

Article

Overcoming the DDoS Attack Vulnerability of an ISO 19847 Shipboard Data Server

Changui Lee ¹  and Seojeong Lee ^{2,*} 

¹ Korea Conformity Laboratories, Changwon 51395, Republic of Korea; phdculee@gmail.com

² Division of Marine System Engineering, Korea Maritime and Ocean University,
Busan 49112, Republic of Korea

* Correspondence: sjlee@kmou.ac.kr; Tel.: +82-51-410-4578

Abstract: The maritime industry, which transports approximately 90% of the world's goods, plays a crucial role in the global economy. However, increasing reliance on digital technologies has made the industry vulnerable to cybersecurity threats that may compromise the safety and security of maritime operations, thereby potentially affecting global supply chain integrity and public safety. This study examines the vulnerability of the ISO 19847:2018 standard shipboard data server to distributed denial-of-service (DDoS) attacks and proposes a method to mitigate this vulnerability. To this end, we propose modifications to the MQTT v5 protocol used by the shipboard data server, which provides streaming data-transfer services, and conduct verification experiments. These modifications allow the shipboard data server to control the frequency of messages published by the MQTT publisher, thereby preventing it from being overwhelmed by massive amounts of traffic in the event of a DDoS attack. Therefore, the proposed method can enhance the overall cybersecurity of the maritime sector by preventing the misuse of onboard MQTT publishers and reducing the impact of DDoS attacks.

Keywords: ISO 19847; shipboard data server; MQTT; cybersecurity; DDoS



Citation: Lee, C.; Lee, S. Overcoming the DDoS Attack Vulnerability of an ISO 19847 Shipboard Data Server. *J. Mar. Sci. Eng.* **2023**, *11*, 1000. <https://doi.org/10.3390/jmse11051000>

Academic Editors:
Gerasimos Theotokatos, Yaseen
Adnan Ahmed, Victor Bolbot and
Osiris Valdez Banda

Received: 16 March 2023
Revised: 27 April 2023
Accepted: 5 May 2023
Published: 8 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The maritime industry, which transports approximately 90% of the volume of goods worldwide, is an essential part of the global economy. However, as the industry increasingly relies on digital technologies to manage and monitor vessels, cargo, and ports, it has become vulnerable to cybersecurity threats that may compromise the safety and security of maritime operations, thereby potentially affecting global supply chain integrity and public safety [1–4].

Cybersecurity threats to the maritime sector range from traditional cyberattacks to those enabled by emerging technologies such as artificial intelligence and the Internet of Things. These threats may affect the operational efficiency and profitability of the industry and pose a significant risk to the environment, human lives, and public safety. These threats can disrupt maritime operations, damage equipment, and compromise sensitive information [1–3]. Recent incidents such as the NotPetya attack on Maersk, the ransomware attack on COSCO, and the cyberattack on the Port of Barcelona demonstrate the need for improved cybersecurity practices [1–3].

The International Organization for Standardization (ISO) developed the ISO 19847:2018 standard to address the increasing need for standardized guidelines for shipboard data servers [5]. This standard provides guidelines for the design, implementation, maintenance, and security of shipboard data servers for sharing field data at sea, thereby helping to improve the overall cybersecurity of the maritime sector.

Distributed denial-of-service (DDoS) attacks, which involve overwhelming a website or network with a flood of traffic from multiple sources, are among the most common and damaging cyberattacks [6–13]. Several types of DDoS attacks exist, including volume-based,

protocol-based, and application-layer attacks, which are becoming more sophisticated and difficult to defend against. Organizations can implement various strategies to mitigate the risks of DDoS attacks, such as deploying anti-DDoS hardware and software solutions, monitoring network traffic for anomalies, and implementing best practices for network security.

Recent DDoS attacks in the maritime sector include those on the US Coast Guard Navigation Center in 2019, the Port of Long Beach in 2020, and the Port of San Diego in 2021. These attacks disrupted critical navigation services and caused significant disruption to maritime operations, which highlights the need for enhanced cybersecurity measures in the industry [6,12,14–16].

The ISO 19847:2018 standard considers using the Message Queuing Telemetry Transport (MQTT) protocol for shipboard and onshore communications. If the MQTT protocol has vulnerabilities to DDoS attacks, the shipboard data server of ISO 19847 could also be at risk of security threats. This study examines whether the ISO 19847 shipboard data server that uses the MQTT protocol is vulnerable to DDoS attacks and proposes means to overcome this vulnerability. Sections 2 and 3 investigate the ISO 19847 shipboard data server, and Section 4 assesses the vulnerability of the shipboard data server to DDoS attacks. A method for structurally and behaviorally modifying the MQTT v5 protocol to mitigate its vulnerabilities is proposed in Section 5. In Section 6, the effectiveness of the proposed method is experimentally validated. Section 7 summarizes the study.

2. Related Work

As cybersecurity incidents continue to occur in the maritime industry, there is increasing interest in the field and various studies are being conducted to mitigate these incidents. Al Ali et al. [17] present an overview of cyber-security in the marine transport sector, discussing the opportunities and challenges in implementing effective cybersecurity measures. They identified key legal challenges, such as the lack of a comprehensive legal framework for maritime cybersecurity, and proposed incorporating emerging technologies such as blockchain and artificial intelligence to enhance cybersecurity. Ben Farah et al. [18] systematically surveyed recent advances and future trends in cybersecurity in the maritime industry. They analyzed various cybersecurity approaches and technologies, including risk assessment, intrusion detection systems, and encryption, and discussed their effectiveness and limitations. Sungjae [19] conducted a study on the vulnerability of Korean shipping companies to cybersecurity threats. The author identified gaps and weaknesses in their cybersecurity systems and proposed recommendations for improving cybersecurity, such as increasing employee awareness and training, establishing a cybersecurity framework, and implementing regular assessments. Ashraf et al. [20] surveyed the security threats faced by the maritime industry in the era of the Internet of Things. They highlighted potential threats such as unauthorized access, data breaches, and system failures, and discussed existing cybersecurity solutions and recommendations for improving security.

Chien et al. [21] proposed a novel MQTT 5.0-based over-the-air updating architecture that enhanced security by implementing access control, message integrity, and confidentiality mechanisms. The architecture updated device firmware using the MQTT 5.0 protocol, which supports advanced features such as bi-directional communication, message expiration, and message properties. The authors validated the effectiveness of the proposed architecture through simulations and experiments. Munshi [22] proposed an improvement to the secure transmission flags in the MQTT protocol for smart home applications. The author focused on enhancing the security of MQTT messages transmitted between smart home devices by adding additional flags to indicate the level of security required for each message. The paper presents simulation results to demonstrate the effectiveness of the proposed method in reducing message interception and improving the overall security of smart home systems.

The research on preventing cybersecurity incidents in the maritime sector generally takes a macro perspective, and there have been no studies on the security threats posed by the MQTT protocol in the maritime sector. In related studies in other industries, cy-

bersecurity techniques such as SSL or TLS have been applied to the MQTT protocol to overcome the security vulnerabilities of the services. Applying cybersecurity techniques can make the MQTT protocol more robust against cyber threats, but it also means that the protocol becomes heavier. Moreover, if the MQTT protocol contains fundamental security vulnerabilities, applying cybersecurity techniques alone will not be sufficient to overcome them. Therefore, this study aims to identify the fundamental structural vulnerabilities of the MQTT protocol and propose ways to overcome them.

3. ISO 19847 Shipboard Data Server

The ISO 19847 standard defines the purpose and technical requirements of shipboard data servers to provide a secure, reliable, and efficient means of sharing field data at sea [5,23]. As the amount of data generated onboard ships has increased significantly in recent years, real-time data sharing among different stakeholders, including crews, vessel owners, and shoreside parties, has become critical for safe and efficient operations [1]. However, significant challenges may arise in the implementation of shipboard data servers, such as ensuring data security, reliability, and compatibility with different systems [23–27].

A key objective of the ISO 19847 standard is to ensure the security of shipboard data servers [5]. The standard provides guidelines for protecting shipboard data servers against cyber threats, such as unauthorized access, data breaches, and malicious attacks [3]. ISO 19847 includes guidelines for access control, encryption, and network and physical security. The standard also provides guidance on ensuring the confidentiality, integrity, and availability of data stored on shipboard servers. Data confidentiality guarantees that the data can only be accessed by authorized parties, whereas data integrity ensures that the data are not modified or corrupted during transmission or storage [16,20,25]. Data availability ensures that data can be accessed and used when required. The security of shipboard data servers is critical for the protection of sensitive data, the safety of onboard systems, and the prevention of cybersecurity incidents [2,26,27].

Another important aspect of ISO 19847 is its emphasis on reliability. The standard provides guidance on designing and installing data servers that can withstand shocks and vibrations, extreme temperatures, and exposure to saltwater and other environmental hazards. This is particularly important because ships often operate in harsh maritime environments, which may affect the performance and integrity of shipboard data servers. The standard also outlines maintenance and testing to ensure that shipboard data servers remain reliable.

The ISO 19847 standard also includes guidelines to ensure that shipboard data servers are compatible with different systems. Ships often use various systems and equipment from different manufacturers, and the data generated by these systems may have different formats. The standard details how to ensure that shipboard data servers can support the transfer of data in various formats, including standardized data exchange formats. This includes guidelines for data encoding, compression, and exchange protocols. It is important to ensure compatibility with different systems to promote interoperability and reduce communication errors between the shipboard and external systems [24–27].

The ISO 19847 standard specifies three data transmission services that can be combined with data transmission methods to transfer field data from shipboard data servers to facilitate data exchange between the shipboard and external systems. These services include streaming, request response, and file input and output, allowing for real-time and non-real-time data exchange. The service selection depends on the type of data to be transmitted and the communication requirements of the shipboard and external systems.

Among these services, streaming is related to the MQTT protocol, which is a commonly used messaging protocol for implementing real-time data transmissions [26,28]. The MQTT protocol is ideal for the real-time transmission of time-series data from shipboard systems to external systems, which makes it a natural fit for streaming services. It is a lightweight publish–subscribe protocol that enables the efficient data transmission of data in a compact

binary format, thereby minimizing the bandwidth that is required for transmitting large amounts of data.

ISO 19847 provides a framework for efficient and secure real-time data exchange using the MQTT protocol to implement the streaming service. In this model, the shipboard data server publishes data on a topic on the MQTT broker and the external system subscribes to that topic to receive the data in real time. This enables continuous data transmission, thereby providing real-time situational awareness to shoreside operators, which can enhance the safety and efficiency of maritime operations by allowing the real-time remote monitoring and troubleshooting of shipboard systems.

4. DDoS Attack Vulnerability of Shipboard Data Server

In this section, we examine whether a shipboard data server as defined by ISO 19847 is vulnerable to DDoS attacks. The shipboard data server adopts the MQTT protocol to provide streaming data-transfer services. Unlike traditional client-server architectures, MQTT is structured around a broker, publisher, and subscriber model. As illustrated in Figure 1, the subscribers apply to the broker server for the topics to which they wish to subscribe, whereas the publishers post messages on topics to be delivered to subscribing clients by the broker [5,28].

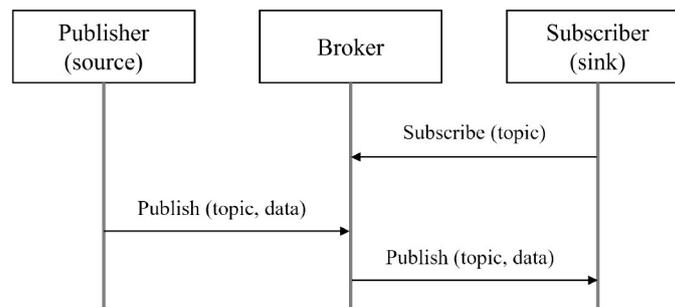


Figure 1. MQTT communication method.

The MQTT protocol only requires the broker server to act as an intermediary for message delivery. Therefore, if numerous packets are received simultaneously, an accumulation of resource tokens may occur in the message queue over time, which will result in service delays, as depicted in the time-Petri net diagram in Figure 2. Here, the square represents the state of the process, and the filled circle represents a resource token.

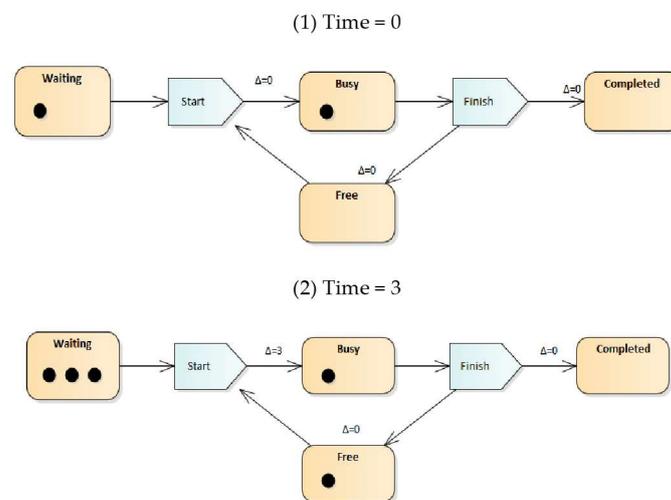


Figure 2. Time-Petri net diagram of high publishing situations.

In particular, if multiple shipboard systems (publishers) are exploited as zombie PCs in a DDoS attack with the intention of disrupting the shipboard data server (broker) services, the network and data server of the ship may be significantly affected. The massive amount of traffic generated by an attack could potentially lead to a shutdown. A DDoS attack may have catastrophic consequences on the network and data server, thereby rendering critical systems and applications inoperable and leaving the ship vulnerable to further attacks, which will compromise its safety. The ship crew may be unable to communicate with the outside world or access essential information, thereby placing their lives at risk.

5. Modification of MQTT v5 Protocol for Overcoming DDoS Attack Vulnerability

In this section, the structure of the MQTT protocol for the streaming services of the shipboard data server is examined in detail to determine its DDoS vulnerability. We investigate the structure of the MQTT protocol to understand the specific vulnerability to DDoS attacks and propose modifications to MQTT from a structural and behavioral perspective to mitigate the identified vulnerability.

5.1. Overview of MQTT v5 Protocol

The MQTT v5 protocol packet structure comprises a fixed header of two bytes and a variable header and payload that may be included depending on the control packet type, as shown in Figure 3 [28].

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Mandatory: Fixed header							
Byte 2								
Byte 3 ... Byte n	Optional: Variable header							
Byte n+1 ... Byte m	Optional: Payload							

Figure 3. Structure of MQTT v5 protocol [28].

The structure of the fixed header is shown in Figure 4, where the first byte consists of a four-bit MQTT control packet type that distinguishes the packet type, and a four-bit flag value that represents the flag value of the header based on the packet type. The second byte represents the remaining number of bytes in the packet after the current byte.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	MQTT control packet type				Flags specific to each MQTT control packet type			
Byte 2	Remaining length							

Figure 4. Structure of fixed header.

A total of 16 MQTT control packet types in the fixed header are defined, as indicated in Table 1, including connection, publish, subscribe, and other message exchange types that allow for flow control.

Table 1. MQTT control packet types [28].

Name	Value	Description
Reserved	0	Reserved
CONNECT	1	Connection request
CONNACK	2	Connect acknowledgment
PUBLISH	3	Publish message
PUBACK	4	Publish acknowledgment (QoS 1)
PUBREC	5	Publish received (QoS 2 delivery part 1)
PUBREL	6	Publish release (QoS 2 delivery part 2)
PUBCOMP	7	Publish complete (QoS 2 delivery part 3)
SUBSCRIBE	8	Subscribe request
SUBACK	9	Subscribe acknowledgment
UNSUBSCRIBE	10	Unsubscribe request
UNSUBACK	11	Unsubscribe acknowledgment
PINGREQ	12	PING request
PINGRESP	13	PING response
DISCONNECT	14	Disconnect notification
AUTH	15	Authentication exchange

When examining the response packet for a connection request, the CONNACK packet in the control packet type of the fixed header has a value of 0x02, indicating the CONNACK packet type, as illustrated in Figure 5 [28].

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	MQTT control packet type (CONNACK = 0x02)				Reserved (set to 0)			
	0	0	1	0	0	0	0	0
Byte 2	Remaining length							

Figure 5. Structure of CONNACK in fixed header.

The CONNACK packet in the variable header includes the connect acknowledge flags of one byte and connect reason code of one byte, which represent a response to the connection request and a variable-length property that allows for various settings, respectively, as indicated in Figure 6 [28].

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	Connect acknowledge flags							
	0	0	0	0	0	0	0	Session present
Byte 2	Connect reason code							
Byte 3 ... Byte n	CONNACK properties							

Figure 6. Structure of CONNACK in variable header.

The CONNACK properties field, which is depicted in Figure 7, is composed of the length of the properties, property identifier, and property value pairs. Pairs of property identifiers and values can be repeated [28].

The server (broker) can set properties to manage sessions, keep alive, and manage QoS for clients (publishers and subscribers), as indicated in Table 2. By including the properties of the response message, the broker can inform the client of the session and connection details [28].

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	Length of properties							
Byte 4	Property identifier							
Byte 5 ... Byte k	Property value (length depends on property)							

Figure 7. Structure of CONNACK properties in variable header.

Table 2. Properties of CONNACK.

Identifier		Name	Type
Dec	Hex		
17	0x11	Session Expiry Interval	4-byte integer
18	0x12	Assigned Client Identifier	UTF-8 encoded string
19	0x13	Server Keep Alive	2-byte integer
21	0x15	Authentication Method	UTF-8 encoded string
22	0x16	Authentication Data	Binary data
26	0x1A	Response Information	UTF-8 encoded string
28	0x1C	Server Reference	UTF-8 encoded string
31	0x1F	Reason String	UTF-8 encoded string
33	0x21	Receive Maximum	2-byte integer
34	0x22	Topic Alias Maximum	2-byte integer
36	0x24	Maximum QoS	Byte
37	0x25	Retain Available	Byte
38	0x26	User Property	UTF-8 string pair
39	0x27	Maximum Packet Size	4-byte integer
40	0x28	Wildcard Subscription Available	Byte
41	0x29	Subscription Identifier Available	Byte
42	0x2A	Shared Subscription Available	Byte

5.2. Proposed Method for Overcoming DDoS Attack Vulnerability

We propose modifications to the MQTT protocol, as illustrated in Figure 8, to mitigate the risks of DDoS attacks and prevent the misuse of onboard MQTT publishers. The proposed method is simple. The shipboard data server (MQTT broker) only acts as a mediator for messages published by the shipboard system (MQTT publisher), and thus has no authority to control the rapid transmission of messages due to DDoS attacks. We set a minimum transmission interval to give the shipboard data server (MQTT broker) the authority to control messages.

The proposed method for mitigating the risk of DDoS attacks and preventing the abuse of MQTT publishers on ships provides a shipboard data server (MQTT broker) with control over the frequency of publishing messages. That is, the shipboard data server should be able to set the message transmission frequency of the shipboard system (MQTT publisher) explicitly. Therefore, even if the shipboard system (MQTT publisher) is compromised by a DDoS attack using zombie PCs, messages will not be published faster than the set frequency. If messages are published faster than the set frequency, the server will be regarded as being infected with a DDoS and access will be blocked to prevent DDoS attacks. By allowing the shipboard data server to regulate the message-publishing rates, the impact of a DDoS attack on the network of a ship can be minimized, thereby preventing the system from being overwhelmed by a massive amount of traffic.

It is crucial to regulate the publisher publish rates to prevent DDoS attacks on a broker. Table 3 lists the properties that can be used to achieve this goal. By defining the minimum publish interval property, publishers can ensure that they do not send data too quickly or overwhelm the brokers. This property can be adjusted to suit the specific needs of the broker applications and capabilities. Note that DDoS attacks can occur when a large number of messages are sent to a broker in a short period, causing it to crash or become

unresponsive. The regulation of the publisher publish rate is a proactive approach for preventing DDoS attacks and ensuring that the broker can handle the amount of data being sent.

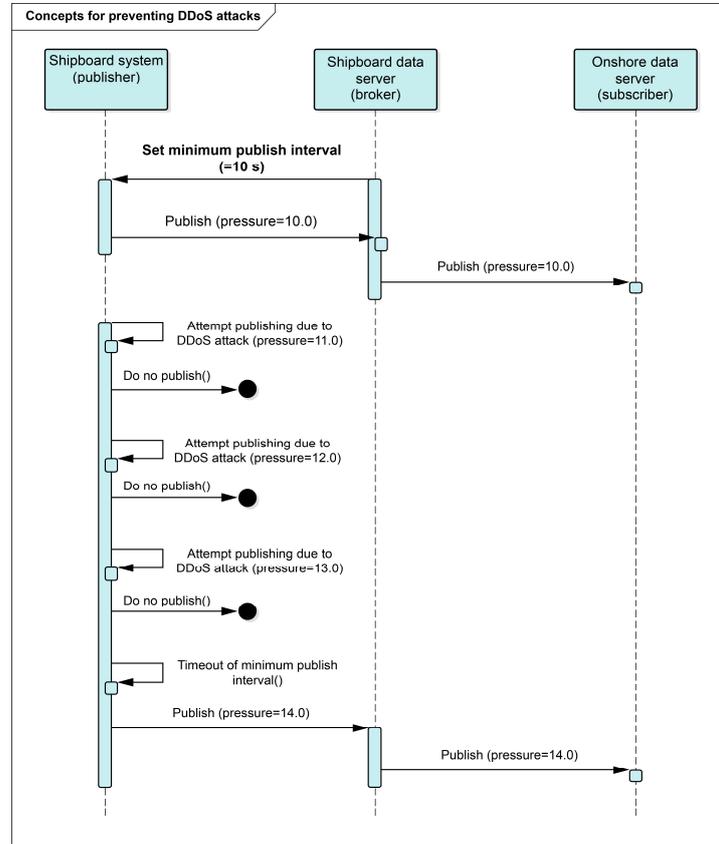


Figure 8. Concepts for preventing DDoS attacks.

Table 3. Additional property definition.

Identifier		Name	Type
Dec	Hex		
101	0x65	Minimum publish interval	Byte

Once the minimum publish interval property has been defined, it can be structured in the variable header of CONNACK properties, as depicted in Figure 9.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 4	Property identifier (0x65)							
	0	1	1	0	0	1	0	1
Byte 5	Setting value of minimum publish interval (seconds)							

Figure 9. Structure of minimum publish interval property in variable header of CONNACK.

Subsequently, the defined properties are placed into the CONNACK properties structure to create a completed CONNACK packet, as shown in Figure 10.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	MQTT control packet type (CONNACK = 0x02)				Reserved (set to 0)			
	0	0	1	0	0	0	0	0
Byte 2	Remaining length (5 bytes)							
	0	0	0	0	0	1	0	1
Byte 3	Connect acknowledge flags (clean start response)							
	0	0	0	0	0	0	0	0
Byte 4	Connect reason code (success = 0x0)							
	0	0	0	0	0	0	0	0
Byte 5	Length of properties (2 bytes)							
	0	0	0	0	0	0	1	0
Byte 6	Property identifier (minimum publish interval = 0x65)							
	0	1	1	0	0	1	0	1
Byte 7	Setting value of minimum publish interval (10 s)							
	0	0	0	0	1	0	1	0

Figure 10. Packet consisting of minimum publish interval property in CONNACK.

The completed packet is composed of seven bytes, consisting of a two-byte fixed header, a five-byte variable header, and no payload. The variable header contains connect knowledge flags and connect reason code fields that respond to the connection request. Furthermore, a property is set to regulate the minimum publish interval, with a time of 10 s. MQTT can ensure that the broker is protected against DDoS attacks by using this packet to regulate the publisher publish rate and create CONNACK packets with defined properties.

The publisher can set the minimum publish interval when attempting to connect to the broker and receiving a response to prevent DDoS attacks, as illustrated in Figure 11. If the connection is accepted, the broker responds with a CONNACK packet that includes the minimum publish interval property. However, if the connection is rejected, there is no need to set the minimum publish interval, so this property is excluded from the response.

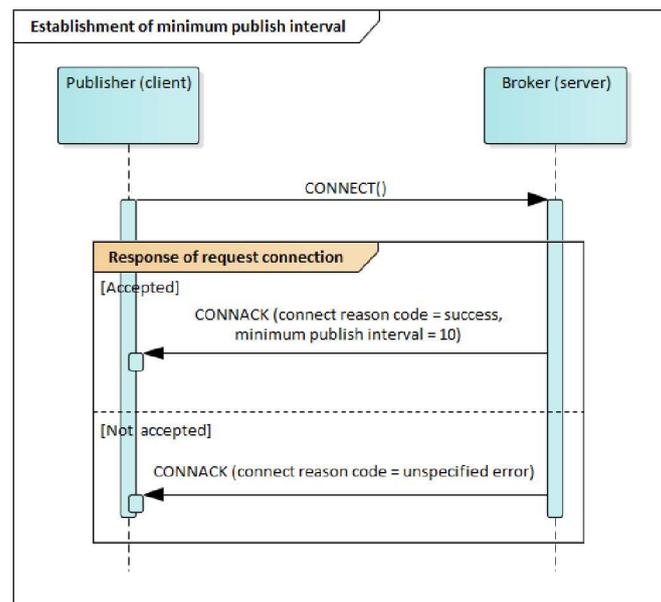


Figure 11. Establishment of minimum publish interval.

6. Experiment and Validation

Various open-source libraries implement MQTT. However, in this study, the source code of the Mosquitto open-source library (mosquitto-2.0.15) was used and modified to fit the proposed design. The Mosquitto library source code is written in C for Linux-based systems. Thus, a Linux development environment was established, and the source code was modified and recompiled to generate binary files.

Two 100-Mbps-supported L4 switches and five shipboard system simulators, along with one shipboard data server and one onshore data server, were prepared, and an isolated network was established, as shown in Figure 12, to simulate a DDoS attack. The number of shipboard systems may vary depending on the size of the ship and equipment configuration, but in this study we created a scenario with a total of 50 shipboard systems by generating 10 virtual shipboard systems on each physical device owing to limited resources.

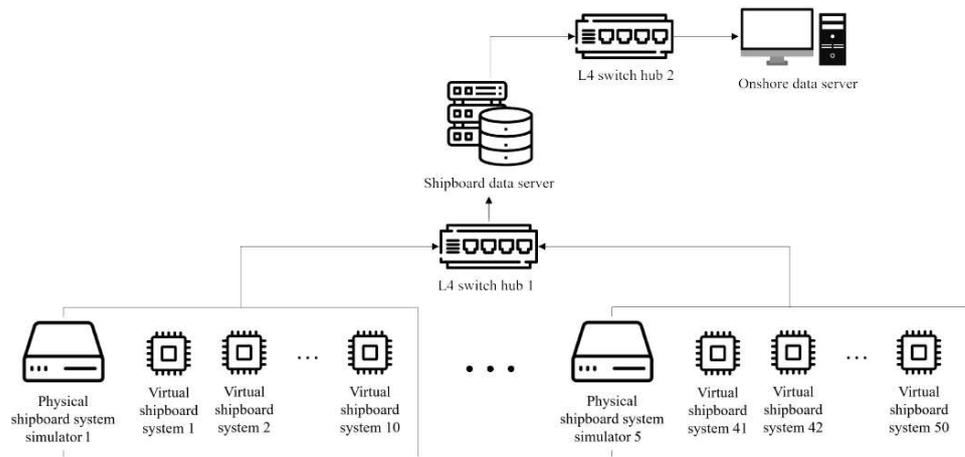


Figure 12. Diagram of experimental environment.

As the purpose of this experiment was to investigate the impact of DDoS attacks on a shipboard data server, the server specifications were important. The specifications of the shipboard data server are presented in Table 4.

Table 4. Specifications of shipboard data server.

Equipment	Specification
CPU	Intel(R) Xeon(R) E-2224 3.5 GHz (4 CPUs)
RAM	16 GB
Storage	SSD 1 TB + HDD 16 TB
Network	100 Mbps Ethernet
OS	Ubuntu 20.04 LTS

A DDoS attack was attempted twice, and the scenario was divided into three phases: preparation, normal operation, and DDoS attack, as outlined in Table 5. The proposed method was not used in the first DDoS attack. In the second attack, the proposed method was used by setting the minimum publish interval attribute in the CONNACK packet.

The scenarios with and without setting the minimum publish interval property were measured 10 times each, and the results are summarized in Tables 6 and 7.

When the minimum publish interval attribute was not used, the average CPU usage was 39.7%, the memory usage was 1937.1 MB, and the network usage was 30.7%. In the second and ninth experiments, the values were measured until the MQTT service was unexpectedly terminated for unknown reasons. Considering that the server used in the experiment had high specifications and operated alone without any other services, it can be observed that it consumed substantial resources.

Table 5. DDoS attack scenario.

Phase		Action
Phase 1 (preparation)	1–1	The shipboard data server boots up and waits for connections.
	1–2	The onshore data server connects to the shipboard data server and requests a subscription.
	1–3	Virtual shipboard systems 1 to 50 attempt to connect to the shipboard data server.
	1–4	The onshore data server responds with a CONNACK packet after approving the connection (with or without the minimum publish interval attribute in the CONNACK packet).
Phase 2 (normal operation)	2–1	The virtual shipboard systems send publish packets once per second.
	2–2	The CPU, memory, and network usage are monitored for 10 min to achieve a steady state.
Phase 3 (DDoS attack)	3–1	The virtual shipboard systems continuously send publish packets without pausing.
	3–2	The CPU, memo, and network usage are measured every 10 s for 1 h.

Table 6. Results of DDoS attack simulation without minimum publish interval.

	CPU Usage	Memory Usage	Network Usage	Remark
1st	38.5%	1837.6 MB	31.8%	
2nd	41.7%	1938.8 MB	33.7%	S/W shut down for unknown reason.
3rd	36.8%	1918.7 MB	29.8%	
4th	38.6%	1899.8 MB	31.4%	
5th	40.8%	1928.1 MB	30.5%	
6th	39.4%	1985.6 MB	29.1%	
7th	41.1%	1999.8 MB	28.8%	
8th	39.9%	1894.2 MB	31.1%	
9th	40.7%	2002.3 MB	30.2%	S/W shut down for unknown reason.
10th	39.2%	1966.5 MB	30.6%	
Average	39.7%	1937.1 MB	30.7%	

Table 7. Results of DDoS attack simulation with minimum publish interval.

	CPU Usage	Memory Usage	Network Usage
1st	3.7%	79.8 MB	2.1%
2nd	3.1%	80.6 MB	2.2%
3rd	3.8%	81.6 MB	2.1%
4th	3.7%	78.8 MB	2.6%
5th	4.2%	83.4 MB	1.9%
6th	4.0%	81.5 MB	2.4%
7th	3.9%	79.8 MB	2.2%
8th	3.8%	77.9 MB	2.4%
9th	4.3%	82.2 MB	2.6%
10th	3.9%	81.3 MB	2.7%
Average	3.8%	80.7 MB	2.3%

When the minimum publish interval attribute was used, the average CPU usage was 3.8%, the memory usage was 80.7 MB, and the network usage was 2.3%. It can be observed that the resource usage decreased significantly compared to that of the previous experiment.

It can be observed from Table 8 that the use of the minimum publish interval attribute reduced the CPU usage, memory usage, and network usage by approximately 1/10 compared to the scenario without its use.

Table 8. Comparison of experimental results.

	CPU Usage	Memory Usage	Network Usage
Without minimum publish interval	39.7%	1937.1 MB	30.7%
With minimum publish interval	3.8%	80.7 MB	2.3%

It may be tempting to assume that sufficient system resources exist even without setting the minimum publish interval. However, in this experiment, a high-performance shipboard data server was used, and only 50 shipboard systems were involved in the DDoS attack. If a shipboard data server handles more services and is connected to more shipboard systems, its condition could deteriorate significantly. In particular, the second and ninth experiments that were conducted without the minimum publish interval property settings experienced sudden software shutdowns for unknown reasons.

This experiment validated that the configuration of the proposed minimum publish interval property, which enables the shipboard data server to assume a leading role in defining the publishing cycle of the shipboard system, is an effective countermeasure against DDoS attacks on MQTT networks. The experimental results demonstrated the significant advantages of setting this property, as it significantly reduces the CPU usage, memory usage, and network traffic. The shipboard data server could proactively manage the publishing cycle of the shipboard system with the minimum publish interval property, thereby enhancing the security and resilience of the MQTT network against malicious attacks.

7. Conclusions

The maritime industry plays a crucial role in the global economy; however, as it increasingly relies on digital technologies, it has become vulnerable to cybersecurity threats. The ISO 19847:2018 standard provides guidelines for the design, implementation, maintenance, and security of shipboard data servers, but these remain susceptible to DDoS attacks.

The proposed method for mitigating the risk of DDoS attacks on shipboard data servers involves modifications to the MQTT protocol to control the frequency with which messages are published in the data server. We conducted experiments that compared scenarios with and without the minimum publish interval property to validate the effectiveness of the approach. The results demonstrated that the scenario in which the minimum publish interval property was used exhibited significantly lower CPU, memory, and network usage than the scenario in which it was not used. Moreover, in reality, the situation could be much more vulnerable to DDoS attacks than in the experiment. This is because the specifications and operational environment of shipboard data servers in real life could be much worse, and there could be many more shipboard systems involved in DDoS attacks. Additionally, the communication infrastructure of the shipboard network and shipboard-to-onshore communication could be much worse than in the experiment. This study demonstrated that the risk of DDoS attacks can be mitigated by allowing publisher transmission intervals to be actively set by the broker. We hope that ISO 19847 shipboard data servers can become better equipped to provide stable and continuous streaming services based on this research. In the future, we plan to formally propose the suggested method to OASIS or ISO for actively defending against DDoS attacks on shipboard data servers, and to conduct further research on this topic.

Author Contributions: Conceptualization, C.L. and S.L.; methodology, C.L.; software, C.L.; validation, C.L. and S.L.; formal analysis, C.L.; investigation, C.L. and S.L.; resources, C.L.; data curation, C.L.; writing—original draft preparation, C.L.; writing—review and editing, S.L.; visualization, C.L.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Korea Institute of Marine Science & Technology Promotion (KIMST) funded by the Ministry of Oceans and Fisheries, Korea (RS-2023-00238653).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bada, M.; ben Barka, L. Maritime Cyber Security: Risks and Challenges. *J. Marit. Law Com.* **2020**, *51*, 121–128.
2. Bauer, J.; Teymourian, A.M.; Scheepers, F.A. Cybersecurity challenges in the maritime sector: How can port authorities and the shipping industry protect themselves? *J. Transp. Sec.* **2021**, *14*, 1–8.
3. Konstantakopoulos, G.; Theotokas, J.; Panagiotou, M. Cybersecurity in the maritime industry: A review of trends and challenges. *Transp. Res. Part C* **2019**, *106*, 239–251.
4. Baltic and International Maritime Council (BIMCO). The Guidelines on Cyber Security onboard Ships—Version 4. Available online: <https://www.bimco.org/about-us-and-our-members/publications/the-guidelines-on-cyber-security-onboard-ships> (accessed on 27 April 2023).
5. ISO 19847:2018; Shipboard Data Servers—Guidelines for Design, Implementation, Maintenance and Security. International Organization for Standardization (ISO): Geneva, Switzerland, 2018.
6. Gkioulos, V.; Oikonomidis, N.; Kotsopoulos, E. DDoS attacks in maritime: Lessons learned and prevention. *J. Cybersecur.* **2020**, *6*, 368–382.
7. Tan, J.; Shashaani, S.; Moore, T. Evaluating the impact of cyber security threats to maritime transportation systems. *J. Transp. Sec.* **2019**, *12*, 27–46. [[CrossRef](#)]
8. Roderick, S.; Chow, Y. Cybersecurity in maritime transportation systems. In *Maritime Cybersecurity*; Springer: Cham, Switzerland, 2021; pp. 27–38. [[CrossRef](#)]
9. Rehman, M.H.U.; Choo, K.K.R. Threats and challenges in maritime cybersecurity. *J. Inf. Sec. Appl.* **2021**, *62*, 102797. [[CrossRef](#)]
10. Kant, K.; Dhillon, G.; Maynard, S.B. Cyber risk management in maritime supply chain: A framework for port resilience. *J. Supply. Chain. Manag. Logist. Procure.* **2019**, *2*, 99–111. [[CrossRef](#)]
11. Ouyang, M.H.; Li, Q. A case study on recent cyber security incidents in the maritime sector. In Proceedings of the International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Glasgow, UK, 15–19 June 2020; pp. 1–7.
12. Zhu, K.Q.; Yang, X.P.; Chen, C.Z. Research on DDoS attack and defense strategies in maritime information system. In Proceedings of the International Conference on Applied System Innovation (ICASI), Chiba, Japan, 13–17 April 2018; pp. 1115–1118.
13. Wallace, W.A.; Chow, J.H. Comparison of cyber security guidelines for the maritime industry. In Proceedings of the IEEE 11th International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Oxford, UK, 3–4 June 2019; pp. 1–7.
14. Wang, L.; Han, X. The application of artificial intelligence in cybersecurity for maritime transportation. In Proceedings of the 3rd International Conference on Information Management (ICIM), Geneva, Switzerland, 24–27 March 2019; pp. 362–367.
15. Edmunds, M.L.; Waidner, M.C. Designing and deploying AI for cyber security. *IEEE Sec. Priv.* **2019**, *17*, 95–99.
16. Gomes, L.S.A.; Vieira, L.F.M.; Ribeiro, F.A.; Alves, T.T. Security in the maritime environment: A survey. *IEEE Access* **2018**, *6*, 13813–13825.
17. Al Ali, N.A.R.; Chebotareva, A.A.; Chebotarev, V.E. Cyber security in marine transport: Opportunities and legal challenges. *Pomorstvo* **2021**, *35*, 248–255. [[CrossRef](#)]
18. Ben Farah, M.A.; Ukwandu, E.; Hindy, H.; Brosset, D.; Bures, M.; Andonovic, I.; Bellekens, X. Cyber security in the maritime industry: A systematic survey of recent advances and future trends. *Information* **2022**, *13*, 22. [[CrossRef](#)]
19. Kim, S. A Study on the Vulnerability of KOREAN Shipping Companies to Cybersecurity Threats. Master Dissertation, World Maritime University, Malmö, Sweden, 2021.
20. Ashraf, I.; Park, Y.; Hur, S.; Kim, S.W.; Alroobaea, R.; Zikria, Y.B.; Nosheen, S. A survey on cyber security threats in IoT-enabled maritime industry. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 1–14. [[CrossRef](#)]
21. Chien, H.Y.; Wang, N.Z. A Novel MQTT 5.0-Based Over-the-Air Updating Architecture Facilitating Stronger Security. *Electronics* **2022**, *11*, 3899. [[CrossRef](#)]
22. Munshi, A. Improved MQTT secure transmission flags in smart homes. *Sensors* **2022**, *22*, 2174. [[CrossRef](#)] [[PubMed](#)]
23. Zhang, L.; Wang, Q.; Xie, L.; Cai, L. Cybersecurity of shipboard networked systems: A review. *Ocean Eng.* **2019**, *183*, 140–152.
24. Liu, X.; Bai, Y. Review of the development of marine big data and its application in ship operations. *J. Mar. Sci. Eng.* **2019**, *7*, 299.

25. Bolbot, V.; Methlouthi, O.; Banda, O.V.; Xiang, L.; Ding, Y.; Brunou, P. Identification of cyber-attack scenarios in a marine Dual-Fuel engine. *Trends Marit. Technol. Eng.* **2022**, *1*, 503–510.
26. Kanwal, K.; Shi, W.; Kontovas, C.; Yang, Z.; Chang, C. Maritime cybersecurity: Are onboard systems ready? *Marit. Policy Manag.* **2022**, 1–19. [[CrossRef](#)]
27. Kechagias, E.P.; Chatzistelios, G.; Papadopoulos, G.A.; Apostolou, P. Digital transformation of the maritime industry: A cybersecurity systemic approach. *Int. J. Crit. Infrastruct Prot.* **2022**, *37*, 100526. [[CrossRef](#)]
28. Eclipse Foundation. MQTT Version 5.0—OASIS Standard. Available online: <http://docs.oasis-open.org/mqtt/mqtt/v5.0/cos01/mqtt-v5.0-cos01.pdf> (accessed on 27 April 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.