adversaire, afin de démontrer la diversité des impacts possibles.

Type d'action	Impacts	Complexité technique
Observation silencieuse	Vol d'information et préparation d'attaque complexe	Basse
Déni de service	Stress de l'équipage, petit risque d'accident et révision du navire	Basse
Navire fantôme	Stress de l'équipage, grand risque d'accident et révision du navire	Moyenne
Déviation	Grand risque d'accident ou perte de temps et de ressources	Moyenne

TABLEAU 4.1 Tableau récapitulatif des attaques de l'ECDIS.

Observation silencieuse

La première possibilité pour l'adversaire est de rester discrètement en place, et de laisser tout le trafic entrant par le port COM4 aller vers le port UDP-4000. Dans ce cas, l'attaque est difficilement détectable, et l'adversaire peut collecter beaucoup d'informations.

Ces informations peuvent avoir plusieurs objectifs. Premièrement, si l'adversaire arrive à les exfiltrer, il peut les utiliser dans un contexte d'espionnage, par exemple pour connaître les déplacements exacts du navire.

Ces données peuvent aussi être utilisées directement par le maliciel, qui peut devenir d'autant plus dangereux. En effet, la connaissance du fonctionnement normal du navire peut lui permettre de très bien le simuler lorsqu'il voudra modifier certaines données, et ainsi déjouer certaines méthodes de détections statistiques, par IA, ou même analyses humaines. C'est notamment une des stratégies utilisées par Stuxnet [35], un maliciel extrêmement avancé visant les installations nucléaires iraniennes.

Ce mode de fonctionnement est aussi considéré comme le mode "par défaut", jusqu'à ce que les autres attaques soient déclenchées. Ainsi, les attaques peuvent être déclenchées au moment opportun, plutôt que de se mettre en place dès l'infection, ce qui réduirait grandement l'impact.

Déni de service

Une seconde possibilité est de simplement bloquer totalement la communication, empêchant donc toute communication avec l'ECDIS. Ce dernier deviendrait alors inutilisable, ce qui priverait l'équipage d'un outil très utile.

Si les risques d'accidents liés à cette attaque semblent limités, puisque ces informations sont sans doute accessibles depuis d'autres IHM, elle force le navire à être révisé. De plus, combinée à d'autres attaques qui désactiveraient d'autres éléments du navire, elle pourrait clairement augmenter le risque d'accident, en plus de mettre l'équipage dans un haut niveau de stress.

Navires fantômes

La troisième action prend avantage des flux de données AIS. L'objectif est de faire paniquer l'équipage et de le priver d'un outil extrêmement utile lors de manœuvres techniques dans des lieux ou plusieurs navires se rencontrent, comme un port. En effet, le maliciel se mettra alors à générer de nombreux faux messages AIS, chacun avec une signature différente, afin de remplir l'ECDIS de signaux, comme vu dans la capture d'écran 4.2.

FIGURE 4.2 Présentation de l'attaque qui fait apparaître des navires fantômes

Cette attaque rend l'ECDIS illisible, camouflant totalement les véritables navires qui essayeraient de communiquer. L'équipage se retrouve donc obligé de se baser sur d'autres éléments comme le radar ou leur vue pour détecter les obstacles, rendant toute manœuvre nettement plus difficile et anxiogène. Encore une fois, cette attaque assure que le navire sera envoyé en révision très rapidement.

De plus, il est important de signaler que cette attaque pourrait aussi être utilisée pour générer des navires dont la trajectoire est faite pour croiser la notre, forçant probablement l'équipage à changer sa trajectoire par précaution, surtout en cas de mauvaise visibilité. Enfin, il serait possible de combiner ça avec du déni de service en cachant certains navires à l'ECDIS, obligeant l'équipage à les voir d'une autre façon, alors même que l'attaque n'est pas forcément encore détectée.

Attaque de déviation

La dernière action possible a pour objectif de dévier le navire de sa trajectoire. Supposons que l'objectif soit de le faire dévier vers bâbord. Dans ce cas, le maliciel va légèrement modifier les coordonnées GPS, afin de les placer légèrement a tribord des coordonnées réelles. Le pilote

ou l'autopilote devrait alors tourner légèrement vers bâbord, entraînant la déviation. Il suffit alors au maliciel de continuer à fournir des coordonnées GPS cohérentes avec la trajectoire initiale pendant que le navire s'en éloigne.

Cette attaque présente deux risques principaux, en fonction de l'environnement ou elle est engagée. Si cette attaque est mise en place dans un couloir d'eau relativement étroit comme le Saint Laurent au niveau du port de Montréal, une déviation légère peut rapidement provoquer un accident. Des conditions météorologiques qui réduisent la visibilité peuvent grandement augmenter la probabilité de réussite de l'attaque.

Un deuxième cas d'usage de cette attaque se présente au large. En effet, en l'absence d'autres repaires visuels, l'attaque peut mettre un certain temps à être découverte, provoquant potentiellement de grands détours. Ces derniers peuvent être extrêmement préjudiciables pour les victimes, en provoquant de grands délais sur le planning de navigation et en augmentant la consommation de carburant.

4.4 Catégorie 2 - Attaque sur les communications du système de propulsion

La deuxième catégorie d'attaque vise le système de propulsion de navire. Il s'agit d'une attaque sur le réseau, qui n'implique pas de prendre le contrôle d'un appareil légitime.

4.4.1 Introduction initiale

Pour parvenir à ses fins, l'adversaire doit introduire un appareil sous son contrôle dans le réseau du navire. Nous supposons que cet appareil est installé lors d'une manutention du navire par un ouvrier. Il existe de nombreuses raisons qui pourraient pousser un ouvrier à installer un appareil malveillant au sein de navire, telles qu'une envie de se venger de son employeur pour de mauvaises conditions de travail, ou encore un désaccord idéologique. Il est aussi possible que la personne mettant en place l'appareil ne soit pas à l'origine de l'attaque, mais qu'elle soit simplement payée, manipulée, ou forcée par un adversaire extérieur. Au vu de l'ampleur d'un chantier naval, il n'est pas possible d'avoir un taux de confiance élevé pour chaque acteur de l'écosystème.

Une autre possibilité serait une attaque de la chaîne d'approvisionnement, où un sous-traitant du chantier naval serait le vecteur ou la source de l'attaque. Dans ce cas, l'appareil malveillant serait caché au sein d'une autre pièce légitime, et installé sans que personne sur le chantier naval n'en ait conscience.

Dans le cadre de notre plateforme, cet appareil malveillant est une Raspberry Pi 4.

4.4.2 Prise de contrôle de la communication

Une fois que l'appareil est installé dans le sous réseau cible, il va déclencher une attaque de Machine au Millieu (Machine In The Middle, soit MitM) sur les communications Modbus TCP. L'objectif d'une attaque MitM est de s'insérer dans une communication entre deux parties, et de faire croire à chacune des parties que l'on est l'autre. Ainsi, notre machine se retrouve au milieu de la communication, d'où le nom.

Cette attaque va être exécutée sur le protocole ARP (Address Resolution Protocol, ou protocole de résolution d'adresses). Ce protocole est celui qui rend possible l'utilisation de l'IP. En effet, si à grande et moyenne échelle les paquets sont dirigés grâce aux adresses IP, à petite échelle, soit au niveau d'un sous réseau, ce sont les adresses MAC qui sont utilisées. Le protocole ARP permet de faire la correspondance entre une adresse IP et une adresse MAC, et donc d'acheminer les paquets IP à leur destination.

Cependant, parce qu'il a été pensé très tôt dans l'histoire de l'informatique, et parce qu'il n'est fait pour être utilisé qu'à une petite échelle, ce protocole n'est absolument pas sécurisé et fonctionne uniquement sur la confiance. Voici comment se déroule une communication ARP légitime :

1. La machine 'A', dont l'IP est X.X.X.1 doit communiquer avec l'IP X.X.X.2, mais elle ne sait pas encore à quelle adresse MAC cette IP correspond. Elle envoie donc une requête de type "Je suis X.X.X.1, et je cherche l'adresse correspondant à l'IP X.X.X.2".
2. La machine 'B', dont l'IP est X.X.X.2, répond alors à la machine 'A' en lui indiquant son adresse MAC.
3. La machine 'A' enregistre dans un tableau nommé le cache ARP la correspondance entre l'IP X.X.X.2 et l'adresse MAC de la machine 'B', et peut donc communiquer avec elle maintenant.

Cependant, il existe aussi un autre type de message ARP, les requêtes spontanées. L'objectif est de limiter la quantité de requêtes ARP venant de toutes les machines d'un sous réseau lors de l'introduction d'une nouvelle machine dans ce dernier. À la place, cette machine va spontanément envoyer une requête pour informer les membres du sous réseau de son adresse IP et de son adresse MAC, que chaque appareil va donc pouvoir inscrire dans son cache ARP. C'est de ce type de requête dont l'adversaire va abuser pour empoisonner le cache ARP des appareils victimes.

En effet, en envoyant une requête de ce type à la machine A, et en prétendant donc associer son adresse MAC à l'IP de la machine B, l'adversaire pourra recevoir tous les paquets que A destinerait à B.

Grâce à cette méthode, l'adversaire peut donc se placer entre le PLC et les capteurs/actionneurs avec lesquels il communique. Les communications entre ces derniers sont en Modbus/TCP, une version du protocole Modbus qui fonctionne à travers une connexion IP, donc vulnérable à l'empoisonnement ARP.

4.4.3 Impacts

À ce point dans l'attaque, l'adversaire peut modifier toutes les données envoyées par le PLC aux moteurs, capteurs de température, et capteurs de rotation. Il peut donc modifier la commande de vitesse qui va du PLC vers le moteur, ainsi que les informations de la vitesse réelle des moteurs, et celle de leur état, à travers la température. De nouveau, nous avons choisi de mettre en place quatre types d'actions pour démontrer la diversité des impacts qu'une telle attaque pourrait avoir. Ces actions sont présentées dans le tableau 4.2, et sont détaillées dans les sous-sections suivantes.

Observation silencieuse

Comme présenté dans la partie 4.3.3, l'adversaire peut tout à fait se contenter de retransmettre les communications légitimes sans rien modifier. Cependant, l'attaque par empoisonnement ARP est une attaque bien plus bruyante que celle de la première catégorie, aussi il est relativement peut probable qu'elle puisse rester en place pendant des longues périodes sans que personne ne s'en rende compte. En effet, le passage par un autre appareil induit un léger délai dans les cycles du PLC, et les requêtes spontanées sont sensées être rares, donc une analyse des communications, même sommaire, peut les rendre visibles.

Malgré tout cela, l'attaque est bien possible et a un intérêt pour certains adversaires, comme

Type d'action	Impacts	Complexité technique
Observation silencieuse	Vol d'information et préparation d'attaque complexe	Basse
Déni de service	Stress de l'équipage, petit risque d'accident et révision du navire	Basse
Contrôle de la vitesse	Stress de l'équipage, grand risque d'accident et révision du navire	Haute
Fausse panne	Longue révision du navire, stress de l'équipage	Haute

Tableau 4.2 Tableau récapitulatif des attaques sur le système de propulsion.

dans des scénarios d'espionnage industriel.

Déni de service

Cet impact est encore une fois semblable à celui correspondant dans la première catégorie, qui est présenté en partie 4.3.3. L'adversaire décide dans ce cas de simplement bloquer toute la communication, empêchant le PLC de communiquer avec les capteurs et actionneurs. Cependant, en brisant le cycle de communication sur lequel repose tout l'écosystème OT, cette attaque garanti de déclencher les alertes de sécurité anti-pannes du navire.

Ainsi, cette attaque n'est pas des discrète et promet d'être rapidement détecter par les spécialistes qui l'investigueront, ce qui dévoilera probablement la présence de l'appareil malveillant qui en est la source.

En terme d'impact, cette attaque coupe entièrement la communication entre les capteurs et les actionneurs, empêchant les capteurs de remonter des informations vitale au bon fonctionnement du navire, et bloquant les actionneurs dans leur fonctionnement actuel. Déclenchée au bon moment, cette attaque pourrait avoir des conséquence désastreuse, mais la probabilité de succès reste basse.

Contrôle de vitesse

Le prochain impact est un peu plus avancé puisqu'il implique de modifier certains paquets que le PLC envoie aux actionneurs. Cette attaque implique de savoir quel registre contient la donnée que l'on veut modifier. Cela peut être découvert à l'avance, lors de la reconnaissance, trouvé sur place lors d'une observation silencieuse, ou bien un mélange des deux.

Une fois le registre cible trouvé, il suffit de modifier chaque communication Modbus qui cherche à l'écrire pour y remplacer la valeur cible. Comme présenté dans l'image 4.3, la vitesse cible des moteurs a été fixée à 100 par l'adversaire, alors que le PLC et le serveur SCADA tentent de la fixer à 44.

Fausse panne

Finalement, le dernier impact présenté correspond à la modification de trames qui vont de capteurs vers le PLC. Ici, on souhaite modifier l'une des valeurs pour faire remonter de fausses informations. L'adversaire modifie alors les données transmises par un thermomètre au PLC pour faire croire à une surchauffe. Puisque pour notre preuve de concept on ne modifie que la température, les moteurs restent en réalité bien fonctionnels et suivent toujours

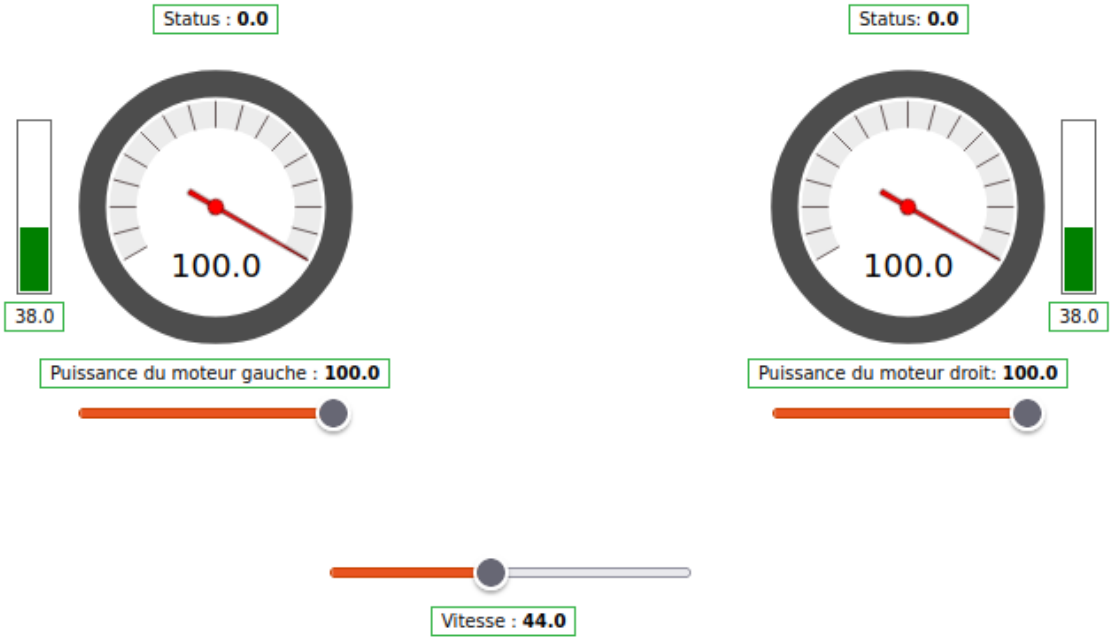

FIGURE 4.3 Présentation de l'attaque de prise de contrôle sur la vitesse

les commandes données, contrairement au cas de la vraie panne où les moteurs cessent de fonctionner, et réduisent leur vitesse presque à zéro.

Évidement, une attaque complexe pourrait combiner cette attaque avec la précédente pour mieux la cacher, et faire croire qu'une panne modifie la vitesse de rotation des moteurs.

4.5 Conclusion

Ces différentes attaques montrent donc la diversité des risques pour un navire. Dans un système aussi interconnecté, perdre le contrôle sur les informations et les commandes qui sont transmises peut entraîner des conséquences humaines, matérielles et financières extrêmement graves, et il est donc important de réfléchir à comment contrer des cyber-attaques. Pour ce faire, la suite de ce mémoire s'attellera à évaluer des outils de détection d'intrusion qui ont fait leurs preuves dans le domaine de l'IT, pour mesurer leurs résultats.

CHAPITRE 5 ÉVALUATION DES MÉTHODES DE DÉTÉCTION D'INTRUSION IT DANS LE CONTEXTE DE LA CYBERSÉCURITÉ MARITIME

Ce chapitre vise à évaluer trois IDS qui sont à l'état de l'art de la cybersécurité dans le domaine de l'IT, et à comprendre ce qu'ils peuvent apporter au domaine maritime, ainsi que leurs limites. Les trois IDS comparés sont Snort, Suricata et Zeek. Chacun est présenté plus en détail au début de la partie le concernant.

5.1 Explication de notre approche

5.1.1 Utilisation d'outil IT

Comme nous avons pu le voir dans l'état de l'art, il existe de la recherche pour développer des outils et des méthodes de détection d'intrusion spécifique au milieu maritime, avec des approches variées. Cependant, bien qu'active, cette recherche est jeune et confrontée à de nombreuses difficultés. Il est donc peu probable que des IDS spécifiques au maritime soit prêt à l'usage à grande échelle avant plusieurs années, voir décennies.

C'est pour cette raison que nous avons souhaité évaluer l'utilité des outils IT dans ce domaine, car ces derniers sont prêts à être déployés et pourraient donc permettre de renforcer la sécurité des navires très rapidement, en l'attente d'une solution spécifique.

Ainsi, notre étude va évaluer les capacités des systèmes de détections d'intrusion IT, afin de vérifier si ces derniers peuvent remplir ce rôle avec de bonnes capacités de détections tout en proposant un déploiement rapide.

5.1.2 Détéction réseau

Notre plateforme utilise de l'émulation afin de reproduire au mieux les communications existantes sur un navire, en exploitant des protocoles industriels et spécifiques au maritime comme NMEA-0183 et Modbus. Cependant, les appareils OT eux-mêmes sont représentés par des Raspberry Pi, qui ont une architecture et un fonctionnement extrêmement différents. Ainsi, nous avons décidé de nous focaliser uniquement sur la détection réseau, car c'est ce pour quoi la plateforme a été faite.

De plus, une détection réseau est bien plus simple à mettre en place, car elle nécessite simplement une sonde sur chaque sous-réseau, là où une détection sur les hôtes nécessite de

déployer des logiciels sur chaque appareil que l'on souhaite protéger. La très grande diversité et spécificité de l'architecture, des systèmes d'exploitations, et des appareils d'OT rendent cette tâche particulièrement ardue, et font donc de la détection au niveau réseau le meilleur choix.

5.1.3 Analyse qualitative

Un des enjeux de la recherche en cybersécurité du maritime est l'absence de datasets d'attaques, ce qui force notre étude à se focaliser sur une analyse qualitative des résultats. Nous avons vérifié au maximum la véracité de toutes nos affirmations sur notre plateforme, mais il n'est pas possible par exemple d'établir des taux de détection représentatifs.

Notre analyse va donc se focaliser sur les types d'attaques que nous avons proposés dans la partie 4.4, ainsi que les méthodes avec lesquelles on peut essayer de les détecter.

5.1.4 Reproductibilité entre nos expériences

Pour assurer que nos expériences se déroulent toutes dans les mêmes conditions, nous avons enregistré un PCAP de toutes les communications sur la plateforme pendant un fonctionnement normal, puis avec chacune des attaques de la seconde catégorie. Tous les IDS testés sont donc comparés sur leur capacité à détecter les attaques contenues dans ce PCAP.

Ce chapitre est découpé en deux grandes parties, la première se focalisant sur l'utilisation des règles par défaut des trois outils évalués, et la seconde essayant de pousser la détection plus loin en proposant des règles écrites spécifiquement pour notre usage.

5.2 Évaluation des outils sans paramétrage spécifique

5.2.1 Snort

Présentation

Snort est un IDS open source développé initialement par Martin Roesch en 1998, puis maintenu par Cisco Systems depuis son acquisition de Sourcefire en 2013. En 2009, Snort a été reconnu par le magazine InfoWorld comme l'un des "meilleurs logiciels libres de tous les temps", dans son "Hall of Fame des logiciels libres".

Snort fonctionne avec plusieurs modules qui ont chacun un rôle dédié :

— Module de capture de paquets : utilise la bibliothèque libpcap pour intercepter les paquets circulant sur le réseau. Dans notre cas, cette étape est déjà effectuée puisqu'on

lui fournit un fichier PCAP.
- Préprocesseurs : préparent les paquets à l'analyse en effectuant diverses opérations comme la défragmentation IP, la reconstitution des sessions TCP, la normalisation HTTP, ou la détection d'anomalies protocolaires.
- Moteur de détection : cœur du système, il applique les règles de détection aux paquets prétraités pour identifier les signatures d'attaques connues ou les comportements anormaux.
- Plugins : ils permettent de contrôler le format des alertes, et de les écrire dans différents formats/localisations, pour de la journalisation système, des bases de données, ou bien des fichiers texte personnalisés.

Snort fonctionne donc en fonction d'un ensemble de règles. Il existe un ensemble de règles officielles, développées et maintenues par l'équipe Talos de Cisco Systems. Ces règles sont accessibles immédiatement via une version commerciale qui est disponible avec un abonnement payant, mais rejoignent la version communautaire gratuite 30 jours après leur sortie. Nous avons donc utilisé la version communautaire de ces règles pour évaluer Snort. De plus, par défaut le module Modbus n'est pas activé, il a donc fallu évidemment l'activer dans le fichier de configuration.

Résultats

Premièrement, Snort inclut un préprocesseur spécifiquement dédié à la détection d'attaques d'empoisonnement ARP, qui est donc le moyen par lequel toutes les attaques sont rendues possibles. C'est relativement peu usuel, car les IDS se concentrent généralement sur les couches OSI trois et au dessus, alors que l'attaque d'empoisonnement ARP se déroule sur la couche deux. Techniquement, Snort détecte donc toutes les attaques par ce moyen.

Cependant, cette méthode de détection se limite à une seule méthode d'attaque, et il est techniquement possible d'implémenter d'autres attaques sans passer par cette méthode. En effet, prenons un appareil malveillant qui n'est pas installé sur un réseau, mais plutôt sur le support de la communication ; par exemple en étant directement branché sur le câble Ethernet avec deux interfaces. Il obtient alors les mêmes capacités d'attaque Machine au Milieu, mais sans pour autant avoir besoin d'une attaque d'empoisonnement ARP. Ce genre de méthode est parfois appelé en anglais "bump-in-the-wire", qui pourrait se traduire par bosse sur le câble. C'est pour ces raisons que nous allons aussi nous intéresser à la capacité de Snort à détecter les attaques en fonction de leurs impacts et de ce qui est visible au-dessus de la couche 2. Cela permet ainsi d'assurer une meilleure défense en profondeur, plutôt que de se reposer sur la détéction de l'empoisonnement ARP.

Et à ce niveau-là, Snort obtient des résultats relativement moyens. Il est capable de détécter l'attaque de déni de service, mais ni l'attaque sur la vitesse ni l'attaque sur la température ne sont détectées.

5.2.2 Suricata

Présentation

Suricata est un moteur de détection d'intrusion, de prévention d'intrusion et de surveillance de sécurité réseau développé par l'Open Information Security Foundation (OISF) depuis 2009. Ce projet open source a été créé comme une alternative moderne à Snort, avec une architecture conçue dès le départ pour tirer parti des environnements multi-cœurs et offrir des performances améliorées.

Suricata fonctionne également avec une architecture modulaire très proche de celle de Snort :

- Module de capture de paquets : similaire à celui de Snort, et de même nous lui donnons directement un fichier PCAP à évaluer.
- Décodeurs de protocoles : remplissent un rôle proche des préprocesseurs, en découpant les différents protocoles en fonction des couches OSI et rendant tout cela disponible pour le moteur de détection.
- Moteur de détection : comme Snort, le cœur du système de détection de Suricata utilise des règles de détection pour identifier les attaques.
- Module d'inspection de fichiers : permet d'extraire et d'analyser les fichiers transmis sur le réseau, avec notamment la capacité de calculer des hachages pour reconnaître des éléments malveillants par signature.
- Système de sortie : gère les alertes et peut les exporter dans différents formats dont JSON, syslog, et des bases de données, et les écrire dans différentes localisations.

Lors de l'installation, Suricata vient directement avec l'ensemble de règles par défaut, appelées les règles de la communauté. C'est donc ces dernières que nous avons choisies pour évaluer le logiciel, et tout comme précédemment nous avons dû activer manuellement la détection pour Modbus.

Résultats

Contrairement a Snort, Suricata n'est pas capable de détecter l'attaque par empoisonnement ARP.

Néanmoins, il lève des alertes dans le cas de Déni de Service, lorsque les capteurs et actionneurs ne reçoivent pas et donc ne répondent pas aux requêtes du PLC. Suricata détecte

donc que la communication ne se déroule pas normalement et qu'un trop grand nombre de requêtes est ignoré.

D'un autre côté, il ne détecte pas que les données sont modifiées et donc ne lève pas d'alerte particulière pour les attaques sur la vitesse et la température.

5.2.3 Zeek

Présentation

Zeek, anciennement connu sous le nom de Bro jusqu'en 2018, est un système de détection d'intrusion réseau open source développé initialement par Vern Paxson à l'Université de Californie à Berkeley en 1994. Contrairement à Snort et Suricata qui sont principalement des IDS basés sur des règles, Zeek adopte une approche orientée analyse comportementale et se présente comme un framework d'analyse de trafic réseau.

Comparé à Snort et Suricata, Zeek s'articule autour d'une architecture différente :

- Module de capture de paquets : utilise également libpcap pour capturer le trafic réseau ou lire des fichiers PCAP existants comme dans notre cas d'étude.
- Moteur d'analyse de protocoles : au lieu de simplement décoder les protocoles, Zeek les analyse en profondeur et retient l'état des connexions, permettant ainsi une compréhension contextuelle du trafic.
- Interpréteur de scripts : cœur de la flexibilité de Zeek, il exécute des scripts écrits dans le langage spécifique de Zeek (Zeek scripting language) qui définissent les comportements à surveiller plutôt que de simples signatures.
- Gestionnaire d'événements : génère des événements de haut niveau basés sur l'activité réseau qui peuvent être traités par les scripts.
- Système de journalisation : plutôt que de se concentrer uniquement sur les alertes, Zeek produit des logs structurés détaillés pour différents protocoles et activités réseau.

Ainsi, Zeek s'appuie sur des scripts d'analyse du réseau plutôt que de simple règles à comparer au paquet. Comme Suricata, des scripts génériques sont fournis avec l'installation, c'est donc ceux-ci que nous avons évalués.

Résultats

Zeek aussi ignore les éléments avant la couche 3 et donc ne détecte pas l'attaque par empoisonnement ARP. Plus encore, les scripts par défaut de Zeek ne permettent pas non plus de détecter les autres attaques, que ce soit le déni de service, l'attaque sur la vitesse, ou l'attaque sur la température.

5.2.4 Analyse globale des résultats

Le tableau 5.1 présente un résumé des différents résultats obtenus, chaque coche représentant un élément qui a été détecté par l'IDS correspondant à la ligne, et chaque croix un élément que l'IDS ne relève pas.

Globalement, comme on pourrait s'y attendre avec des règles plutôt pensées pour des technologies d'information, la majorité des attaques ne sont pas détectées. En effet, seuls des éléments particulièrement bruyants (comme l'empoisonnement ARP, qui implique d'envoyer une certaine quantité de messages ARP non prévus, ou le déni de service, qui implique de briser le cycle normal des échanges) ont pu être détectés.

L'ensemble des détections qui fonctionnent sont techniquement des détections d'anomalies du déroulé normal des échanges, tels que décrits par les protocoles. C'est une bonne manière de détecter des attaques en détectant leurs méthodes, mais malheureusement, de nouvelles attaques sont inventées régulièrement, et certaines méthodes ne peuvent tout simplement pas facilement être détectées ainsi (par exemple, la "bosse dans le câble" présentée dans la section 5.2.1, sur les résultats de Snort).

Ainsi, il nous semble pertinent d'essayer de mettre en place des méthodes de détection qui visent directement les impacts de l'attaque, c'est à dire qui visent à vérifier l'intégrité des données transportées sur le réseau. C'est donc le sujet de la prochaine partie.

5.3 Mise en place d'une détection d'impact

5.3.1 Définition des objectifs

Comme précisé dans la partie précédente, l'objectif principal de notre méthode de détection d'intrusion sera de détecter les impacts des attaques, c'est-à-dire en vérifiant l'intégrité des données. L'objectif est donc de trouver des données qui doivent correspondre, puis de vérifier à tout moment que ces données correspondent. Par exemple, lorsque le PLC envoie une

	Empoisonnement ARP	Déni de Service	Attaque sur la vitesse	Attaque sur la température
Snort	✓	✓	X	X
Suricata	X	✓	X	X
Zeek	X	X	X	X

Tableau 5.1 Tableau récapitulatif des capacités de détections des trois IDS évalués, dans leur configuration par défaut.

commande au moteur pour qu'il tourne à 20 RPM, le capteur de vitesse de rotation doit détecter une vitesse de 20 RPM.

L'objectif est simplement de présenter une preuve de concept de ce genre de méthode de détection, afin de voir comment elle pourrait être implémentée, et de pouvoir évaluer ses avantages et ses intérêts. Pour ces raisons, nous nous sommes focalisés sur une seule des attaques qui modifient des données, l'attaque sur la vitesse. Ainsi, dans la suite de cette partie, nous prendrons cette attaque comme exemple, bien que la méthode puisse être appliquée à n'importe quel type de donnée.

Évidemment, l'objectif est de rester le plus générique possible, et d'essayer de ne pas faire un système de détection trop spécifique à cette attaque, mais bien d'assurer une certaine cohérence entre les données.

Enfin, il fallait choisir l'IDS dans lequel implémenter cette méthode de détection : Zeek, Suricata, et Snort. Nous avons choisi de partir de celui qui avait les meilleurs résultats, à savoir Snort.

5.3.2 Design de règles

Avant d'écrire des règles qui puissent remplir notre objectif, nous devons comprendre comment apparaissent les communications modifiées par l'attaque pour un IDS.

Analyse des communications

Dans notre réseau, après que l'empoisonnement ARP ait été effectué, tous les messages passent par l'appareil malveillant, et certains sont modifiés. Ainsi, si on ignore la couche deux du modèle OSI (comme le font la majorité des IDS pour leurs analyses), ces messages traversent le réseau deux fois, mais sont identiques : même IP source et destination, même ports source et destination, même numéro dans la séquence TCP, même identifiant de transaction dans l'entête Modbus-TCP... Les seuls changements, quant il y en a, sont uniquement dans le champ de données. Ainsi, ces deux messages (entre le PLC et l'appareil adverse et entre l'appareil adverse et le capteur, ou actionneur), sont donc considérés comme les mêmes. Par conséquent, le second ne va pas être capturé, et ne va pas être analysé par l'IDS, qui ne verra que le premier (donc avec la commande "originale" de la vitesse, et non pas celle modifiée par l'appareil malveillant).

Ce comportement est finalement tout à fait logique, en effet dans le cas où plusieurs sous-réseaux sont monitorés par le même IDS, des messages qui traversent plusieurs de ces sous-réseaux ne doivent être analysés qu'une seule fois.

En conséquence, cela signifie que lors de notre analyse, nous aurons toujours accès aux messages avant qu'ils ne soient modifiés par l'adversaire sur le réseau. Avec cela en tête, nous pouvons maintenant nous pencher sur la structure d'une règle Snort.

La figure 5.1 présente un exemple de règle. Cette dernière peut être découpée en deux parties principales : l'en-tête, sur la première ligne, et le corps de règle, entre parenthèses.

Structure de l'en-tête

L'en-tête contient plusieurs mots-clés. Le premier, `alert`, permet de dire à Snort que faire s'il détecte un paquet qui correspond à la règle. L'action `alert` est la plus fréquente, mais il existe des mots-clés comme `block` ou `reject`, pour bloquer la communication lorsque Snort est utilisé comme IPS. Pour notre utilisation, nous avons le mot-clé `alert`, puisque nous voulons simplement détecter les attaques.

Le second mot-clé, `tcp` dans l'exemple, est le premier filtre qui va permettre de matcher les paquets. Il s'agit du protocole. Il est possible d'y mettre un protocole de couche 3, comme ICMP ou IP, un protocole de couche quatre comme TCP ou UDP, ou enfin un service applicatif de couche 7, comme par exemple Modbus ou HTTP. Ce mot-clé définit le préprocesseur qui sera appelé pour s'occuper de ces paquets, et donc donne accès à certains mots-clés spécifiques dans le corps de la règle. Évidemment, nos règles utilisent le mot-clé `modbus`.

Le reste de l'en-tête permet de filtrer les paquets en fonction de leur destination et de leur source. De chaque côté de la flèche se trouve une adresse IP et un port. Dans l'exemple, les IP sont remplacées par des variables, mais elles peuvent aussi être une adresse précise, un sous-réseau caractérisé par son masque, ou même une liste d'IP discontinue. Enfin, il est possible d'utiliser le mot-clé `any`, pour permettre à la règle de matcher n'importe quelle IP. Dans notre cas, la source est le PLC, et son IP `10.123.0.3`, et la destination est les deux

```
alert tcp $EXTERNAL_NET 80 -> $HOME_NET any
(
msg:"Attack attempt!";
flow:to_client,established;
file_data;
content:"1337 hackz 1337",fast_pattern,nocase;
service:http;
sid:1;
)
```

FIGURE 5.1 Exemple d'une règle Snort, issu de la documentation officielle [1]

capteurs/actionneurs, donc `10.123.0.4, 10.123.0.6`.

De même, les ports peuvent être un seul port, une liste, ou `any`. Nous laisserons le mot-clé `any`, car le port sur lequel les capteurs et actionneurs communiquent est aléatoire, et que le port 502 de la source est déjà vérifié par le mot-clé `modbus`.

Enfin, la flèche au milieu peut être unidirectionnelle (`->`), pour s'assurer que la source et la destination soient précisément identifiées, ou bidirectionnelle (`<>`), pour que la source et la destination soient interchangeables. Dans notre cas, nous utiliserons la version bidirectionnelle.

Après l'en-tête, vient le corps de la règle. Nous ne détaillerons pas tous les mots-clés qui peuvent être utilisés ici, et nous contenterons de présenter ceux que nous utiliserons.

Mots clef du corps de règle

Il existe deux mots-clés présents dans quasiment toutes les règles. Le premier est `msg`, et permet d'assigner un message, qui sera celui inscrit dans les journaux lorsque la règle sera appliquée. Il contient donc en argument une chaîne de caractères qui décrit la raison d'être de la règle. Ce message devrait rester relativement générique et ne peut donc pas contenir de variables.

Le second est `sid`, qui correspond simplement à un identifiant pour la règle. Ce dernier doit être unique pour chaque règle chargée dans Snort à chaque utilisation, aussi des plages sont réservées pour l'usage personnel, et ce sont ces dernières que nous utiliserons.

Un autre mot-clé que nous allons beaucoup utiliser est `flowbits`. Ce dernier permet d'enregistrer des bits de donnée dans un contexte global, afin de pouvoir utiliser cette information dans une autre règle. Cette fonctionnalité nous sera indispensable, car les deux données que l'on souhaite corréler se trouvent dans deux communications Modbus différentes : la première, la commande de vitesse de rotation que le PLC envoie au moteur, se trouve dans un message Modbus de type 16, qui permet d'écrire des données dans la mémoire lecture/écriture de l'actionneur. La seconde, la donnée de vitesse de rotation du moteur qui vient du capteur et est remontée vers le PLC, se trouve dans la réponse à un message de type 04, qui permet de lire des données dans la mémoire de lecture seule du capteur. Ainsi, sans flowbits, nous ne pourrions pas connecter les informations contenues dans ces deux communications distinctes.

Dans notre système de détection, un premier groupe de règles enregistre la commande du PLC dans des flowbits, et un deuxième groupe de règles compare ces flowbits à la vitesse remontée par le capteur. Puisqu'il s'agit de bits, chacun correspondra à une certaine plage de valeur. Enfin, pour éviter un trop grand nombre de faux positifs, puisque la valeur du

capteur oscillera probablement autour de la valeur de commande, nous accepterons que les deux vitesses se trouvent dans des plages adjacentes.

Enfin, il existe un flowbit spécial, défini par Snort, qui a un rôle particulier. Il s'agit du flowbit `noalert`. Ce dernier permet de demander à Snort de ne pas déclencher d'alerte, même si le mot-clé `alert` est utilisé en action. Nous l'utilisons donc pour le premier groupe de règles, chargé d'enregistrer la vitesse dans les flowbits, car ces règles s'activeront pour chaque communication afin de garder la valeur enregistrée valide, mais cela ne signifie pas qu'il y a une attaque.

Il existe par ailleurs un mot-clé spécifique à Modbus qui permet de vérifier la fonction Modbus utilisée, il s'agit de `modbus_func`. Un comportement intéressant que nous avons remarqué, est que le PLC ne se contente jamais de lire uniquement certaines données, mais que lorsqu'il envoie une requête de lecture, il va lire toutes les données présentes dans la cible de la requête. Ce comportement est la conséquence directe du fait que le PLC essaie de mettre le plus régulièrement possible à jour toutes les données dont il pourrait avoir besoin dans un cycle de calcul. Ainsi, il suffit de capturer la bonne requête, et l'information que l'on y cherche sera toujours là, et toujours au même offset.

Pour lire ces données, on utilise avant tout le mot-clé `modbus_data`, qui permet de placer le curseur de lecture au début de la zone de données Modbus. Ensuite, on utilise un dernier mot-clé, `byte_test`. Ce dernier permet simplement de comparer un certain nombre d'octets dans le paquet à une valeur arbitraire. Il a cependant l'effet de déplacer le curseur d'autant d'octets, il faut donc le replacer après.

Règles proposées

En assemblant tout ce qui a été proposé jusqu'à présent, voici à quoi ressemblent les premières règles, qui sont chargées d'enregistrer la commande de vitesse :

```
alert modbus  10.123.0.3 any <> [10.123.0.4, 10.123.0.6] any
(
msg: "Logging speed_target - 5";
modbus_func:16;
modbus_data;
byte_test: 2, <,10, 5,relative;
modbus_data;
byte_test: 2, >=, 0, 5, relative;
flowbits:set, speed_5;
```

```
flowbits:set, speed_15;
flowbits:unset, speed_25;
flowbits:unset, speed_35;
flowbits:unset, speed_45;
flowbits:unset, speed_55;
flowbits:unset, speed_65;
flowbits:unset, speed_75;
flowbits:unset, speed_85;
flowbits:unset, speed_95;
flowbits:noalert;
sid:1000001;
)
```

Comme écrit dans le message, cette règle sert à enregistrer la vitesse cible que le PLC demande aux moteurs. Cette règle, spécifiquement, vérifie si cette vitesse est dans l'intervalle le plus bas, c'est-à-dire entre 0 et 10. Elle utilise la fonction `byte_test` à deux reprises, pour vérifier que la valeur est dans l'intervalle de 10 % correspondant. Si les deux tests sont passés, alors la règle choisit l'état des 10 flowbits, autorisant donc la vitesse remontée par le capteur à être dans certains des intervalles. Les intervalles autorisés sont ceux avec le mot-clé `set`, soit celui correspondant exactement à la vitesse (`speed_5`), et ceux adjacents (ici seulement celui au-dessus, `speed_15`). Tous les autres ont donc le mot-clé `unset`. Enfin, la règle utilise le flowbit `noalert`, pour ne pas lever une alerte à chaque fois que cette valeur est enregistrée. Ces règles sont donc au nombre de 10, pour chacun des intervalles, et sont associées aux ID 1000001 à 1000010.

Voici un exemple du second type de règle, qui va se charger de lever une alerte si la vitesse remontée par le capteur n'est pas dans un des intervalles autorisés :

```
alert modbus  10.123.0.3 any <> [10.123.0.4, 10.123.0.6] any
(
msg: "MODBUS CUSTOM, Speed from motor does not correspond
to target speed set by PLC. ";
modbus_func:4;
modbus_data;
byte_test: 2,<, 10, 3, relative;
modbus_data;
byte_test: 2,>=, 0, 3, relative;
```

```
flowbits:isnotset, speed_5;
sid:10000011;
)
```

Tout le début de cette règle est semblable à la règle précédente, au détail près de la fonction Modbus cherchée, et de l'adresse à laquelle on cherche la donnée. Le message associé est aussi plus formel, puisque celui-ci va vraiment apparaître dans les alertes, en l'absence du flowbit `noalert`. Cependant, au lieu d'assigner une valeur aux 10 flowbits, ce type de règle va simplement vérifier si le flowbit correspondant est `set` ou non. Dans le deuxième cas, cela signifie que la vitesse lue dans le capteur n'est pas proche de la vitesse cible, et une alerte est générée. Il existe 10 de ces règles, une par intervalle, et elles portent les IDs 10000011 à 10000020.

5.3.3 Résultats

Nous avons donc utilisé ces 20 règles pour détecter une attaque sur la vitesse. Pour nos expérimentations, nous avons utilisé un PCAP qui a enregistré le réseau pendant 7 minutes, dont 4 minutes pendant lesquelles l'attaque était en cours. Au total, 31 357 paquets ont été capturés, dont 11 236 sont des paquets Modbus, soit 35,8 %. L'attaque en tant que telle est bien détectée, et en tout le système a généré 681 alertes, dont 675 d'entre elles sont de vrais positifs.

Les six faux positifs correspondent à des moments où le navire change de vitesse, et donc où les moteurs mettent un peu de temps à atteindre la vitesse cible. Cette inertie entraîne en moyenne deux alertes comme symptôme d'une attaque lors de chaque changement de vitesse. Il est important de noter ici que l'inertie présente sur la plateforme a été manuellement codée pour plus de fidélité, et qu'en réalité les phases d'accélérations et de décélérations d'un navire sont bien plus longues. Pour limiter le nombre de faux positifs, il serait envisageable d'utiliser un système de seuil, où il faut qu'une règle soit activée plusieurs fois dans un intervalle de temps donné pour qu'une alerte soit générée. Cependant, l'inclusion d'un seuil donne aussi de nouvelles possibilités à l'adversaire pour éviter la détection, par exemple en changeant régulièrement la cible de vitesse pour que la vitesse réelle oscille dans plusieurs intervalles et donc ne dépasse jamais le seuil attaché à une règle. Aussi, il nous semble que le plus stratégique est de garder cette règle sans seuil, afin d'assurer un maximum de détection, mais de la rendre informative, par exemple en la rendant visible depuis le pont pour que l'équipage puisse s'en servir et donc ne s'en alarmer que lorsqu'il n'y a pas de changement de vitesse prévu.

Aucun faux négatif n'a existé dans notre configuration, mais ils pourraient arriver si la vitesse cible n'est que marginalement modifiée (c'est-à-dire que la vitesse fixée par l'adversaire est dans l'un des intervalles marqués comme valide par la vitesse fixée par le PLC). Au maximum, dans notre configuration actuelle, cette attaque pourrait impacter la vitesse de 20 %, dans une configuration où la commande réelle serait à l'une des bornes d'un intervalle, et où l'adversaire la modifie vers la borne opposée de l'ensemble des trois intervalles autorisés. De manière plus réaliste, en considérant les oscillations du capteur, il est probablement impossible de modifier la vitesse de 10 ou 15 %, mais cela peut tout de même avoir des impacts non négligeables sur des délais de livraison par exemple. Une façon de réduire encore cet impact potentiel est d'augmenter le nombre d'intervalles, réduisant encore la fenêtre de l'adversaire. Le seul désavantage de cette solution est l'augmentation de la puissance de calcul nécessaire pour faire fonctionner le système en temps réel.

5.3.4 Difficultées d'adoption

Par rapport à notre objectif initial d'utiliser les techniques IT pour obtenir un système de détection d'intrusion facilement déployable sur des navires à court terme, cette méthode apporte certaines complications. En effet, elle est extrêmement dépendante de l'infrastructure du navire pour générer ces règles. Il faut savoir exactement les adresses IP du PLC et des appareils auxquels il parle, les valeurs minimales et maximales des données que l'on veut lier, ainsi que leur adresse dans les paquets. Tout cela fait que les règles doivent être écrites individuellement pour chaque navire.

Ce n'est pas forcément un expert en cybersécurité qui doit intervenir, car un script peut tout à fait générer les règles en fonction des données d'entrée, donc ce travail peut être fait par un ingénieur de bord. Néanmoins, cette étape nécessaire empêche un déploiement aussi rapide qu'espéré.

Cependant, étant donné qu'il est, dans tous les cas, nécessaire de déployer ces IDS pour sécuriser les parties IT de l'infrastructure des navires, notre ajouté les règles proposées permet tout de même d'ajouter une couche de sécurité supplémentaire sans investissement supplémentaire. Ainsi, même si la solution proposée n'est pas aussi facile et rapide à mettre en place que nous l'avions supposé, elle reste intéressante à implémenter au vu de la sécurité ajoutée et de l'investissement que cela représente.

5.3.5 Puissance de calcul

Enfin, nous nous sommes intéressés à la vitesse de traitement de Snort sur ces règles. En effet, la quantité de règles et de flowbits pour suivre une seule donnée nous semble relativement importante et nous avions donc peur qu'un IDS soit rapidement dépassé. Nous avons donc écrit un deuxième groupe de règles, avec cette fois-ci seulement deux paliers (supérieur à 50, et inférieur à 50), et nous avons donc successivement demandé à Snort d'analyser le même PCAP avec ces deux groupes de règles, sur le même système et sans programmes parasites allumés, pour avoir des capacités de calcul les plus proches possibles dans les deux cas. Dans le cas avec 10 paliers, Snort a été capable d'analyser 405 292 paquets par seconde, et dans le cas avec deux paliers, Snort a été capable d'analyser 472 728 paquets par seconde. Il y a donc un écart de 17%, qui, sans être négligeable, est acceptable.

De plus, même si la charge devient trop importante pour un IDS centralisé, il est tout à fait envisageable d'utiliser un système réparti pour surveiller les différents sous-réseaux.Il pourrait aussi être intéressant d'évaluer la capacité d'agents autonomes comme des LLM à trouver les données nécessaires à la rédaction des règles dans de la documentation. Cette solution pourrait rendre bien plus facile à mettre en place notre méthode de détection.

CHAPITRE 6 CONCLUSION

Cette conclusion présente une synthèse des travaux réalisés sur nos différents angles de recherche, puis les limitations que notre travail peut avoir, avant de proposer une réflexion sur des améliorations ou des développements du projet.

6.1 Synthèse des travaux

Cette section revient sur les différents travaux engagés durant cette maîtrise.

6.1.1 Plateforme de tests

La question de recherche qui encadre le développement de la plateforme de test est la suivante :

Comment concevoir une infrastructure de simulation, de test, et d'entraînement sur la cybersécurité des technologies opérationnelles maritimes ?

Pour répondre à cette question, nous avons identifié différentes caractéristiques qui font une bonne infrastructure de test en cybersécurité, et nous avons choisi d'axer nos efforts sur la fidélité, l'évolutivité, et le coût-efficacité. Notre plateforme repose donc sur l'émulation pour s'assurer que le réseau est le plus fidèle possible à la réalité. Dans la même idée, nous avons fait le choix d'un logiciel professionnel pour notre ECDIS, TimeZero.

Nous avons donc mis en place notre plateforme pour inclure des éléments d'interfaces avec l'équipage, tels que l'ECDIS, les commandes de contrôle du navire, et le SCADA, et un ensemble d'appareils de technologies opérationnelles pour représenter la chaîne de contrôle de la propulsion. Enfin, pour certains éléments que nous ne pouvions pas générer, nous avons choisi la simulation, comme pour la position GPS ou les données météorologiques.

Pour interconnecter ces éléments, nous avons utilisé des protocoles adaptés au domaine, à savoir NMEA-0183, un protocole développé spécifiquement pour les systèmes maritimes, et modbus-TCP, un protocole générique au milieu industriel, mais qui est utilisé au sein des navires.

6.1.2 Déploiement d'attaques

Notre seconde question de recherche est la suivante :

Quels types de cyberattaques implémenter pour démontrer les impacts potentiels sur le système naval et son environnement opérationnel ?

Nous avons choisi de placer des attaques contre deux éléments, afin de pouvoir obtenir des résultats complémentaires :

Dans un premier temps, nous avons attaqué l'interface par laquelle l'équipage obtient la majorité de ses informations, l'ECDIS. Par cette catégorie d'attaques, nous avons démontré plusieurs façons d'interférer avec ces informations, et d'augmenter les risques d'accidents, ou de forcer le navire à être arrêté pour être réparé. Cette attaque a lieu directement sur la machine de l'ECDIS, et affecte la communication NMEA-0183.

La seconde catégorie d'attaque se focalise sur le réseau des appareils opérationnels. Elle vise à démontrer la vaste possibilité des impacts qu'une attaque sur ce réseau pourrait entraîner, notamment vu le manque de sécurité du protocole modbus. En montrant des attaques qui modifient les commandes que reçoivent les actionneurs et les données que remontent les capteurs, nous démontrons l'amplitude des possibilités qu'un adversaire aurait dans ce genre de conditions.

Nous avons ainsi démontré les risques multiples qu'une cyberattaque sur les systèmes opérationnels pouvait faire courir à un navire.

6.1.3 Détection d'intrusion

La troisième question de recherche était formulée comme suit :

Quelles sont les limites et intérêts des solutions de détection d'intrusion prévues pour les technologies informationnelles dans le contexte du domaine maritime ?

Pour pouvoir répondre à cette question, nous avons d'abord déployé les IDS reconnus dans le domaine de la sécurité TI que sont Suricata, Snort, et Zeek, afin d'évaluer leurs capacités à détecter les intrusions nativement. Nous nous sommes ici concentrés sur l'aspect réseau, donc la seconde catégorie des attaques présentées précédemment.

En utilisant donc les règles initiales de ces outils, nous avons pu voir leur efficacité limitée dans le contexte des systèmes opérationnels navals, en ne détectant que les attaques les plus bruyantes.

Nous avons donc décidé d'utiliser Snort pour implémenter un nouveau type de règles qui permet de corréler l'évolution de deux valeurs, une commande vers un actionneur et la même donnée physique mais capturée par un capteur. Pour faire une preuve de concept, nous avons utilisé la vitesse de rotation des moteurs comme valeur à suivre. Cette méthode nous a permis

de détecter l'attaque de prise de contrôle sur la vitesse.

6.2 Limitations

L'ensemble de notre travail explore des éléments de la cybersécurité du maritime qui n'ont pas encore été beaucoup étudiés, mais il possède certaines limites que nous détaillons ici.

Notre plateforme fait partie des plus complètes et complexes existantes. Cependant, une limite inhérente à ce niveau de complexité et au choix de l'émulation plutôt que de la simulation est le manque de reproductibilité par d'autres laboratoires.

De plus, l'émulation permet de simuler au mieux le réseau, mais l'utilisation de logiciels libres plutôt que de véritables appareils de technologies opérationnelles (en dehors de l'ECDIS) nous empêche de faire des tests d'intrusion et de détection directement sur les hôtes.

Du côté de la détection d'intrusion, les limites de notre solution sont principalement liées aux limites du système par règles de Snort. En effet, le fait que nous devions utiliser des flowbits pour comparer deux valeurs complexifie grandement les règles que nous proposons. De plus, l'aspect sans mémoire de ce système empêche d'analyser l'évolution dans le temps des données, et entraîne donc des difficultés supplémentaires, telles que les faux positifs liés a l'inertie.

Il y a aussi le problème des faux négatifs, mais l'idée même de suivre des données physiques implique des variations dues aux capteurs, aux actionneurs, et à leur environnement physique, ce qui oblige à avoir un intervalle acceptable. Pour passer dans cet intervalle, l'adversaire doit garder ces valeurs relativement proches des valeurs réelles, donc la réduction de ses possibilités d'attaque est déjà un point positif. Néanmoins, l'utilisation d'un modèle par règle implique aussi que cet intervalle a un impact sur la vitesse de calcul de Snort, et cette contrainte est probablement plus importante que celle liée aux variations statistiques.

Les faux positifs générés à chaque changement brusque de la donnée observée sont aussi une limite de nos travaux. Comme décrit dans la section dédiée, il nous semble difficile de les éviter, et cela fait donc de cette règle une information supplémentaire pour l'équipage, que ce dernier peut corréler avec d'autres pour savoir si une cyberattaque est en cours ou non.

6.3 Améliorations futures

Cette section présente des angles de recherche potentiels dans la suite de ce projet.

L'extension de la plateforme peut se faire dans plusieurs directions. Premièrement, ajouter d'autres sous-réseaux dédiés à d'autres éléments de TO, comme la gestion des ballasts, du

gouvernail, du carburant ou de l'énergie électrique. Au sein de ces sous-réseaux, il serait particulièrement intéressant de pouvoir déployer de vrais PLC, plutôt que des OpenPLC, afin de pouvoir mettre en place et détecter des attaques contre le PLC en lui-même en plus d'attaques sur le réseau. Cela permettrait aussi de relativement facilement déployer d'autres protocoles propriétaires avec ces appareils, afin de pouvoir en évaluer la sécurité.

Un autre angle de développement de la plateforme serait celui d'un sous-réseau orienté TI, qui représenterait les ordinateurs de l'équipage ou de passagers. Ces ordinateurs peuvent être un vecteur d'attaque intéressant, particulièrement s'il s'agit de machines portables qui entrent et sortent du sous-réseau.

Ces deux éléments peuvent conduire à la conception de scénarios d'attaques détaillés, plus complexes, et qui incluraient des pivots d'un sous-réseau à l'autre, et la combinaison de plusieurs attaques pour augmenter l'impact sur le navire.

Sur le sujet de notre méthode de détection d'attaques, il serait intéressant d'ajouter une couche de traitement après les alertes Snort pour supprimer les faux positifs liés à l'inertie. Cette couche pourrait être une couche de calcul, qui prendrait en compte l'évolution dans le temps des données, ou bien une couche d'apprentissage automatisé.

Il pourrait aussi être intéressant d'évaluer la capacité d'agents autonomes comme des LLM à trouver les données nécessaires à la rédaction des règles dans de la documentation. Cette solution pourrait rendre bien plus facile à mettre en place notre méthode de détection.

RÉFÉRENCES

[1] Cisco. (2021) Première page de la documentation de snort. [En ligne]. Disponible : https://docs.snort.org/rules/

[2] United Nations Conference on Trade and Development, "Review of maritime transport 2020," United Nations, New York, NY, Technical Report, janv. 2021.

[3] ——, "Review of maritime transport 2024," United Nations, New York, NY, Technical Report, 2024.

[4] Professional Mariner. (2020) Naval dome : Les cyberattaques maritime augmentent de 900 pour cents en trois ans. [En ligne]. Disponible : https://professionalmariner.com/naval-dome-maritime-cyberattacks-up-900-percent-in-three-years/

[5] T. Nakashima, B. Moser et K. Hiekata, "Accelerated adoption of maritime autonomous vessels by simulating the interplay of stakeholder decisions and learning," *Technological Forecasting and Social Change*, vol. 194, 2023.

[6] The Nippon Foundation. (2022) Targeting 2025 for fully autonomous navigation. [En ligne]. Disponible : https://en.nippon-foundation.or.jp/news/articles/2022/20220602-74388.html

[7] Rødseth, Ørnulf Jan and Burmeister, Hans-Christoph, "Developments toward the unmanned ship," dans *Proceedings of International Symposium Information on Ships–ISIS*, vol. 201, 2012, p. 30–31.

[8] Det Norske Veritas, "Maritime cyber priority 2024/25 : Managing cyber risk to enable innovation," Det Norske Veritas, Technical Report, 2024, based on a survey of almost 500 maritime professionals and expert interviews with Wärtsilä, Seatrium, and DNV representatives. Second edition of the Maritime Cyber Priority report series.

[9] L. Martin. (2011) The cyber kill chain. [En ligne]. Disponible : https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

[10] Paul Pols, Fox-IT, Leiden University. (2017) The unified kill chain. [En ligne]. Disponible : https://www.unifiedkillchain.com/

[11] MITRE. (2013) Framwork mitre att&ck. [En ligne]. Disponible : https://attack.mitre.org/

[12] Corporation for Digital Scholarship. (2006) Site web de l'outil zotero. [En ligne]. Disponible : https://www.zotero.org/

[13] U. D. Ani *et al.*, “Design considerations for building credible security testbed : A systematic study of industrial control system use cases,” *Cryptography and Security*, nov. 2019.

[14] Free Software Fundation. (2007) Gnu public licence version 3. [En ligne]. Disponible : https://www.gnu.org/licenses/gpl-3.0.html

[15] Fathom 5. Grace maritime cybertestbed. [En ligne]. Disponible : https://www.fathom5.com/grace

[16] DEF CON. (2025) Site web de la def con. [En ligne]. Disponible : https://defcon.org

[17] G. Longo *et al.*, “Macyste : A virtual testbed for maritime cybersecurity,” *SoftwareX*, vol. 23, p. 101426, 2023. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S235271102300122X

[18] G. Longo et A. Orlich. (2023) Répértoire git du projet macyste. [En ligne]. Disponible : https://github.com/CRACK-MCR/MaCySTe

[19] Annonyme. (2009) Site web du projet bridge command. [En ligne]. Disponible : https://www.bridgecommand.co.uk/

[20] D. Register. (2009) Site web du projet opencpn. [En ligne]. Disponible : https://www.opencpn.org/

[21] F. Sicard, E. Hotellier et J. Francq, “An industrial control system physical testbed for naval defense cybersecurity research,” *2022 IEEE European Symposium on Security and Privacy Workshops (EuroSPW).*, p. 413 – 22, 2022.

[22] A. Amro *et al.*, “Navigation data anomaly analysis and detection,” *Information (Switzerland)*, vol. 13, n°. 3, 2022. [En ligne]. Disponible : http://dx.doi.org/10.3390/info13030104

[23] C. Boudehenn *et al.*, “Navigation anomaly detection : An added value for maritime cyber situational awareness,” *2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, CyberSA 2021*, 2021.

[24] W. Liu *et al.*, “Intrusion detection for maritime transportation systems with batch federated aggregation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, n°. 2, p. 2503–2514, 2023.

[25] N. Pelissero, P. M. Laso et J. Puentes, “Naval cyber-physical anomaly propagation analysis based on a quality assessed graph,” dans *2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, 2020, p. 1–8.

[26] N. Pelissero, J. Puentes et P. M. Laso, “Model graph generation for naval cyber-physical systems,” dans *OCEANS 2021 : San Diego – Porto*, 2021, p. 1–5.

[27] N. Pelissero, P. M. Laso et J. Puentes, “Impact assessment of anomaly propagation in a naval water distribution cyber-physical system,” dans *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2021, p. 518–523.

[28] M. C. Nissov *et al.*, “Analysing cyber-resiliency of a marine navigation system using behavioural relations,” dans *2021 European Control Conference (ECC)*, 2021, p. 1385–1392.

[29] G. L. Babineau, R. A. Jones et B. Horowitz, “A system-aware cyber security method for shipboard control systems with a method described to evaluate cyber security solutions,” dans *2012 IEEE Conference on Technologies for Homeland Security (HST)*, 2012, p. 99–104.

[30] MaxSEA. (2008) Site web de time zero. [En ligne]. Disponible : https://mytimezero.com/

[31] S. Hykes. (2013) Site web de docker. [En ligne]. Disponible : www.docker.com

[32] L. Torvald et J. C. Hamano. (2005) Site web du projet git. [En ligne]. Disponible : https://git-scm.com/

[33] Celso. (2006) Site web du projet scadabr. [En ligne]. Disponible : https://scadabr.org/

[34] T. Ralves. (2014) Site web du projet openplc. [En ligne]. Disponible : https://scadabr.org/

[35] N. Falliere, L. O. Murchu et E. Chien, “W32.stuxnet dossier,” Symantec Security Response, Rapport technique, 2011.