CyberShip: An SDN-based Autonomic Attack Mitigation Framework for Ship Systems

Sahay, Rishikesh; Sepúlveda Estay, Daniel Alberto; Meng, Weizhi; Jensen, Christian D.; Barfod, Michael Bruhn

Published in:
International Conference on Science of Cyber Security

Link to article, DOI:
10.1007/978-3-030-03026-1_14

Publication date:
2018

Document Version
Peer reviewed version

CyberShip: An SDN-based Autonomic Attack Mitigation Framework for Ship Systems

Rishikesh Sahay[1], D.A.Sepulveda[2], Weizhi Meng[1], Christian D. Jensen[1], and Michael Bruhn Barfod[2]

[1] Dept. of Applied mathematics & Computer Science,Technical University of Denmark, DK-2800 Kgs., Lyngby, Denmark
[2] Dept. of Management Engineering, Technical University of Denmark, DK-2800 Kgs., Lyngby, Denmark
{risa,dasep,weme,cdje,mbba}@dtu.dk

Abstract. The use of Information and Communication Technology (ICT) in the ship communication network brings new security vulnerabilities and make communication links a potential target for various kinds of cyber physical attacks, which results in the degradation of the performance. Moreover, crew members are burdened with the task of configuring the network devices with low-level device specific syntax for mitigating the attacks. Heavy reliance on the crew members and additional software and hardware devices makes the mitigation difficult and time consuming process. Recently, the emergence of Software-Defined Networking (SDN) offers a solution to reduce the complexity in the network management tasks. To explore the advantages of using SDN, we propose a framework based on SDN and a use case to mitigate the attacks in an automated way for improved resilience in the ship communication network.

Keywords: SDN · Policy language · Ship system · DDoS attack.

1 Introduction

Development in the ICT has also revolutionized the shipping technology. All the ships' components such as global navigation satellite system (GNSS), Automatic Identification Systems (AIS), Electronic Chart Display Systems (ECDIS) are integrated with the cyber systems. This advancement enhances the monitoring and communication capabilities to control and manage the ship. However, these devices on board are also vulnerable to Distributed Denial of Service (DDoS) attack, jamming, spoofing and malware attacks [4]. Moreover, network devices that are used to propagate signals in the ship are also vulnerable to such attacks. For instance, a DDoS attack on the network could result in the inability to control the engine, bridge, and alarm system endangering the ship. However, mitigation of these network attacks requires crew members to perform manual network configuration using low-level device specific syntax. This tedious, complex and error prone manual configuration leads to network downtime and degradation in the performance of the ship control systems. It motivates us to

design a framework that will be capable of mitigating cyber attacks within ship environment in an automated way. To this end, we propose utilizing the capabilities of SDN architecture to design a framework for mitigating these attacks in an automatic way.

Therefore, in this paper, we attempt to design a framework based on SDN to defend the ship's communication infrastructure against cyber attacks in an automated way, with an attempt to improve the resilience against the attacks. Particularly, decoupling of the control and data plane in the SDN provides the flexibility to simplify the network operation compared to traditional network management techniques, since it facilitates us to express the policies at the controller, which can be enforced into network devices depending on the status of the network [7, 11]. Moreover, our framework offers a high-level policy language to specify the network and security policies, which are translated into low-level rules for the enforcement in the network devices in an automatic way. Especially, the focus of this paper is on mitigating the attacks rather than detection.

The rest of the paper is organized as follows. Section 2 reviews some related work. Section 3 introduces our cyber ship framework and its different components. Section 4 presents a use case showing the applicability of the framework. Finally, Section 6 concludes the paper.

2 Related Work

The widespread adoption of ICT throughout today's ships has led researchers to focus on security breaches within ship's technologies that results in a variety of harmful impacts on ship operation and its crew members. However, the research into ship security is in its early stage and many work focus on identifying potential threats and vulnerabilities [3, 4]. In particular, the guidelines of the BIMCO draw special attention to the different types of cyber attacks exploiting the vulnerabilities in the critical components of the ship [4]. These are management guidelines on how to approach the cybersecurity issue in the context of shipping.

To the best of our knowledge, there are very few works dealing with the protection of the communication infrastructure of the ship from cyber attacks. Babineau et.al. [5] proposed to periodically diverting the traffic through different switches in the network to protect the critical components of the ship. It relies on the redundancy in the design of the ship's communication network to divert the traffic through different paths. ABB a leading company in industrial automation proposed to protect the critical components of the ship in the core of the network that typically requires firewalls to enter from outside [1]. Yunfei et.al. [13] and Chen et.al [12] proposed an architecture which rely on statically deployed access controls, firewall and intrusion detection system (IDS) in the network to mitigate the attacks. Our work aims at proposing a framework to mitigate the attacks in an automated way to improve the resilience of the ship control system and reduce the burden on network operator and crew mem-

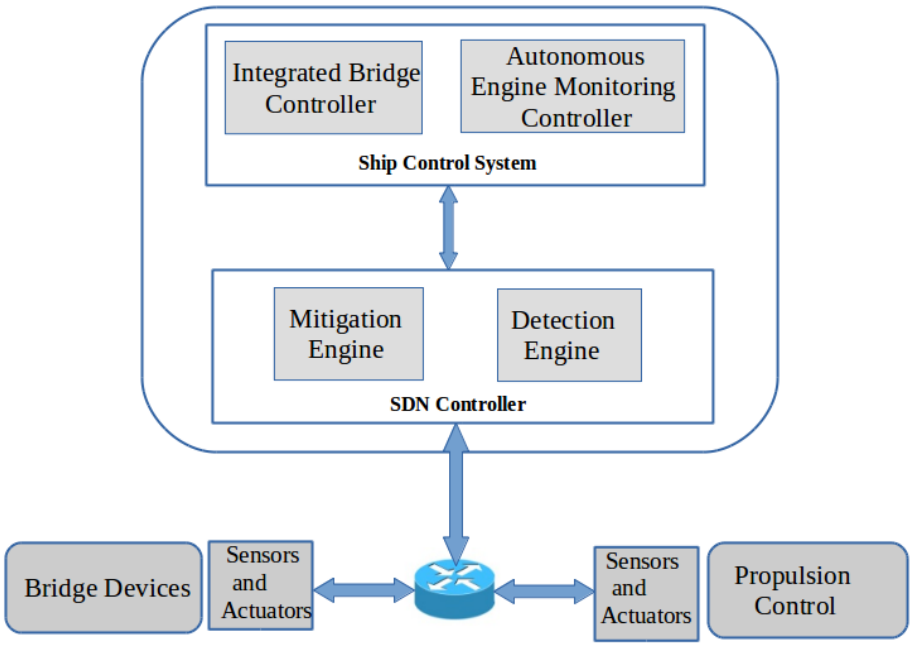

Fig. 1: Framework of the CyberShip

ber of configuring the network devices manually. In Section 3, we present our framework to mitigate the attacks in the ship communication network.

3 SDN Enabled CyberShip Architecture

In this Section, we propose our `CyberShip` framework to mitigate the attacks in an automated way in the ship communication network. The major components are shown in Fig. 1, while the details are given below:

3.1 Components of the Framework

In this Section, we describe the components of our framework. It consists of the different cyber physical components. The components are as follows:

1. **Sensors and Actuators:** Sensors and actuators are attached to the different physical components of the ship related to the bridge, engine and propulsion control devices. These sensors forward the data related to these physical devices to `Integrated Bridge Controller` and the `Autonomous Engine Monitoring Controller` for analysis.

2. **Detection Engine:** It examines the network traffic to identify suspicious and malicious activities. Network operators can deploy mechanisms to classify the suspicious and malicious flows according to their requirements [8, 10]. Upon detection of the suspicious or malicious traffic, it reports a security alert to the mitigation engine.
3. **Mitigation Engine:** It is responsible to take appropriate countermeasures to mitigate the attacks in the framework. It contains a repository consisting of security and network policies defined in high-level language to mitigate the attacks. Depending on the security alert, countermeasure policy is instantiated to mitigate the suspicious or malicious traffic. Details about the high-level policy is given in the Section 3.2. Furthermore, it maintains a list of network paths to reach the different middleboxes (firewalls, IDS, etc.) or to reroute the traffic through different path.
4. **Autonomous Engine Monitoring Controller (AEMC):** It manages the propulsion control, main engine, propeller devices of the ship [2]. Depending on the scenario, it issues the control command to start or stop the propulsion system, increase or decrease the speed of the ship, reroute the ship through different routes. Moreover, it periodically analyses the data received from the sensors of the propulsion, propeller and other components of the engine to check the status of the devices i.e. whether they are working properly or not.
5. **Integrated Bridge Controller (IBC):** It supervises the functioning of the different bridge components of the ship such as a GNSS, ECDIS, radar, and AIS [4]. It receives the data from the sensors of these devices and provide a centralized interface to the crew on-board to access the data. Moreover, it also issues control commands to the `AEMC` to start/stop the propulsion control system, reroute the ship to different routes depending on the information from the bridge devices. In case, it detects the fault or failure on the bridge devices, it notifies the `Mitigation Engine` to divert the network traffic through another route.

3.2 Security Policy Specification

In this section, we describe how the high-level policies are expressed in the mitigation engine module of the `CyberShip` framework. These high-level policies are translated into low-level OpenFlow rules in an automated way for the enforcement in the SDN switches when the need arises. The automated policy deployment eliminates the need of manual configuration for policy enforcement by the crew members.

Grammar of High-level Policy The high-level policy syntax provides the guidelines to the security operators to define the policy. It enables the network operator or the crew member with little IT (Information Technology) expertise to express the security policies into an easy to understand language without getting into low level implementation details.

Listing 1.1: Grammar for the High-level policy language

```
<Policy>=<PolicyID><Target><Rule>
<Target>=<DeviceID><Flow>
<Flow>=<sourceIP><DestinationIP><Protocol><Port>
<Rule>=<Event><Conditions><Action>
<Event>=<Attack_Type>|<Fault_Type>
<Conditions>=<Condition>[<Connective><Conditions>]
<Condition>=<Parameter_Name><Comparison_Operator><Value>
<Connective>=And|Or
<Comparison_Operator>=less than|equal to|greater than|not equal
<Actions>=Block|Forward|Redirect
```

We use Event-Condition-Action (ECA) model for policy representation [6] in CyberShip framework. The reasons for choosing ECA are: (1) it offers flexibility to express different type events which can trigger conditioned actions; (2) Conditions are not needed to be periodically evaluated. Listing 1.1 provides the policy grammar to express the security and network policies in a human readable format, which are specified through the northbound API of SDN controller.

Our policy is composed of a `PolicyID`, `Target` and a set of `rules`. The `PolicyID` assists in uniquely identifying a policy, as there are many different policies in the mitigation engine module. The `Target` specifies the device for which policy should be enforced. Each `rule` is comprised of an event, conditions and an action. The `Event` is an entity which instantiates the policy. Attack and fault type are shown as events in the Listing 1.1. However, it is not limited to these events only, other types of events can also be defined using our policy language.

When an event is triggered, the corresponding conditions are checked against the specified policy. `Condition` is generally a boolean expression that can be evaluated as true, false or not applicable. Not applicable shows that no condition is specified for the event. In our grammar shown in Listing 1.1, `Condition` is specified with the parameter name and a value for the condition.

`Action` represents the high-level decision which should be enforced when the conditions are met for the event. In Listing 1.1 three actions have been specified. High-level action `Drop` is enforced when it is confirmed as a malicious. `Redirect` action is enforced to divert a flow through another path to avoid the congestion or when a flow needs to be processed through middleboxes. Forward action is enforced for the legitimate traffic.

4 Use Case

This section presents a use case exemplifying how the framework enables us to achieve the resiliency by mitigating the attack traffic. We focus on a scenario of mitigating the impact of the DDoS attack targeting the `AEMC` and congesting the network.

The scenario consists of an attacker denoted as A, IBC and AEMC is shown in Fig. 2. Moreover, Mitigation and Detection engine are deployed on a separate controller denoted as C_1 and C_2. Controller C_1 is connected through the switch S_1 and manages all the switches in the network except the switch S_4. Controller C_2 and AEMC are connected through the switch S_4. In the scenario, Detection Engine is deployed close to the AEMC, as the detection can be performed effectively close to the system under protection. In this use case, we assume that the detection is performed based on the threshold set for packet arrival rate, average of bytes per flow, average of duration per flow [8].

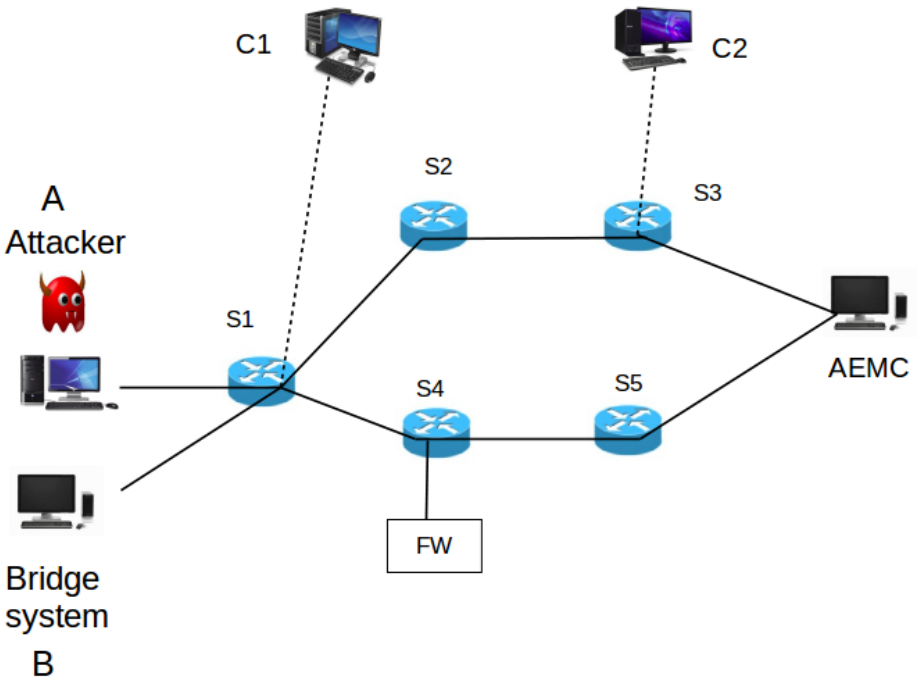

Fig. 2: Configuration of the network topology in the experimentation

IBC sends the messages to the AEMC either to increase or decrease the speed or to reroute the ship through different waypoints. Attacker (A) shown in the scenario which is a compromised machine in the ship communication network, launch the UDP flood traffic towards the AEMC to flood the system and network with bogus packets, so that the AEMC can not receive the messages from the IBC. A firewall (FW) is deployed at the switch S_5 respectively to process the suspicious traffic. Upon detecting an attack Detection engine sends an alert message to the Mitigation Engine module deployed at the controller C_1. It sends an alert in the IDMEF [9] format for processing the UDP flood traffic.

After receiving the alert, `Mitigation engine` module extract the informations from the alert message. Extracted alert information are: source IP of attacker (10.0.0.1), destination IP (10.0.0.3), event type (UDP Flood), flow class which is ”malicious”. Depending on the event type (UDP_Flood) and conditions: flow class (malicious) it gets the high-level action as a ”Redirect_Firewall” from its policy in the mitigation engine. `Mitigation engine` module maintains the information about the different paths along with the middlebox deployment location in the network to divert the flow. The high-level action "Redirect_Firewall" along with flow information (are used by the mitigation engine to configure the rules in the switches to redirect the flow towards the firewall.

5 Discussion

The previous design description and use case demonstrate that our architecture enables us to achieve the dynamic and automated mitigation of attacks.

Multi-path routing approach in the framework provides failover in case of link failure or congestion. Thanks to the global visibility of the network achieved through the SDN controller, flow details, labels and low-level actions can be quickly modified for the concerned flow.

The high-level policy language and translation mechanism in the framework reduces the burden on the crew member to enforce the low-level rules manually. Moreover, it is not required to learn the device specific syntax to express the policies, since our high-level policy language offers to express the policies in human understandable language.

Furthermore, the framework promotes the collaboration between the controllers managing different network devices and the critical components of the ship. For instance, in case of a fault in the engine system, `AEMC` can request mitigation engine to divert the traffic through different path to reach the secondary engine. Moreover, the `Detection engine` in the framework is responsible to detect cyber attacks at the network layer. This reduces the burden on the controllers managing the ship system as they are responsible to manage and control only bridge and engine system.

6 Conclusion and Future Work

In this paper, we presented an SDN-based mitigation framework for ship systems to provide dynamic and automated mitigation of attack traffic for improved resilience. Our framework allows the crew members with little security expertise to specify the network and security policies in a human readable language and automatically translate them into low-level rules for dynamic deployment into data plane devices. By doing so, it hides the low-level complexity of the underlying network to the crew member, who only need to focus on expressing the network and security policies. Another major advantage of the framework is that it allows different controllers to collaboratively manage the critical components of the ship and the underlying networking devices, which

can provide more efficient mitigation of threats. We also presented a concrete use case showing the applicability of the framework as an SDN-based application using multipath routing to increase the resilience against cyber attacks such as DDoS attacks. Our future work will be focused on improving and implementing the framework and its components for further performance evaluation.

References

1. Cyber threat to ships – real but manageable. Tech. rep., ABB (2014)
2. Final Report:Autonomous Engine Room. Tech. rep., MUNIN:Maritime Unmanned Navigation through Intelligence in Network (2015)
3. Guidelines on Maritime Cyber Risk Management. Tech. rep., IMO (2017)
4. The Guidelines on Cyber Security Onboard Ships. Tech. rep., BIMCO (2017)
5. Babineau, G.L., Jones, R.A., Horowitz, B.: A system-aware cyber security method for shipboard control systems with a method described to evaluate cyber security solutions. In: 2012 IEEE Conference on Technologies for Homeland Security (HST). pp. 99–104 (Nov 2012). https://doi.org/10.1109/THS.2012.6459832
6. Bandara, A.K., Lupu, E.C., Russo, A.: Using event calculus to formalise policy specification and analysis. In: Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks. pp. 26–39 (June 2003). https://doi.org/10.1109/POLICY.2003.1206955
7. Ben-Itzhak, Y., Barabash, K., Cohen, R., Levin, A., Raichstein, E.: Enforsdn: Network policies enforcement with sdn. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). pp. 80–88 (May 2015). https://doi.org/10.1109/INM.2015.7140279
8. Braga, R., Mota, E., Passito, A.: Lightweight ddos flooding attack detection using nox/openflow. In: IEEE Local Computer Network Conference. pp. 408–415 (Oct 2010). https://doi.org/10.1109/LCN.2010.5735752
9. Feinstein, B., Curry, D., Debar, H.: The Intrusion Detection Message Exchange Format (IDMEF). RFC 4765 (Oct 2015). https://doi.org/10.17487/rfc4765, https://rfc-editor.org/rfc/rfc4765.txt
10. Mahimkar, A., Dange, J., Shmatikov, V., Vin, H., Zhang, Y.: dfence: Transparent network-based denial of service mitigation. In: 4th USENIX Symposium on Networked Systems Design & Implementation (NSDI 07). USENIX Association, Cambridge, MA (2007)
11. Shin, S., Porras, P.A., Yegneswaran, V., Fong, M.W., Gu, G., Tyson, M.: FRESCO: Modular Composable Security Services for Software-Defined Networks. In: Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS) (2013)
12. Yuanbao, C., Shuang, H., Yunfei, L.: Intrusion tolerant control for warship systems. In: 4th International Conference on Computer, Mechatronics, Control and Electronic Engineering (ICCMCEE 2015). pp. 165–170 (2015). https://doi.org/10.2991/iccmcee-15.2015.31
13. Yunfei, L., Yuanbao, C., Xuan, W., Xuan, L., Qi, Z.: A framework of cyber-security protection for warship systems. In: 2015 Sixth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA). pp. 17–20 (Aug 2015). https://doi.org/10.1109/ISDEA.2015.14