

# Attack Simulation on Data Distribution Service-based Infrastructure System

Basem Al-Madani<sup>1</sup><sup>a</sup>, Hawazen Alzahrani<sup>1</sup><sup>b</sup> and Farouq Aliyu<sup>2</sup><sup>c</sup>

<sup>1</sup>Department of Computer Engineering, King Fahad University of Petroleum and Minerals, Dhahran, Saudi Arabia

<sup>2</sup>Center for Excellence in Development of Non-Profit Organization, KFUPM, Dhahran, Saudi Arabia

**Keywords:** Data Distribution Service, Quality of Service Policy, QoS, Security, Middleware, Real-Time, Critical Systems, Iiot.

**Abstract:** Data Distributed Service (DDS) is a widely used publish-subscribe-based middleware protocol for real-time machine-to-machine communication. Many critical infrastructure systems employ DDS for Real-Time applications. These DDS-based systems must operate effectively. This study examined the possibility of manipulating or improperly configuring DDS to facilitate malicious activities. A client-side attack on a DDS-based system and its consequences were the main topics of the research since DDS systems are isolated from other networks and external users. We investigated two security flaws in DDS in an isolated environment to show how they could be employed to compromise a DDS feature. The manipulation of QoS policy configurations in the DDS system demonstrated that it has become more secure than the early versions.

## 1 INTRODUCTION


The Data Distribution Service (DDS) powers systems like; autonomous vehicles, airports, robots, and military tanks, among others. It has been around for ten years, and its use is constantly growing. DDS is a middleware application that supports the creation of middleware layers for machine-to-machine communication. It follows the publish-subscribe model. This software is essential for embedded devices or applications that need real-time functionality (OMG, 2015).


DDS is a reliable communication layer between sensors, controllers, and actuators. It is maintained by the Object Management Group (OMG) and utilized in many critical applications. DDS is for peer-to-peer and publish-subscribe applications because most services cannot tolerate a single point of failure. The middleware works on multicast for discovery, enabling everything to function without any initial setups (DDS Foundation, 2020).


Generally, the DDS layer is (encapsulated) in the Real-time publish-subscribe (RTPS) packets. In other words, each DDS implementation delivers its own

RTPS implementation. The DDS relies on the RTPS, which is a lower-layer standard protocol. DDS is a data-centric communication protocol that enables developers to create a flexible shared data domain for any application requiring two or more nodes to exchange typed data. DDS and its layers are not available as an off-the-shelf product; it is a library. Developers use to create custom middleware protocols with advanced functionality like custom data types, QoS policies, network partitioning, and authentication (DDS Foundation, 2019; Object Management Group, 2022).

DDS's mechanisms give developers greater flexibility when building their systems (OMG, 2018). However, it introduces potential security issues that attackers may exploit. DDS is a prime target for attackers since it lies at the earliest stage of the software supply chain. Thus, it is simple to lose track of it (ENISA, 2021). Demonstrating an exploit's effects is not as straightforward. The requirements, priorities, and operational conditions will vary depending on the vertical, making it challenging to develop representative attack scenarios. In addition, accessing testbed devices for aggressive research is

<sup>a</sup> <https://orcid.org/0000-0002-5875-154X>

<sup>b</sup> <https://orcid.org/0000-0000-0000-0000>

<sup>c</sup> <https://orcid.org/0000-0003-4481-8851>

restricted due to the significance of the systems where DDS is employed (Maggi, 2022).

This research aims to examine and demonstrate the effect of tampering or misuse attacks on the software configuration of DDS systems. The rest of the paper is organized as follows: Section 2 gives a background of the DDS model - its architecture and security implementation. Section 3 presents our problem statement. Section 4 comprises of literature review on M2M protocols and the implementation of security in DDS. The proposed work and simulation setup is in Section 5, followed by experiment works conducted for this research in Section 6. Section 7 provides an analysis and discussion. Lastly, the conclusion and future work are in Section 8.

## 2 BACKGROUNDS

### 2.1 Data Distribution Service

OMG developed the DDS (DDS Foundation, 2019) as a publish-subscribe middleware standard for distributed systems where participants exchange information in a DDS Domain. A Publisher and a Subscriber can write or read in Domain, respectively. The DDS standard allows an efficient, high-performance, and interoperable data share for mission-critical real-time systems. DDS provides QoS policies and protocols that enable applications to establish communication without; specifying the underlying network architecture, changing publisher and subscriber membership, and intermittent connectivity (DDS Foundation, 2020).

The OMG DDS specification comprises two layers: the DDSI (DDS Interoperability Wire Protocol) layer, which describes the layer above network transport, and the DDS layer, which provides Data-Centric Publish-Subscribe (DCPS) application communication behavior with each other and defines QoS policies (Object Management Group, 2022).

### 2.2 DDS in Real-Time Systems

Networking protocols and hardware increasingly become heterogeneous, and distributed computer systems become increasingly dynamic, which require loosely connected infrastructures. Traditional software systems do not address these concerns not adequately.

DDS for real-time systems enables timely and reliable collection of data and transmission via a publish-subscribe method that supports dynamic node discovery, topic-based data distribution, and

data stream time-space decoupling (Object Management Group, 2022). A real-time system consists of many networked devices such as platforms, sensors, actuators, and services to connect, which requires a middleware to manage production processes, communication, and decision-making.

One of the implementations of DDS is RTI Connex DDS. It finds application in; mission-critical, ATC, SCADA, C2 systems, and machinery control.

### 2.3 DDS Security

DDS security design avoids Man-in-the-Middle attacks and ensures only members with permissions can publish, subscribe, or access the data. Security standards include Authentication, Access Control, Cryptography, Logging, and Data Tagging. These functionalities are in the plugins (OMG, 2018). DDS gives developers maximum flexibility in building DDS-based systems. Therefore, the DSS makes a low effort to protect the software and its configuration on the host from tampering. Protecting the host containing DDS applications depends on physical security and network isolation from other systems and external users.

Security issues are investigated by identifying potential threats and associated attack vectors, then examined DDS-based systems to see if they are vulnerable to the attacks.

## 3 PROBLEM STATEMENT

DDS middleware gives developers flexibility when designing and building their systems, for example, specifying the communication details between applications (OMG, 2015; OMG, 2018). A published application does not need to know where the data is going, while a subscribed application does not need to know how or from where the data comes. This model and the complexity of DDS exacerbate its security issues (Abaimov, 2021). These security issues include ease of access to all data, the loss of control over data routing and communication paths, and the opportunity for entities to join the network.

Thus, developing methods for identifying malicious activities within a DDS-based system is crucial. However, access to testbeds for offensive cybersecurity research on DDS-based applications is difficult because of the significance of where the middleware is employed. The first step in defending against a cyberattack on a DDS-based critical system is understanding how DDS can be compromised.

## 4 LITERATURE REVIEW

This section introduced previous studies related to this paper in two subsections. The first one is the limitation in middleware for industrial communications and IIoT. The second subsection discusses the previous works that investigated DDS security.

### 4.1 M2M Protocols

Machine-to-Machine M2M communication is the interaction and data exchange between two or more interconnected machines, which enables devices like smartphones, laptops, factory equipment, robots, and autonomous sensors to react and modify their internal processes based on external feedback.

Cybersecurity in the automated production industry mainly focuses on securing organizational and operational boundaries, such as preventing illegal access to the industrial network. Several messaging communication protocols are available for industrial communications and IIoT, including; MQTT, OPC UA, CoAP, SECS/GEM, DDS, and others. Many of these protocols only offer basic security measures. The following subsections provide a brief explanation of the security limitation in the mentioned M2M protocols.

#### 4.1.1 MQTT

Message Queuing Telemetry Transport (MQTT) is a resource-constrained communications protocol designed by IBM. It uses the publish/subscribe interface paradigm. It works effectively in networks where bandwidth requirements must be minimum. It is best suited for machine-to-machine (M2M) communications.

But since MQTT is for lightweight applications, it neither encrypts the header nor the payload. Instead, it transfers data in plaintext, which is insecure. As a result, encryption must be employed as a separate function, such as by Transport Layer Security (TLS), increasing the computational overhead for resource-constrained devices (Chen, 2020; Patel, 2020).

#### 4.1.2 OPC-UA

The Open Platform Communications Unified Architecture (OPC-UA) is an M2M communication protocol designed by OPC Foundation for Industrial Automation. OPC-UA provides security. It includes a variety of security characteristics, such as authentication, integrity, and confidentiality.

Several protection levels include; no security, integrity only, integrity and confidentiality, and security protocols that define encryption and signature cryptographic techniques for secured communication. Generally, OPC-UA provides industrial automation with robust security characteristics (Mühlbauer, 2020).

#### 4.1.3 CoAP

The Constrained Application Protocol (CoAP) is an Internet Application Protocol developed for resource-constrained devices and networks (Iglesias, 2019). CoAP uses DTLS instead of TLS as a security solution because it runs on the UDP transport layer protocol. It also encodes messages in a simple binary format to provide a lightweight reliability mechanism.

Some of the features by which CoAP keeps the connection and prevents link termination in the event of missed or out-of-order packets. Also, it has an additional feature that prevents DoS attacks by requiring a server to issue a verification query to check the source address's authenticity (Rathod, 2017).

#### 4.1.4 SECS/GEM

The SECS/GEM Protocol is a group of connectivity standards developed by the Semiconductor Equipment Materials Initiative (SEMI) (Laghari, 2021). The SECS/GEM protocol is a widely used global industry standard. The transport communication protocol used by SECS/GEM is called High-Speed SECS Message Services (HSMS); it uses TCP.

The SECS/GEM protocol has a weak security mechanism and provides no security features to connect to other network entities. In addition, the host computer and factory equipment exchange messages with no integrity verification. Hence, it creates potential for cyberattacks such as denial-of-service attacks, data tampering attacks, and spoofing attacks.

#### 4.1.5 DDS

The Data Distribution Service DDS is a publish-subscribe, data-centric middleware for real-time systems developed by OMG (DDS Foundation, 2019). The OMG's DDS security requirements provide a robust security architecture suitable for IoT devices. Also, it can work on both UDP and TCP. Given that devices participate in DDS in a distributed manner and that there are numerous brokers, there isn't a single point of failure, making DDS more

resilient and reliable, ensuring system availability (Friesen, 2020).

The main risk to DDS systems is insider attacks. These are attacks perpetrated by individuals who have already been compromised (Michaud, 2018).

## 4.2 DDS Security Research

Researchers have conducted several investigations on DDS security. However, the research has mainly focused on how to enhance DDS's capabilities to meet customer requirements. OMG has created a new DDS Security specification to address DDS security challenges on a large scale and consistently (OMG, 2018). However, it has not addressed any security issues that could result from a hacked host. As a result, even DDS-based systems built with the new DDS Security features are yet vulnerable to client-side attacks.

White et al. (2019) proposed a method for performing passive network reconnaissance on systems that rely on Secure DDS. The authors execute vulnerability excavation offline without actively engaging the targeted system. It opens the system for targeted attacks such as selective denial of service, adversarial databus segmentation, or vendor implementation vulnerability excavation.

Friesen et al. (2020) investigated the security of the RTPS and DCPS, along with the TLS and DTLS protocols. The DDS Security Plugins are as follows: Authentication Service Plugin, Logging Service Plugin, Access Control Service Plugin, and Cryptographic Service Plugin.

Michaud et al. (2018) this study examined the DDS standard and vendor implementations for potential security flaws before demonstrating and validating five malicious attacks that took advantage of the DDS systems' weaknesses using RTI Connex DDS software. The following are the five attacks:

- Anonymous Subscribe and Republish Functionality: It required only knowledge of the transmitted data type and the Ownership QoS policy.
- OWNERSHIP STRENGTH QoS Policy and EXCLUSIVE Data Ownership: It allows the Data Writer with the highest OWNERSHIP STRENGTH parameter to send data samples when the Ownership QoS Policy is EXCLUSIVE.
- OWNERSHIP KIND QoS Policy and SHARED Data Ownership: For this attack, the attacker updates the Publisher's Ownership QoS policy to differ from the original Subscriber. The malicious Subscriber then

infiltrates the network with the same Ownership QoS policy.

- LIFESPAN QoS Policy Causing Immediate Data Expiration: This attack prevents a Subscriber from receiving data samples.
- LocatorList Environment Variable Causing Participant Discovery Domain Misdirection: This attack takes advantage of this environment variable which determines the IP address that belongs to the DDS entities.

Abaimov et al. (2021) presented an empirical study on the simulation of three types of attacks, Malicious Publisher, Malicious Subscribe, and Clone on DDS. Then, the authors use Deep Learning to detect them. The result showed that Deep Learning detects all the simulated attacks using metadata analysis. However, advanced attacks are more difficult to detect.

Park et al. (2021) evaluated DDS. They used Kerberos to authenticate all Publishers and Subscribers by creating tickets. They utilize ROS 2 as a testbed. The results showed that DDS-C improves DDS security by blocking impersonation attacks.

Kim et al. (2021) proposed the ABAC-based security model for DDS and its execution. The model enhances the authorization of participation and nodes in the RTPS discovery mechanism. The results reveal that the approach meets high-tier time criteria in the healthcare and electricity domains.

Maggi et al. (2022) launched an attack against an autonomous-driving mobile robot simulation and then a physical one in a controlled environment. They tested the vulnerabilities in ROS 2 that affect the design and implementations of DDS by DoS attacks. The result shows that the RTI Connex DDS node crashes and causes the ROS 2 node to crash too.

## 5 PROPOSED WORK AND SIMULATION SETUP

DDS systems find application in areas physically isolated from other networks or systems; and are access-controlled (Michaud et al., 2018). The main risk would most likely come from a host compromise created by someone with a basic understanding of the system's use, architecture, and implementation. This research assumed that the attacker had accessed the DDS system through a software environment or physical system. Thus, we concentrate on the attack action to exploit DDS flexibility to customize the QoS.



### 5.1 Selection of Security Issues and Attack Methods

The analysis of DDS security gaps (Michaud, 2017) revealed 60 security flaws in four areas RTPS, DCPS, configuration mechanism, and Transport mechanism. We selected two of these security issues for modeling and demonstration. The issues are in the DCPS QoS Policy area: *Misuse of RELIABILITY and RESOURCE LIMITS DCPS QoS Policies* and *Misuse of DEADLINE and TIME BASED FILTER*. The selection of these issues covers Data Hijacking attacks.

Table 1: DDS QoS Policies.

QoS Policy	Purpose	Conflicting
RELIABILITY	Indicates the level of reliability offered or requested	
RESOURCE LIMITS	Specifies the resources that the Service can consume.	depth <= max_samples_per_instance <= max_samples
HISTORY	Specifies how the service will buffer and send data	
DEADLINE	Specifies maximum waiting time	deadline >= time_based_filter
TIME BASED FILTER	Specifies minimum separation time	

The DDS DCPS specification specifies Quality of Service (QoS) policies, which outline the minimal service levels necessary for communication between DCPS entities and place restrictions on the shared data. This work focuses on the following QoS policies and their conflict: Ownership, Resource-limits, Reliability, Deadline, and Time-based-Filter. Table 1 describes them in detail.

### 5.2 DDS Demonstration and Validation Environment

The following setup was built in an isolated environment to model each DDS security flaw and attack method. The environment consists of; a Virtual Box (to virtualize the network environment), an Ubuntu operation system (as a host environment for DDS software), RTI Connex DDS Shapes Demo application v 6.1.1 (to demonstrate the attack methods), and a Microsoft Windows operating system.

## 6 EXPERIMENT WORK

The demonstration environment of DDS security issues applied on The RTI Connex DDS Shapes Demo application. The Shapes Demo shows how various QoS policies impact data flow by providing simple methods for identifying unique data types as shapes with different features (RTI Shapes Demo, 2019). The Shapes Demo can play the role of both a publisher and a subscriber to simulate DDS Entities on the hosts.

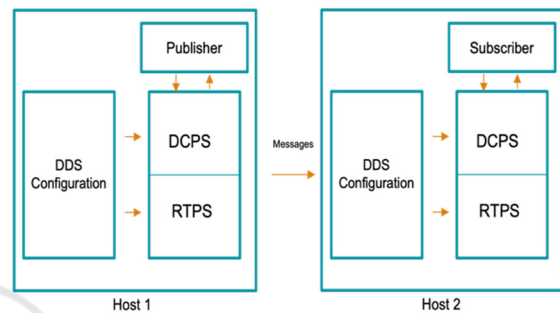


Figure 1: DDS entities interaction.

The domain participants define the QoS for entities: Topic, Publisher, DataWriter, Subscriber, and DataReader. The Publisher specifies and offers the QoS policies to all Subscribers; the Subscribers request the set of QoS policies they require, while the DCPS ensures the policies from both sides match. Communications can only start between the Publisher and Subscriber when their QoSs are consistent.

Figure 1 shows the initial state of the environment. It depicts the interaction between DDS entities and the hosts within the network. This isolated demonstration allows us to analyze malicious changes in the configuration of a DDS publisher. It also allows us to determine whether the domain, the system, or a subscribing application could be affected negatively by the change. The intended DDS-based system has three DDS Entities, with two hosts running a Shapes Demo application as follows:

- Publisher #1: published Red Square with SHARED ownership
- Publisher #2: published Green Triangles with SHARED ownership
- Subscriber #1: subscribed to Red Square and Green Triangles with SHARED ownership

### 6.1 Misuse of RELIABILITY and RESOURCE\_LIMITS DCPS QoS Policies

This DDS security issue occurred using the DDS RELIABILITY and RESOURCE\_LIMITS QoS Policy. The RELIABILITY Indicates the level of reliability offered/requested by the Service that determines whether data can be dropped or delayed (OMG, 2015). The RESOURCE\_LIMITS specifies the resources the Service can consume to meet the requested QoS. This demonstration involved changing the values of a data topic for these QoSs on the publisher side. The QoS Policies change by inserting a DDS configuration file with higher priority.

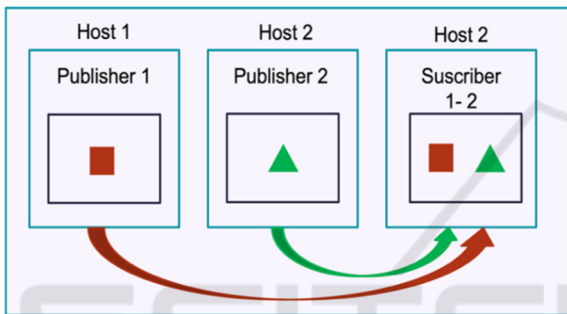


Figure 2: Initial state of DDS entities.

Figure 2 shows the initial state of data distribution of the intended DDS-based system with two publishers and one subscriber. The subscriber listens to data from both publishers. To simulate the attack, the attacker changes the DDS configuration file of the target publisher (Publisher#1).

The Shapes Demo application has a hardcoded file that contains one line - the default QoS policy values to the RTI DDS library. We initially removed the reference to the RTI DDS library's default QoS policy values since the DDS offers flexibility to modify and customize the QoS. Then, a new file named "USER QOS PROFILES.xml" was created, containing new and modified default QoS Policy values. It must be in the same repository as the script that ran the application (RTI Core Libraries, 2016).

Figure 3 shows Publisher#1 (in red) after compromising as a malicious entity. The configuration of other DDS entities (Publisher#2 and Subscriber#1) was unaltered. The HISTORY and RESOURCE\_LIMITS QoS policies affect RELIABILITY QoS. The configuration change involved the depth value of HISTORY, max\_samples\_per\_instance, and max\_samples values of RESOURCE\_LIMITS policy with RELIABLE.

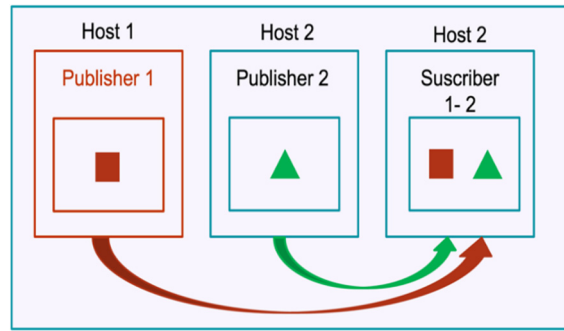


Figure 3: Post exploit state of DDS entities.

Execution of the misuse of DCPS QoS policies resulted in RTPS messages showing on the Publisher side. The message is; Samples lost for Square (count #). On the subscriber side, nothing appears regarding lost samples. The most important observation was that the domain runs using the default QoS file. After stopping, there are two cases: if the QoS policies values were reasonable, it runs by reading from the modified file; otherwise, it will show an error message. The error message says: this file is not going to be used. Therefore, the attacker cannot modify the QoS policy directory and files after the Entity enables in the domain.

### 6.2 Misuse of DEADLINE and TIME\_BASED\_FILTER QoS Policies

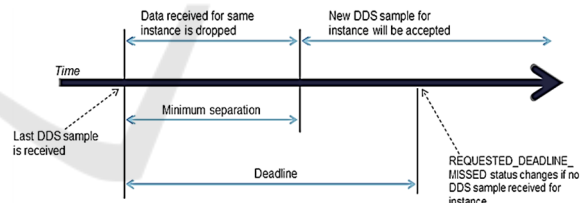


Figure 4: Time constraint of Deadline and Time\_Based\_Filter.

This DDS security issue occurred using the DDS DEADLINE and TIME\_BASED\_FILTER QoS Policies. The DEADLINE for a DataReader expects a new sample updating the value of each instance at least once every deadline period. TIME\_BASED\_FILTER is a filter that allows a DataReader to specify that it is interested only in a subset of the values of the data. Note that it is inconsistent for a DataReader to have a minimum\_separation value longer than its DEADLINE period, as Figure 4 illustrates.

This demonstration involved changing the values of a data topic for these QoS on the publisher side.

We modified the QoS Policies by inserting a DDS configuration file with higher priority. Figure 2 shows the initial state of data distribution of the intended DDS-based system with two publishers and one subscriber (see Figure 3). The reason for implementing this misuse is the restriction of the values for these QoS policies and the ability to modify the values at any time while running. We used the same methods in Subsection 6.1 to simulate the alteration of `TIME_BASED_FILTER` and `DEADLINE` QoS Policies on Publisher #1. The altering of the configuration file involved altering the `minumm_separation` value to be longer than the `deadline` period value.

Execution of this misuse of DCPS QoS policies resulted in the domain running and reading from the default file from the beginning of the experiment. Then, after stopping the middleware, an error message will appear because of the inconsistency of the values. Therefore, the QoS Policy directory and files will not read from the modified file even when starting a new domain with new entities.

## 7 ANALYSIS AND DISCUSSION

This study examined the manipulation of the software and the DDS protocol to harm a DDS-based system. It is significant because a DDS-based system's security and resilience depend on the assumption that the hosts on which DDS applications run are in a trusted zone and thus secure from misuse or abuse. This paper works on two potential client-side attack methods against DDS-based systems. We selected them from 60 DDS security threats in the literature (Michaud, 2017). Also, we used the same techniques to investigate, demonstrate, and model our experiments.

Our research showed that the DDS in RTI Connext v6.1 is secured compared to the previous versions. Misusing the `DEADLINE` and `TIME_BASED_FILTER`, or `RELIABILITY` and `RESOURCE_LIMITS` QoS policies could cause DoS. However, in our experiments, the DDS system did not respond to the altering and modifying of the QoS values. Although the `TIME_BASED_FILTER` QoS policy is modifiable at any time (OMG, 2015), the system would not read the new data while running if it is inconsistent or incorrect.

The security in the current DDS system does not allow reading policies from another file after it has started running, albeit it allows data altering from the high precedence. A client-side attack might expand to

produce more problems and more efficient ways to compromise a DDS-based system.

## 8 CONCLUSION AND FUTURE WORK

In conclusion, this study examines two client-side attack techniques that might affect systems by tampering with or changing their configuration file. DDS structures provide developers with flexibility. But they also raise the potential security issues of malicious attacks on a DDS-based application. DDS is a prime target for attackers because it is located at the beginning of the software supply chain, making it easy to lose track of it. The first step in detecting and defending against a cyberattack on a critical infrastructure system is understanding DDS security issues and how an attacker might use them to compromise a DDS-based system.

For future work, more research is necessary to evaluate the security of various DDS vendor implementations. Real-world examination and evaluation are needed to assess the ease with which client-side attacks can compromise DDS-based systems. In addition, it is necessary to evaluate the damages that could occur if a DDS-based system is compromised.

Another topic that requires further study and development is intrusion detection and prevention-based DDS systems: anomaly-based intrusion detection systems (IDS) could be helpful since it identifies contextual dangers DDS faces, and implementing protection rules for Intrusion Prevention Systems (IPS) can recognize anomalous XML files.

## REFERENCES

- DDS Foundation. (2019). What is DDS? DDS Portal – Data Distribution Services. <https://www.dds-foundation.org/what-is-dds-3/>
- DDS Foundation. (2020). How Does DDS Work? DDS Portal – Data Distribution Services. <https://www.dds-foundation.org/%20how-dds-works/>
- Object Management Group. (2022, April). The Real-time Publish-Subscribe Protocol DDS Interoperability Wire Protocol (DDSI-RTPSTM) Specification. DDS Interoperability Wire Protocol. <https://www.omg.org/spec/DDSI-RTPS>
- OMG. (2018, July). About the DDS Security Specification Version 1.1. OMG. <https://www.omg.org/spec/DDSI-SECURITY/1.1>

- OMG. (2015). About the Data Distribution Service Specification Version 1.4. Object Management Group. <https://www.omg.org/spec/DDS/1.4>
- Kulkarni, A. M., Nayak, V. S., & Rao, P. B. (2012, April). Comparative study of middleware for C4I systems Web Services vis-a-vis Data Distribution Service. In *2012 International Conference on Recent Advances in Computing and Software Systems* (pp. 305-310). IEEE.
- Maggi, F., Vosseler, R., Cheng, M., Kuo, P., Toyama, C., Yen, T. L., & Robotics, A. (2022) A Security Analysis of the Data Distribution Service (DDS) Protocol.
- Chen, F., Huo, Y., Zhu, J., & Fan, D. (2020, November). A review on the study on MQTT security challenge. In *2020 IEEE International Conference on Smart Cloud (SmartCloud)*(pp. 128-133). IEEE.
- Patel, C., & Doshi, N. (2020). A novel MQTT security framework in generic IoT model. *Procedia Computer Science*, 171, 1399-1408.
- Mühlbauer, N., Kirdan, E., Pahl, M. O., & Carle, G. (2020, September). Open-source OPC UA security and scalability. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (Vol. 1, pp. 262-269). IEEE.
- Iglesias-Urkia, M., Orive, A., Urbieto, A., & Casado-Mansilla, D. (2019). Analysis of CoAP implementations for industrial Internet of Things: a survey. *Journal of Ambient Intelligence and Humanized Computing*, 10(7), 2505-2518.
- Rathod, D., & Patil, S. (2017). Security analysis of constrained application protocol (CoAP): IoT protocol. *International Journal of Advanced Studies in Computers, Science and Engineering*, 6(8), 37.
- Laghari, S. A., Manickam, S., & Karuppayah, S. (2021). A review on SECS/GEM: A machine-to-machine (M2M) communication protocol for industry 4.0. *International Journal of Electrical and Electronic Engineering & Telecommunications*, 10(2), 105-114.
- Michaud, M. J., Dean, T., & Leblanc, S. P. (2018, October). Attacking omg data distribution service (DDS) based real-time mission-critical distributed systems. In *2018 13th International Conference on Malicious and Unwanted Software (MALWARE)* (pp. 68-77). IEEE.
- Friesen, M., Karthikeyan, G., Heiss, S., Wisniewski, L., & Trsek, H. (2020). A comparative evaluation of security mechanisms in DDS, TLS and DTLS. In *Kommunikation und Bildverarbeitung in der Automation* (pp. 201-216). Springer Vieweg, Berlin, Heidelberg.
- White, R., Caiazza, G., Jiang, C., Ou, X., Yang, Z., Cortesi, A., & Christensen, H. (2019, June). Network reconnaissance and vulnerability excavation of secure DDS systems. In *2019 IEEE European symposium on security and privacy workshops (EUROS&PW)* (pp. 57-66). IEEE.
- Michaud, M. (2017). *Malicious use of omg data distribution service (dds) in real-time mission critical distributed systems* (Doctoral dissertation).
- Kim, H., Kim, D. K., & Alaerjan, A. (2021). ABAC-based security model for DDS. *IEEE Transactions on Dependable and Secure Computing*.
- Park, A. T., Dill, R., Hodson, D. D., & Henry, W. C. (2021, December). DDS-Cerberus: Ticketing performance experiments and analysis. In *2021 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 1465-1469). IEEE.
- Abaimov, S. (2021). Towards a Privacy-preserving Deep Learning-based Network Intrusion Detection in Data Distribution Services. arXiv preprint arXiv:2106.06765.
- RTI Shapes Demo. (2019). RTI Community. [https://community.rti.com/static/documentation/connext-dds/6.0.0/doc/manuals/shapes\\_demo/RTI\\_Shapes\\_UsersManual.pdf](https://community.rti.com/static/documentation/connext-dds/6.0.0/doc/manuals/shapes_demo/RTI_Shapes_UsersManual.pdf)
- RTI Core Libraries. (2016). RTI Community. [https://community.rti.com/static/documentation/connext-dds/5.2.3/doc/manuals/connext\\_dds/RTI\\_ConnextDDS\\_CoreLibraries\\_UsersManual.pdf](https://community.rti.com/static/documentation/connext-dds/5.2.3/doc/manuals/connext_dds/RTI_ConnextDDS_CoreLibraries_UsersManual.pdf)
- ENISA Threat Landscape 2021. (2021). ENISA. <https://www.enisa.europa.eu/publications/enisa-threat-landscape-2021>