Research Article GU J Sci 36(3): 1050-1069 (2023) DOI: 10.35378/gujs.976496

Gazi University

Journal of Science

http://dergipark.gov.tr/gujs

An Analysis of DoS Attack on Robot Operating System

Elif DEGIRMENCI[*], Yunus Sabri KIRCA, Esra Nergis YOLACAN, Ahmet YAZICI

Eskişehir Osmangazi University, Faculty of Engineering and Architecture, Computer Engineering Department, 26048, Eskişehir, Türkiye

Highlights

- This paper focuses on analysis of DoS attacks for Robot operating system.
- This paper proposes an analysis based on packet loss and delay in transport layer.
- This paper proposes an analysis based on packet loss and delay in application layer.

Article Info

Received: 30 July 2021
Accepted: 18 July 2022

Keywords

Robot operating system
DoS attack
Network security

Abstract

The emergence of robotic technologies has made a significant contribution in industry. Robot Operating System (ROS) is becoming a standard framework for industrial systems uses as a middleware system with many versions. However, the initial design of ROS does not include cyber-security concepts. The intense interest in robot systems, the security concerns and vulnerabilities of these systems have started to attract the attention of attackers. One of these attacks is DoS attack that targeting system availability by slowing down or crashing a service rather than obtaining the information or system. In this study, the impact of DoS attack has been analyzed in various scenarios for both in application and transport layer of the ROS middleware. In the experiments four different volume of DoS attacks are performed in five different experiment scenarios on ROS. To understand the impact of DoS attack, network traffics are monitored using Tshark. The resulting effects measured with some Quality of Service parameters that are delay and packet loss.

1. INTRODUCTION

Industrial robots are one of the key components for digitalization of industry. According to the report of "World Robotics 2020 Industrial Robots" by International Federation of Robotics (IFR), 2.7 million industrial robots operating in factories [1]. The rapid digitalization of industry has caused the concept of cyber-attack to become more common in recent years. In addition to the intense interest in robot systems, the security concerns and vulnerabilities of these systems have started to attract the attention of attackers. Various malware and attack tools have been developed by attackers for the purpose of security attack. These attacks, which pose many hazards such as the loss, alteration or disclosure of your data, can also cause devastating consequences such as system malfunction or out of service. Industrial robotic systems are vulnerable to various security attacks some common types of these can be listed as phishing attacks, malware attacks, web attacks, network attacks.

Robotic Operating System (ROS) is one of the widely used middleware for industrial robotics systems [2]. Along with the various versions of ROS, it has become one of the primary standards for industry. The initial design of ROS does not include cyber-security concepts. While the Master node controls the status of the system, publisher and subscriber can directly communicate with each other due to the design architecture of ROS. Therefore, the ease of use of ROS communication creates vulnerabilities against various attacks like Denial of Service (DoS), Distributed Denial of Service (DDoS), Blackhole, Eavesdropping and data leakage [3]. Attacks on robotic systems on critical environment such as robotic arm using ROS in Tele surgical operation may cause serious safety consequences [4]. In recent years, many studies have been carried out to both eliminate these concerns and take countermeasures [5-7]. In order to ensure security on ROS, technologies such as encrypting messages, making communication more secure, etc. have been developed [5]. In [5], security is improved by controlling the ROS network against attacks such as intrusion,

[*]Corresponding author, e-mail:edegirmenci@ogu.edu.tr

setting policies and restrictions on nodes. There are solutions that provide transparent security at the application level on ROS, explain the vulnerabilities of recent years and how to exploit them [6]. They enable the communication of ROS nodes secure by making the (D)TLS channel secure. ROS also widely used in manufacturing environment, attacks on production could affect the entire flow [7]. In [7], some stealthy product sabotage attacks to a robotic arm in the manufacturing process detected with learning the normal behavior. It is necessary to test evaluated systems in order to find out how much the measures taken provide countermeasures to possible security gaps or to find the security vulnerabilities [8]. In recent years some attack and monitoring tools proposed in the literature for ROS such as ROSPloit [3], ROSChaos [8] and ROSPenTo [8]. Although some security attack tools are developed for ROS in the literature, as the author knowledge there is no tool to include DoS attack to application layer. In this paper, we analyze the effect of DoS attacks against ROS. DoS attack may affect both transportation layer and application layer. In this study, the effect of DoS attack that targets to transport layer and application layer are analyzed. Two type of DoS attack performed. One of the performed one is ROSPloit tool’s DoS attack targets to transport layer. The other performed DoS attack is targeting to the application layer. In ROS middleware both layers are monitored with Tshark. The impact of DoS attacks on the ROS middleware are evaluated according to quality of service criteria i.e. delay and packet loss within the system.

In the following section, literature review of network security and ROS security are given. In the third section, proposed system for DoS attack is given in details. In the fourth section, experimental results are given. The last section, conclude the results of the study.

2. LITERATURE REVIEW

Industry 4.0 revolution can be expressed as a radical change for digital manufacturing by combining various new technologies such as cyber-physical systems, autonomous robots and the Internet of Things. Although, this new era may result many advantages such as efficiency, increase in the quality of product, easy monitoring of systems, the biggest disadvantage is the weakness in terms of system security. Any attack on the communication technologies or sub devices used in the system can affect the functioning of the whole system, causing financial losses as well as major effects on the system itself. In this section, firstly, the literature on network security in terms of communication security in industrial systems is examined. Then, literature studies on ROS security, one of the most popular frameworks used in industrial systems, were examined.

2.1. Network Security

A network is an environment where things, computers or devices are connected and communicating. Protocols such as OSI and TCP/IP are used for communication from one point to another on the network. Network security covers all issues such as detecting any attack on these communication protocols and taking countermeasures to protect against them. One of most common type of network security attacks is DoS attack. A DoS attack targets system availability by slowing down or crashing a service rather than obtaining the information or system. Security attacks can be classified in different ways, such as the action performed by the attack, the attack's domain, and the network protocol stacks [9]. In this context, DoS attacks can be defined as active, external, and multi-layered type of attack.

A DoS attack is an attack from a system that targets the availability of a server, while DDoS attack is an attack where multiple systems target a single system with a DoS attack [10]. In the literature, taxonomy of DoS/DDoS attacks has been carried out in many studies [11-13]. In this work, we examine DoS attacks according to the OSI layer they are performed in, which are the transport layer and the application layer. Some of the most known attacks in these layers are UDP flood, CLDAP, ICMP flood, DNS, NTP attacks [13]. Transport layer DoS attacks mostly target bandwidth or resource depletion using protocols such as UDP (User Datagram Protocol) or ICMP (Internet Control Message Protocol). Manavi [14] divides these attacks into two categories: 1) flooding attacks which sends high volume traffic to exhaust bandwidth, and 2) amplification attack which consumes bandwidth by sending a packet to all IP addresses in the broadcast address range. The protection mechanisms against these attacks are divided into four categories, these are: source-based, network-based, destination based and hybrid based mechanisms.

Application layer DoS attacks appear to have some advantages over other layers. For example: application layer attacks are difficult to detect because they send valid requests rather than fake requests and are performed without causing spikes in traffic [15, 16]. While doing this, it aims to consume CPU or memory resources by using vulnerabilities of protocols such as HTTP, VOIP, DNS instead of overflow attacks [14]. Application layer DoS attacks are more difficult to detect than transport / network layer attacks, so more sophisticated and intelligent defense methods are needed for such attacks. These methods mostly use machine learning to analyze according to application or traffic behaviors [17].

In order to protect industrial systems, detection of these attacks is crucial. Undetected DoS attacks can cause intruders to infiltrate the network, access the control software, and cause unwanted damage due to changes in the operating conditions of the system. For example, failure of a software component in a collaborative robotic system would be extremely dangerous for human robot interactions. Monitoring the effects of DoS attacks within the network enables new approaches to detect these attacks. Therefore, "quality of service" criteria can be used as an indicator of whether the system is working properly or not for systems using ROS framework.

2.2. ROS Security

ROS is one of the widely used middleware for industrial robot control. Although many studies have been developed for network security, ROS security research is quite new. Due to the subscriber-publisher communication structure of ROS, any node is allowed to be subscribe to any topic or each published topic can be listened to not only by the requested node but also who wants to listen. This structure broad some challenges about basic security standards confidentiality, integrity, availability and authenticity of topics and nodes. Since ROS is an open-source meta-operating system, it does not have a common accepted security mechanism.

Encryption or policy creation studies [5, 18-22] are one of the widely used security approach for ROS in literature. In study [19], encryption is used in data transmission to prevent malicious attacks to the ROS node that publishes the position of the KUKA iwa robot arm joint angle. Each common message/topic is encrypted using the encryption key, preventing the publisher from pushing the robotic arm out of the planned position by changing the message and could be threaten environmental safety. In [5], secure data transmission achieves by encryption. Considering the security vulnerabilities arising from the encryption of headers, a secure transmission layer has been created in the transmission layer. Security countermeasures are taken been against man in the middle attacks. Data security is also important in surgical operation robotic applications are use such as Raven II or DaVinci Research kit robots [20]. In [20], especially in the telesurvey area, the security of the entire communication channel and messages/topics between robots is ensured by encryption and authentication. Balsa-Comerón, et al. [18] improves security with encrypted communication with AES algorithm and adds semantic rules with ROSRV [21] framework.

The general drawbacks in encryption studies are that although the message context is encrypted and cannot accessed by unauthorized nodes, still sniffing can be on publisher's activity and publishes frequency. This can be solved by end-to-end encryption of all messages integrated into the ROS itself. Another shortcoming is that malicious broadcasters cannot be prevented from broadcasting messages. Only these messages can be prevented from being received by normal nodes. However, these systems could be target to a DoS attack with a high publishing frequency. Also, another security problem is that a subscriber cannot be prevented from subscribing to random topics. These malevolent activities could get high network load to the system.

ROS tools that are used for attack and monitoring are required to test the system against possible vulnerabilities in the system. In the literature there is a few attack tools that operate manually or automatically perform attacks. To the best of author knowledge, ROSPloit [3], ROSChaos [8] and ROSPenTo [8] are priori attack tools that developed for ROS. ROSPloit is an attack tool that includes many attack methods like DoS, kill node, port exhaust, replace node, change parameter developed for ROS [3]. ROSPloit can be used two purposes: investigation and exploitation. Investigation part is similar to NMAP, which is network level investigation tool, and also works with it. ROSPloit tool scans the open ports in the system and running node names on ROS middleware, and demonstrates whole system connection. The exploitation part is similar to Metasploit that uses network level security vulnerabilities. Exploitation part of the ROSPloit offers performing Man in the Middle attack or sending malicious TCPROS messages without installing ROS. ROSChaos and ROSPenTo are both proposed by Dieber et. al. [8]. ROSChaos tool directly attacks the Master node as a target. Master based attacks are easily detectable, but their effects can be devastating if there is no defense mechanism in the system. ROSPenTo is a tool for system scans and attacks on ROS using commands in ROS. The main idea of ROSPenTo is to interact with the master as little as possible where most attacks occur directly on nodes. It is a .Net-based tool that is used to analyze and manipulate ROS applications. ROSPenTo can be used on all platforms that can run .Net or Mono. Damage to the non-master nodes causes damage only to that node, while damage to the Master node disables the entire system. Therefore, security attacks using these tools mostly targets the Master node in ROS. Comparison of tools are given in Table 1, based on year published, implementation languages, attacks target in the system and target operating layer. As mentioned in table, ROSPenTo uses C# different from the other tools. All examined tools are targeting to the nodes/master/Endpoint in the system and also all mentioned tools run for the transport layer.

Table 1. *The Comparison of the ROS Attack Tools*

Tool	Year	Implementation Languages	Target (Link/Endpoint)	Operating Layer (Application/Transport)
ROSploit	2019	Python	Endpoint	Transport
ROSChaos	2017	Python	Endpoint	Transport
ROSPenTo	2019	C#	Endpoint	Transport

Monitoring the situation of the system is as important as attack tools [21, 23, 24]. ROSMon is a debug tool that can be used in place of the “ROSlaunch” command, that can boot the system directly on itself [24]. Unlike “ROSlaunch”, while the system is running, it can close the desired nodes, rerun them, and view the data flow and operating status. Unlike what happened in “ROSlaunch”, “ROScore” does not work on ROSMon. The main reason for this is to prevent the entire ROS from completely shutting down if an error occurs on the ROSMon. ROSDefender is a tool that includes monitoring the application layer and transport layer. ROSDefender integrates and proposes three parts: ROSDN like SIEM, ROSWatch and Policy Language creation parts are like IPS and robotic system firewall [23]. The tool combines control, monitoring and dynamically generating policies to protect against attacks such as anomalous behavior or malicious huge activities. ROSWatch, referred to as a monitoring tool, takes logs about network traffic and the state of the ROS. ROSRV is a framework that captures messages, monitoring messages and commands passing to the ROS Master through RVMaster node which acts as a secure layer that between nodes and master [21]. This node ensures application layer and transport layer security. The framework also provides a specification language to define access control policies for unintended shutdown actions, unauthorized kill nodes and restrict publishers to publish topics.

3. PROPOSED SYSTEM ARCHITECTURE FOR ROS

DoS attacks may interrupt the service of ROS. A DoS attack can be directly realized on the ROS. On the other hand, the attack can also be realized on the transport layer or the application layer in the network where ROS is running. The later one will have an indirect effect on the ROS. Both direct and indirect attack types will affect the operation of ROS. Monitoring the impact of all DoS attacks on ROS is valuable to determine the quality of service.

In the proposed system, two types of DoS attacks are performed as in Figure 1. The attacks are applied to application layer and transportation layer (Figure 1). One of the DoS attack is ROSPloit tool’s DoS attack targets to transport layer. The other DoS attack that is our proposed DoS attack targets to ROS nodes communication and applied to application layer. Although the ROSPloit tool targets the transportation layer, this attack not directly affect ROS but indirectly effect ROS Master and ROS nodes due to ROS nodes communication requests to ROS Master goes through transport layer. If the ROS Master stops providing service, the joining of nodes to the system and publisher-subscriber request services may also be disrupted. In our implemented ROS-DoS attack targets to application layer. This DoS attack directly targets to ROS. This attack is intended to block or interrupt ROS nodes communication by large number of subscriber to publisher node. ROS-DoS attack could drop ROS nodes to out of service. Since the publish / subscribe request goes through ROS Master, this attack could also affect the transport layer communication. While the attacks occur, logs are recorded with Tshark from the transport layer and application layer. Logs are analyzed for understanding the impact of attacks on different layer's traffic flow.

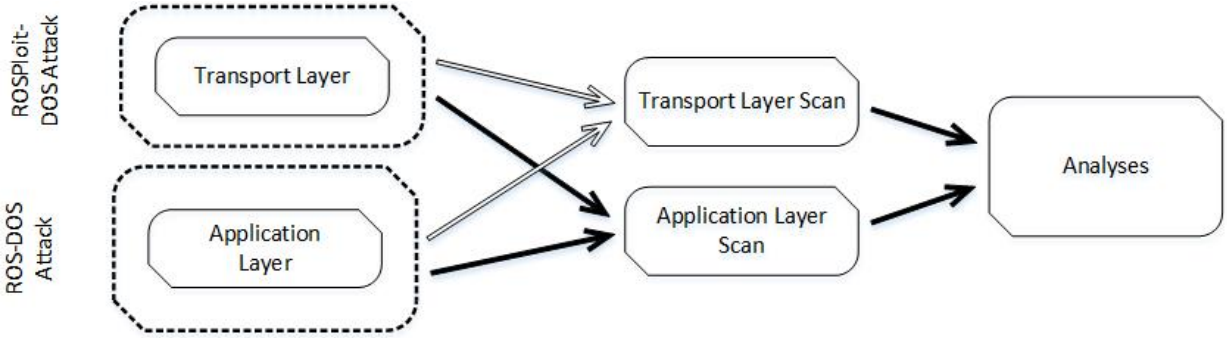

Figure 1. *Proposed system analysis schema*

ROS is increasingly used for autonomous robots in industrial applications. In a smart factory environment, autonomous robot usually performs a task according to given plan [25]. Figure 2 shows a typical ROS architecture for autonomous robots in a smart factory. Autonomous robots track safe and accurate route tracking by sensor's data (Odom, Laser, Lidar, Camera, …) on robots to generate information about the environment. Communication of this system takes place over ROS. The communication flow for an autonomous vehicle to receive routes and send sensor data during a task is given in Figure 2. In the proposed communication scenario ROS Master already runs on the system and autonomous robots both publish and subscribe. Planner node is also publisher/subscriber node to get the sensor data and send the new route to autonomous robot. In a publisher-subscriber communication structure, publisher node publishes messages under a topic and subscriber node tracks data through that topic. In the communication scenario, when autonomous robot wants to communicate with the planner node, it first notifies the ROS Master. Also, the planner node notifies the ROS Master to subscribe autonomous robot. Then autonomous robot publishes sensor's data under a topic. This communication continues between autonomous robot and planner node without any notification to ROS Master again.

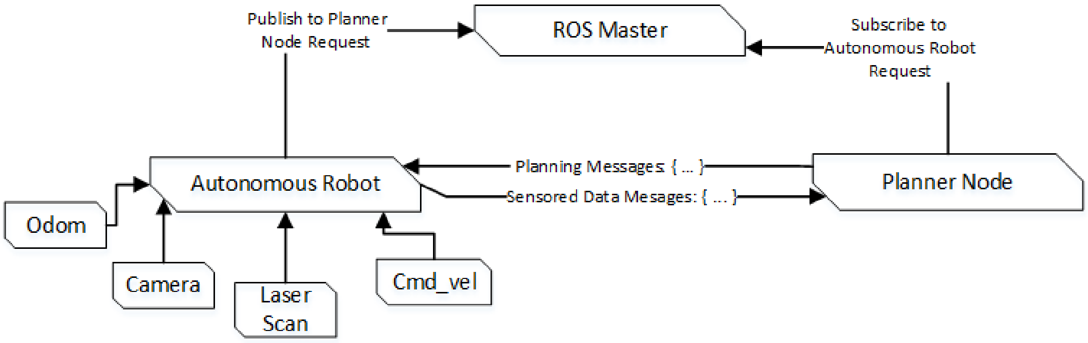
Publish to Planner Node Request
ROS Master
Subscribe to Autonomous Robot Request
Autonomous Robot
Planning Messages { ... }
Sensored Data Mesages: { ... }
Planner Node
Odom
Camera
Laser Scan
Cmd_vel

Figure 2. *ROS architecture for an autonomous robot application*

ROS communication ease of use reveals some security problems. In a network where ROS is running, if ROS Master information is known, any device could be freely publishing messages over ROS or subscribe to any node in the system. If the ROS Master information is not known, Master information could be detected easily with tools such as "NMAP". DoS attack makes the system inoperable by sending meaningful or meaningless packets to the target server or network. In the ROS, the bandwidth can be filled by publishing large packets at very high frequencies either secretly or explicitly. However, in a system where ROS running, DoS attacks can be achieved by occupying various different points such as ROS master, ROS nodes, bandwidth of the system. One of the important points of ROS is ROS master. If the attack to ROS Master successful, it may prevent real nodes that want to join the system or unable to meet subscriber/publisher requests. The DoS attack, which is inspired by the ROSPloit tool used in this study is carried out in this way. ROS nodes are another important part of the system. After the ROS master is captured, multiple ROS nodes can be created with a single attacker device. Another DoS attack that can be made to ROS is by completely filling the nodes' response ability to subscriber data requests. This is another DoS attack method used in the proposed system. The DoS attack targets to ROS nodes and transport layer of the ROS. For this DoS attack multiple ROS nodes is created with a single attacker device. And these attacker nodes subscribe the autonomous vehicle's sensor data (Figure 3). While the autonomous vehicle tries to respond to this large amount of data requests that comes from the attacker node. The autonomous vehicle node becomes unable to provide data to the planner node. As an example of Odom sensor data request from multiple attacker nodes of RQt graph is given in Figure 3.

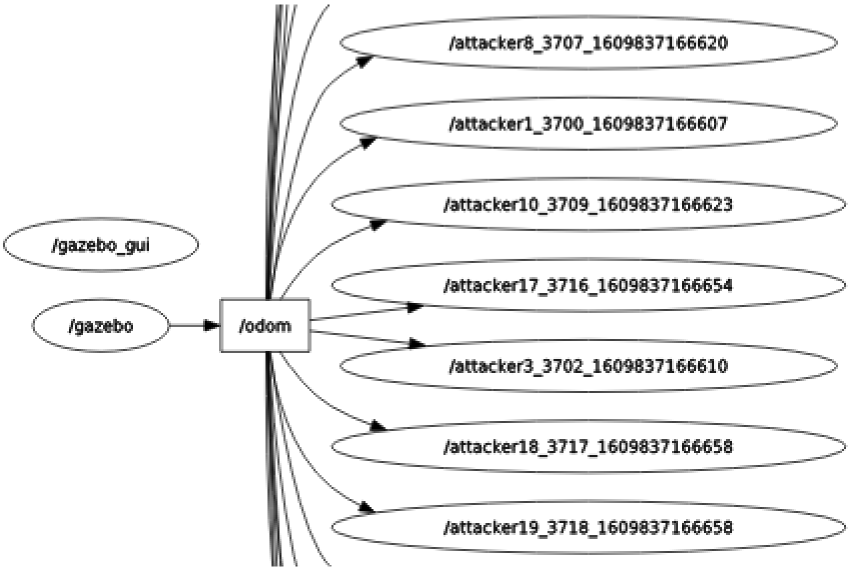
/attacker8_3707_1609837166620
/attacker1_3700_1609837166607
/attacker10_3709_1609837166623
/gazebo_gui
/attacker17_3716_1609837166654
/gazebo
/odom
/attacker3_3702_1609837166610
/attacker18_3717_1609837166658
/attacker19_3718_1609837166658

Figure 3. *An example of DoS attack to ROS nodes*

4. EXPERIMENTAL STUDIES

In this section, we first describe our experimental environment, then present the scenarios applied in this environment and present the results obtained.

4.1. Experimental Environment

The experiments that performed on ROS is focused on packet loss and delays for various DoS attack scenarios for the experimental setup in Figure 4. The resulting effects are monitored.

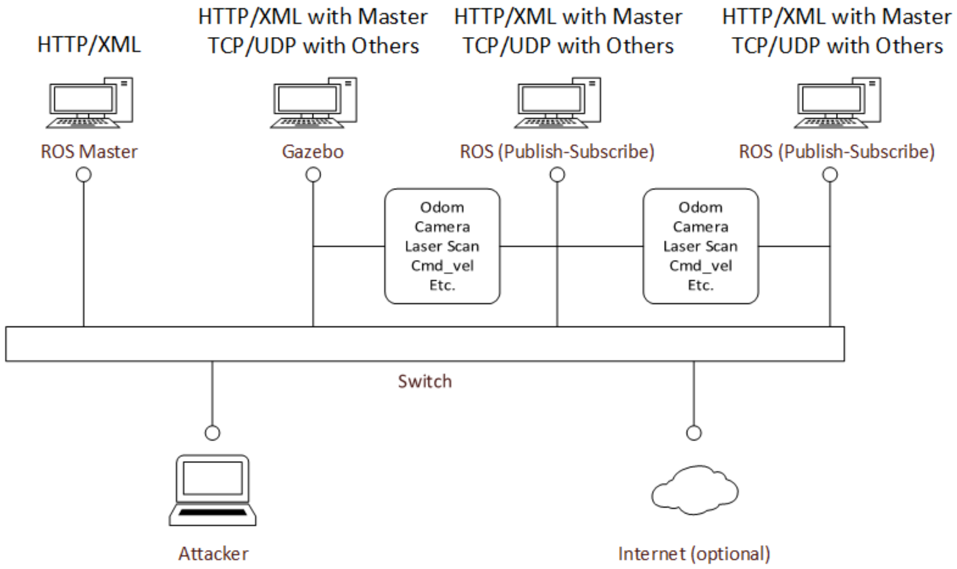

Figure 4. *Proposed system diagram for DoS attack on ROS*

All devices used in this experimental setup such as ROS Master PC, Gazebo PC, Publisher/Subscriber PCs have the same hardware and software. The experiments are performed using the PCs with the specifications listed in Table 2.

Table 2. *Specifications of PCs and Network Switch*

CPU: Intel Core i7 6700 @ 3.40GHz Skylake 14nm RAM: 8.00 GB Dual-Channel DDR3 @ 797MHz Motherboard: Dell Inc. 0FTVXT Graphics: 2GB NVIDIA GeForce GTX 1050 Storage: 250 GB KINGSTON SA400S37240G SATA SSD Ethernet: Onboard OS: Ubuntu 20.04.1 LTS 64-bit 3.36.8 ROS Version: Noetic Ninjemys Gazebo Version: 11.0.0
Network Switch: TP-Link SF-1008D 8-Port 10/100Mbps

4.2. Experiment Design

In this study, we examine the effects of security attacks under various network traffic scenarios. The experiment design is based on variation of both background traffic volume and attack volume. While creating the test scenarios, it is basically based on changing one of these two variables while remaining the other one constant. In each of the first four scenarios, a fixed attack scenario has been implemented under different background traffic volume. The fifth scenario consists of performing three different attack volumes consecutively under a single background traffic type. It is possible to increase the number of scenarios much more by increasing the number of variables or by testing with different values/volumes of the existing variables. In this work, it is aimed to understand the general characteristic with basic comparisons.

The four background traffic data volumes created for the scenarios are; 1) No Data Traffic/Only Master, 2) Minimum Level Data Traffic, 3) Medium Level Data Traffic (Basic Movement), 4) High Level Data

Traffic. ROS Master and Gazebo nodes are always running for each scenario. The details of these scenarios are as follow:

1. **No Data Traffic /Only Master** means that only ROS Master and Gazebo nodes are running. Data traffic on the network does not include any traffic other than ROS Master and Gazebo startup. Except these there is no traffic with constant communication.
2. **Minimum Level Data Traffic** includes ROS Master, Gazebo nodes and one subscriber node for each ROS (Publish-Subscribe) devices. These subscriber nodes are listening one topic which is odometry data for this scenario.
3. **Medium Level Data Traffic** means one node running for each ROS (Publish-Subscribe) devices. These nodes are subscriber of odometry data, laser sensor scan data and publish data to control robots on Gazebo simulator on a constant path without crashing.
4. **High Level Data Traffic** covers all tasks in Medium Level Data Traffic, in addition, nodes in ROS (Publish-Subscribe) devices listen to every possible sensor and data that the robot (Turtlebot 3Waffle) has, such as; camera data, depth data, inertial measurement unit data.

On the other hand, there are two types of DoS attacks in our experiments: 1) DoS attack on master ports 2) Subscriber attack (Subscriber node (n) attack on publisher node), 3) Publisher Attack (publisher node attack on subscriber node)

1. **DoS attack on master ports:** ROS architecture consists of a ROS Master and other nodes. ROS Master manages the entire system. If the ROS Master stops working or its communication with other nodes on the network is interrupted, the nodes that cannot communicate with the master will also stop working. In this type of attack, large amounts of data are transferred to the ports in the range specified in the software by targeting the computer acting as the ROS Master. This attack occurs over the Transport layer and is a DoS attack in the attack tool called ROSPloit.
2. **Subscriber Attack (Subscriber node (n) attack on publisher node):** ROS is an IoT middleware software that enables the communication of endpoints called nodes. A node can send data to multiple nodes at the same time or receive data from multiple nodes simultaneously. Data flow is carried out through a structure called topic. Topic only flows data in one direction between two nodes, and there cannot be more than one topic with the same name between two nodes. In this attack, a variable number (n) of nodes requesting to listen from a node with data publishing feature is created to force the network and the system to fail to respond as it should (Figure 5). For each node created, a high amount of traffic occurs in the application layer between the ROS master and attacker, and in the transportation layer between the publisher node (Gazebo) and the attacker. The number of nodes is chosen as n = 1000 in the first four scenarios, and n value is chosen as 10, 100 and 1000 in the fifth scenario.

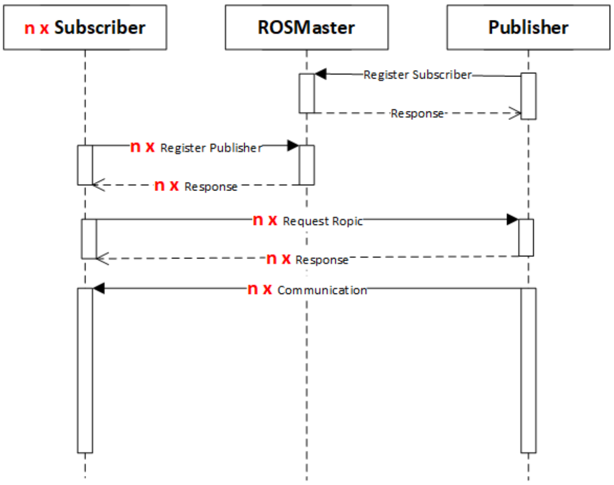

Figure 5*. Subscriber DoS attack; Publisher-ROSMaster-Subscriber communication flow*

3. **Publisher Attack (publisher node (n) attack on subscriber node):** In this attack, very similar to the Subscriber attack a large number of ROS nodes are created (Figure 6). Nodes formed within the scope of the scenario act as Publisher (Figure 6). In ROS environment, a listener node should be chosen as the target. The effect of this attack varies according to what kind of operation the listener node performs on the incoming data. For instance, if the subscriber node is logging data, the storage space will be full, if it is processing data, it will consume processing power, etc. may have short, long-term or permanent effects on the system. Since the attacking device can both run these created nodes on itself and send data over these nodes, it needs much more processing power per node compared to the Subscriber attack. Therefore, the upper limit of the number of nodes that are kept active during this attack is less compared to the Subscriber attack.

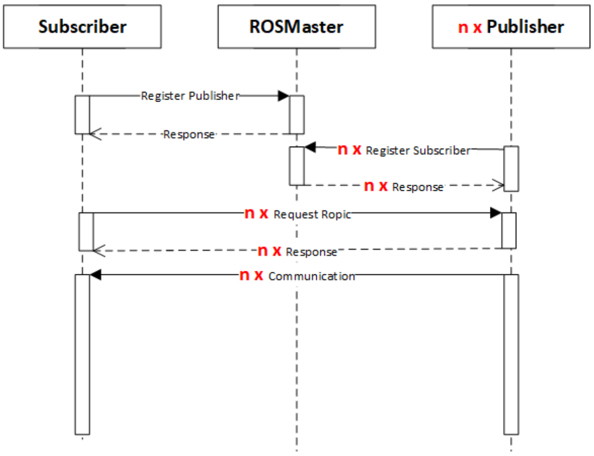

Figure 6*. Publisher DoS attack; Publisher-ROSMaster-Subscriber communication flow*

The list of events in the experiment performed in the first four scenarios is shown in rows of the Table 3 and the fifth scenario is shown in rows of the Table 4. The difference between the scenarios detail in the first four experiments is the Data Traffic volume as explained previously in this section. On the other hand, the fifth scenario includes a “DoS attack on master ports” and following with four different volume of attack which are performed under the medium level data traffic volume on ROS. Meanwhile columns of the Tables 3 and 4 divided by two conditions; one is given details about Subscriber attack conditions, other is given details about Publisher attack conditions.

Table 3. *The input conditions for the first four experiments*

THE CHARACTERISTIC OF ROS DATA TRAFFIC	CONDITIONS 1	CONDITIONS 2
ROS not running	Network data started to be saved	
No Data Traffic/ Only Master	ROS initialized (active)	
Background Traffic Data Volumes: 1) No Data Traffic/Only Master 2) Minimum Level Data Traffic 3) Medium Level Data Traffic 4) High Level Data Traffic.	Run Simulation (initialize)	
	DoS attack on master ports	
	End of the attack	
	Subscriber node (n) attack on publisher node (1000 nodes)	Publisher node (n) attack on subscriber node (1000 nodes)
	End of the attack	
ROS not running	ROS shut down	
No Data	End of the network traffic logging	

Table 4. *The input condition for the fifth experiment*

THE CHARACTERISTIC OF ROS DATA TRAFFIC	CONDITIONS 3	CONDITIONS 4
ROS not running	Network data started to be saved	
No Data Traffic/ Only Master	ROS initialized (active)	
Data Traffic Volume (Fixed to Medium Level Data Traffic)	Run Simulation (initialize)	
	DoS attack on master ports	
	End of the attack	
	Subscriber node (n) attack on publisher node (10 nodes)	Publisher node (n) attack on subscriber node (10 nodes)
	End of the attack	End of the attack
	Subscriber node (n) attack on publisher node (100 nodes)	Publisher node (n) attack on subscriber node (100 nodes)
	End of the attack	End of the attack
	Subscriber node (n) attack on publisher node (1000 nodes)	Publisher node (n) attack on subscriber node (1000 nodes)
	End of the attack	End of the attack
	Subscriber node (n) attack on publisher node (10 nodes)	Publisher node (n) attack on subscriber node (10 nodes)
	End of the attack	End of the attack
ROS not running	ROS shut down	
No Data	End of the network traffic logging	

4.3. Test Results

In this work, two of Quality of Service (QoS) parameters, which are delay and packet loss, are analyzed to reflect the behavior of ROS under DoS attack. Delay is defined as the time interval between the request and the response time. The delay value in this experiment is based on the communication between publisher and subscriber nodes. Packet loss is defined as the number of packets or bytes lost during publishing and subscribing within a test scenario.

4.3.1. Test results: subscriber attack for the first four scenario

Figure 7 shows the delay on the application layer for the first four scenarios of condition 1 (details given in Table 3). Figure shows the "Subscriber node (n) attack on publisher node" attacks impacts on application layer. In Figure 7, the y-axis indicates the delay amount of the graph, and the x-axis indicates the time in seconds. It can be seen that the delay increases significantly during periods of attacks in all scenarios. Through this graphic, it can be said that the DoS attacks in such a system can be easily detected with delay measurement only. Another remarkable case here is that the highest delay value is the same for each scenario despite four different background traffic. It has been evaluated that the reason for this may be that the created background traffic does not make much difference compared to the network bandwidth.

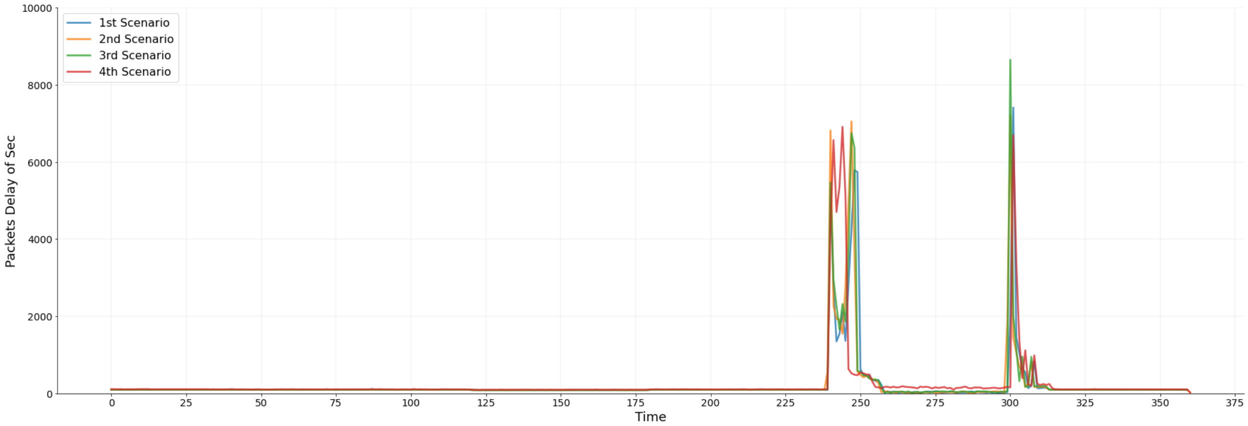

Figure 7. *Subscriber attack: Comparison of first four scenarios in terms of Delay on Application Layer*

Figure 8 shows the packet loss on application layer for the first four scenarios of condition 1 (details given in Table 3). Figure 8 shows "Subscriber node (n) attack on publisher node" attacks impact on application layer through the loss packets. In this graph, it is seen that packet losses start with the first attack and take g highest place in time frames which are the times higher delay occur. The reason for this is considered to be the events specified in the scenario. While there was no remarkable packet loss starts even though there was a delay during the attack, it was evaluated that the delayed packets were lost when the attack was highest volume and terminated interval. In Figure 8, the y-axis shows the number of packets lost per second, and the x-axis shows the time in seconds.

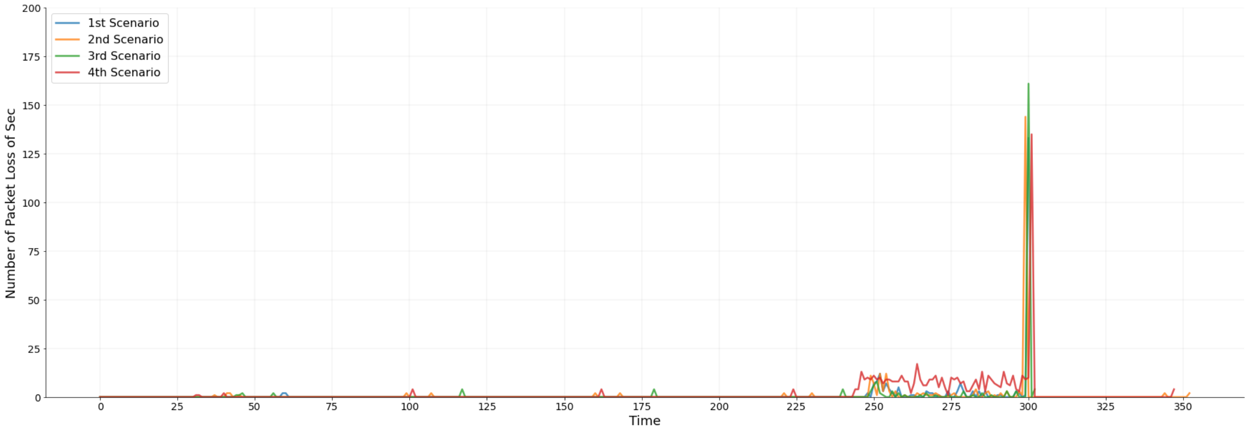

Figure 8. *Subscriber attack: Comparison of first four scenarios in terms of Packet Loss on Application Layer*

Figure 9 shows the subscriber attack impact of delay parameter for the first four scenarios on transport layer. As can be seen from the graph, the delays until the start of the attack are positioned in proportion to the traffic volume. When the attack started, it reached similar delay characteristics. It can be seen that attack volume delay occurs regardless of normal traffic volume.

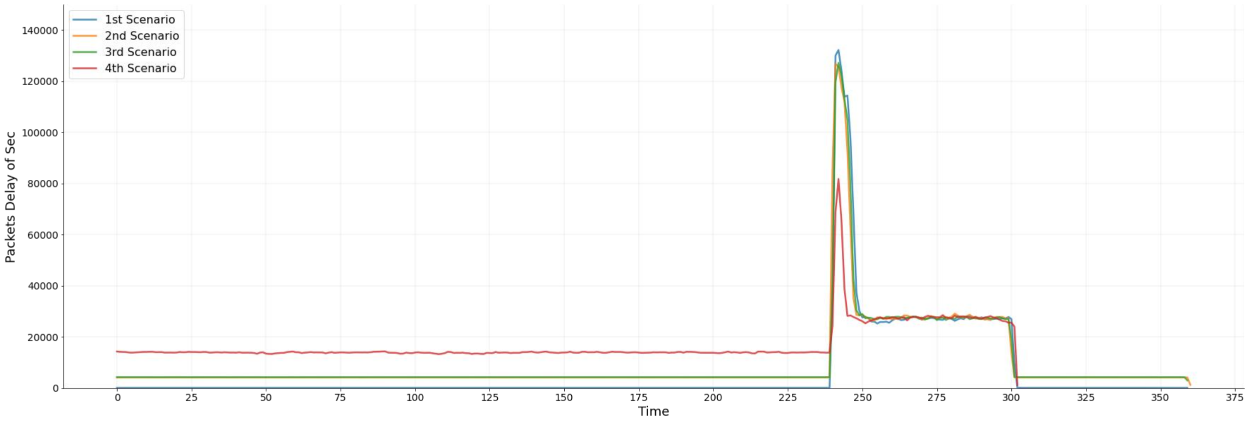

Figure 9. *Subscriber attack: Comparison of first four scenarios in terms of Delay on Transport Layer*

In Figure 10 shows subscriber attack, mentioned in Table 3 condition 1, effects the loss of packets on transport layer. It can be seen from the graph that until the attack starts there is no packet losses in the transport layer. And after the attack starts there is no immediate impact on packet losses. This is because the system becomes unable to receive packages after a while attack starts. Among the parameters that can be used in methods such as early-time attack detection, it has been observed that the packet losses effect can be seen that after than the delay parameter.

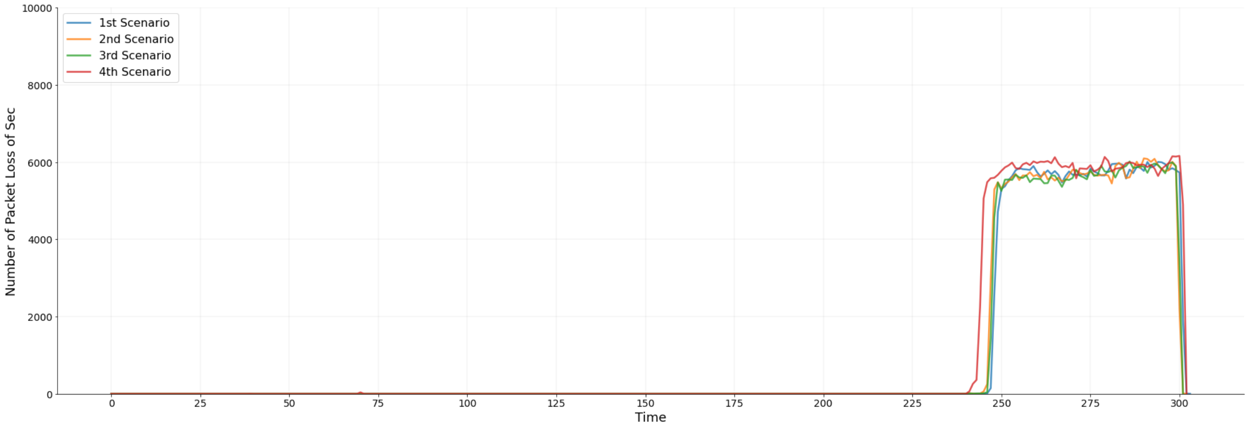

Figure 10. *Subscriber attack: Comparison of first four scenarios in terms of Packet Loss on Transport Layer*

4.3.2. Test results: publisher attack for the first four scenario

Figure 11 shows the delay on the application layer of second type of attacks publisher attack for the first four scenarios. It can be seen that the delay starts to increase after 225 sec. and reaches highest delay on around 300 sec. Through the graphics it can be seen that the publisher attack and the subscriber attack show similar results in the case of the highest attack, it has been seen that the publisher attack causes less delay at the beginning of the attack (around 225 seconds). When the publisher attack reached its highest values in y-axis, it was around 14000, while the subscriber attack reached a delay of around 130000. As observed in the subscriber attack, similar delays occurred in all scenarios during the attack, regardless of the network traffic volume in the publisher attack.

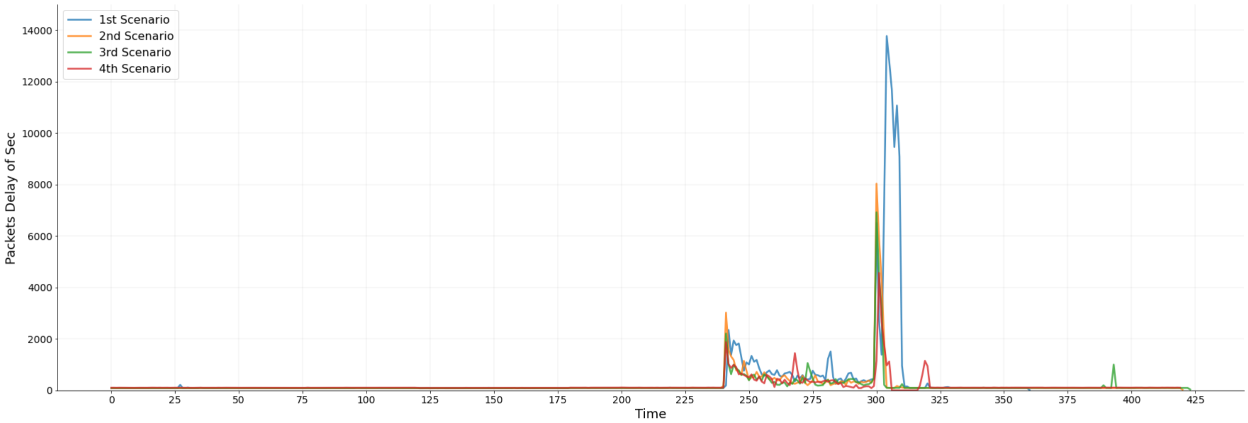

Figure 11. *Publisher attack: Comparison of first four scenarios in terms of Delay on Application Layer*

In Figure 12 shows packet loss for the publisher attack on application layer for the first four scenarios. In Figure 12 shows that there is an increase in packet losses after the publisher attack started around the 250th second and the delay started to increase. It is observed that publisher attack can cause highest loss of packets for high network traffic volume scenario (scenario 4). It is observed from the tests that the publisher attack has less packet losses than the subscriber attack on application layer. The reason for this is considered to be the subscriber attack characteristics create more network traffic than publisher attack.

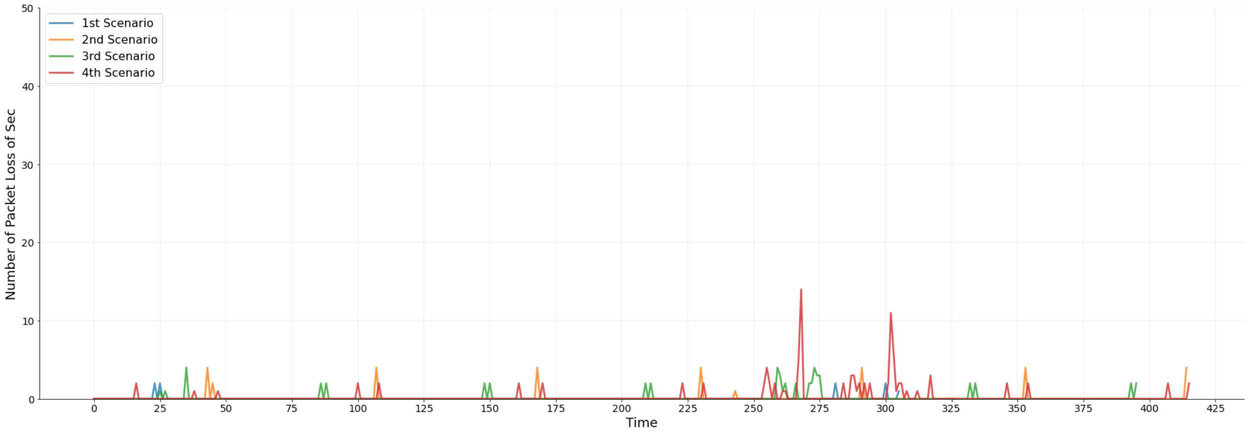

Figure 12. *Publisher attack: Comparison of first four scenarios in terms of Loss Packet on Application Layer*

Publisher attack on the Gazebo environment creates more resource usage because it is both to raise the nodes and to send the package contents as mentioned in Section 4.2. In Figure 13, all scenarios showed similar delays until the attack started. However, it has been observed that there are delays inversely proportional to the normal network traffic generated after the attack started around 240 sec. It is evaluated that after 300 sec. the delays continue until the end of the scenario even if the attack was over. It is seen that the publisher attack causes delays in the Gazebo environment in the short, medium or long term after a while even the attack ends.

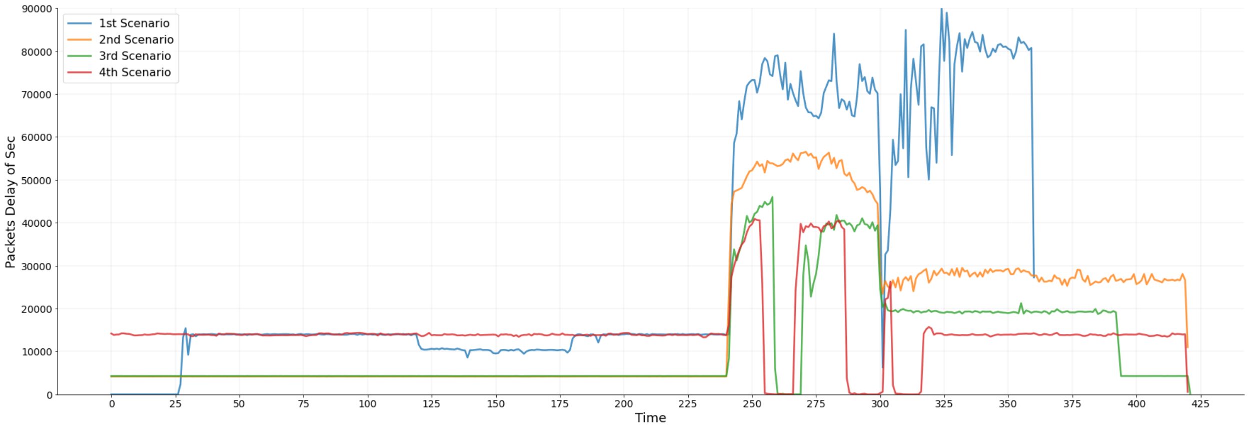

Figure 13. *Publisher attack: Comparison of first four scenarios in terms of Delay on Transport Layer*

In Figure 14, the effect of publisher attack on packet loss is evaluated from the transport layer. The evaluations have shown that packet losses are seen more clearly than the application layer. After the attack started, packet losses started around 250 secs and reached the highest level at 300 secs. In scenario 1 (Figure 13) that there is no data traffic, it has been observed that since the delays are completely caused by the traffic generated by the attack, it does not cause any significant packet losses. It has been observed that when a publisher attack is added to a scenario with self-traffic in the system, it may cause loss of packets in self-traffic communication packets.

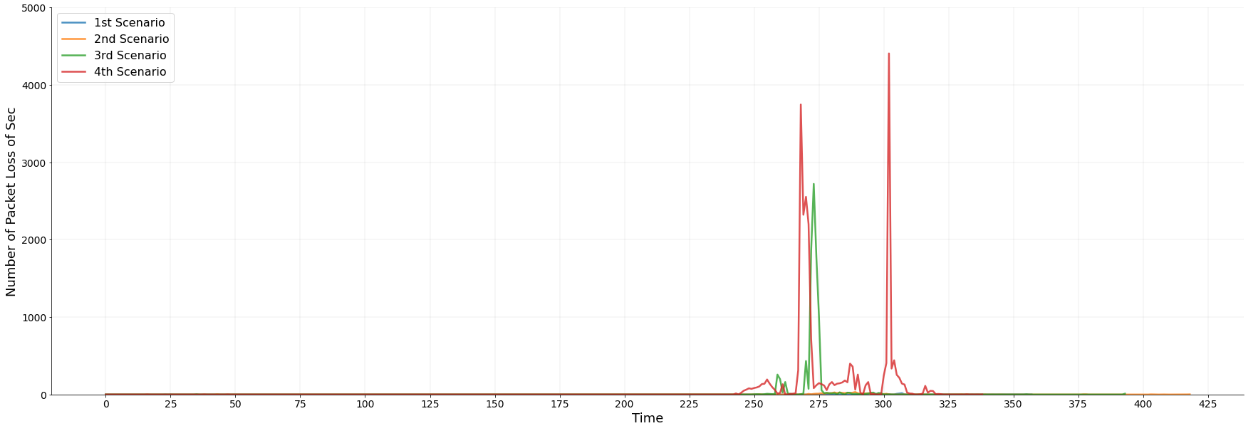

Figure 14. *Publisher attack: Comparison of first four scenarios in terms of Loss Packet on Transport Layer*

4.3.3. Test results: subscriber/publisher attack for the fifth scenario

In this subsection we evaluated fifth scenario for Subscriber attack and Publisher attack for the packet loss and delay parameters. Figures 16 and 17 show the packet losses on the side of ROSMaster, which is considered the application layer of the ROS middleware. Since the fifth scenario longer than other scenarios contains more packages than other four scenarios. As shown in Figures 15 and 16, packet losses increase during the attack however in subscriber attack graph has more packet losses than publisher attack. It may be cause for subscriber attack has more packages in the network than publisher attack, however publisher attack has heavy packages than publisher attack. It has been determined that in the traffic on the ROS Master side (application layer), packet losses have tremendously increased only in the time interval when the attack reaches 1000 nodes.

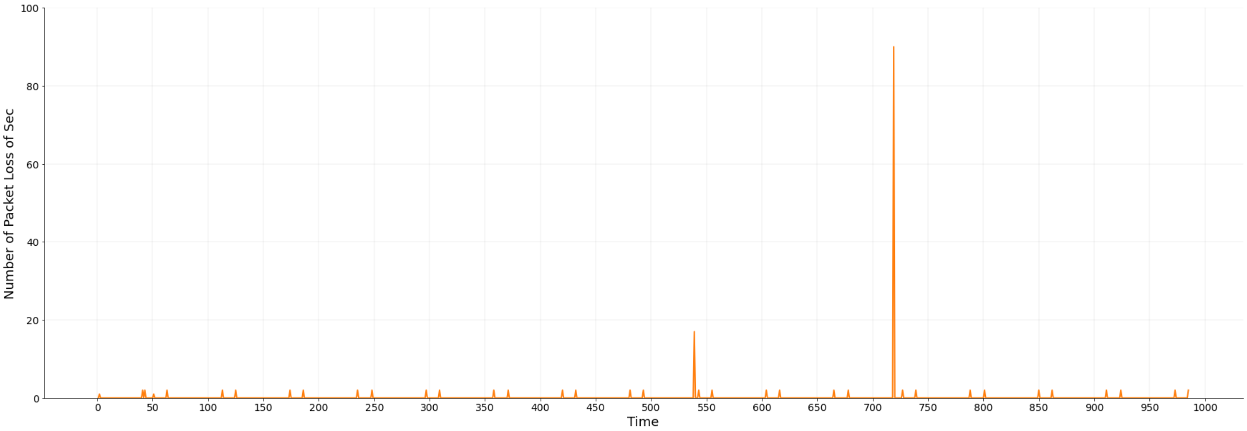

Figure 15. *Subscriber attack Packet Loss scenario five on Application Layer*

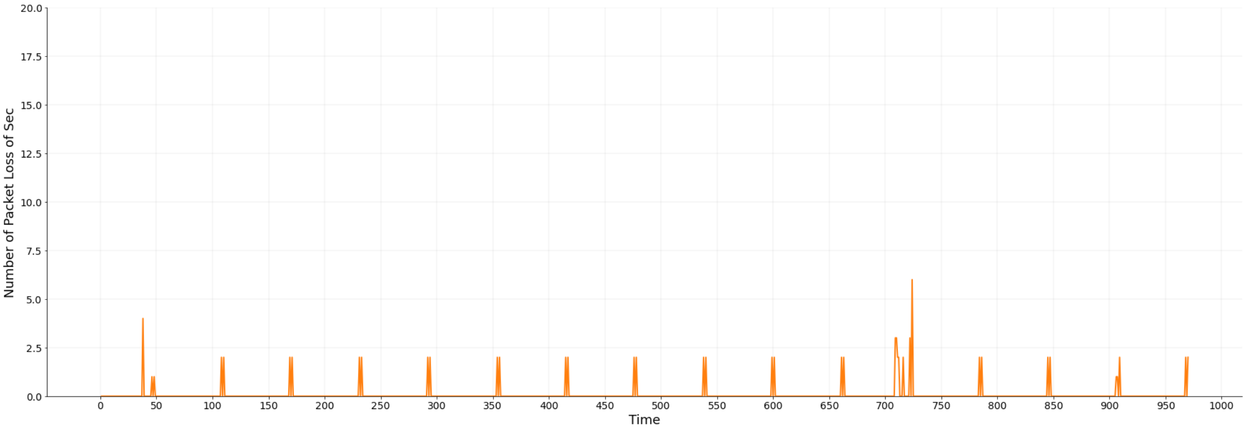

Figure 16. *Publisher attack Packet Loss for scenario five on Application Layer*

Figures 17 and 18 show the Delay under different attack volumes on application layer for the fifth scenario. Delay figure reveals that a similar flow as Figures 15 and 16. In fifth scenario the attack volume differentiates with low, medium and high which mentioned in the experimental scenarios. In Figures 17 and 18 show that low attack effects can be seen in around 300 sec., medium level attack effect can be seen around after 450 sec. and high volume attack effect can be seen around 650 sec. and the last low volume of attack delay around 850 sec. in Figures 17 and 18. As a result, as can be seen in Figures 17 and 18, the effect of the attacks can be observed by monitoring the delay parameter in the application layer. Through the delay graphs, it can be observed that subscriber attack may higher delay impact on application layer than publisher attack. It can be observed from the figures, the delay increases by proportion to the attack volumes for the both attack type.

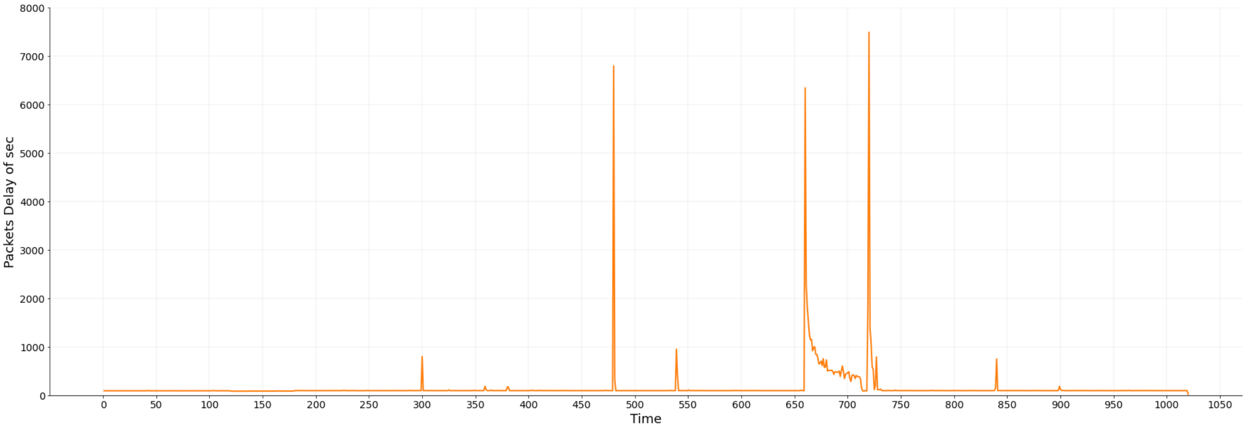

Figure 17. *Subscriber Attack experimental scenario five: analysis of Delay under different attack volumes on Application Layer*

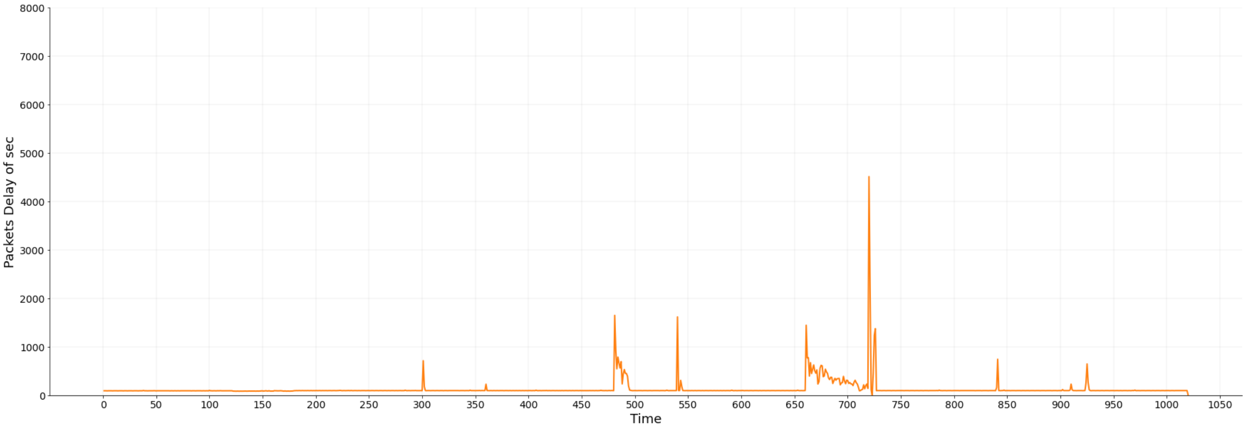

Figure 18. *Publisher Attack experimental scenario five: analysis of Delay under different attack volumes on Application Layer*

Below two graph (Figures 19 and 20) are examined the delay of Subscriber attack and Publisher attack for the fifth scenario. The same static upper limits are not specified for the graphs y-axis, as the attack types are different in all test results' graphs, and the upper limits on the y-axis could give information about network volume or attack-related characteristics. In Figures 19 and 20, the y-axis show the packet delay of second, and the x-axis show the time in seconds. In the fifth scenario, delay is examined for both ROS Master and Gazebo node for observing the effects of repeated DoS attacks. In the fifth scenario three different volume of DoS attacks are evaluated as mentioned in previous section (see in Table 4). Due to this evaluation, fifth scenario takes longer time than other scenarios and the monitored data reached huge size. Due to this reason in Figures 19 and 20 delay graphs scales are going huge amount of delay values. Besides Figure 19, Master side delay can be seen detailed in Figure 17. In the ROS master side delay effect of attacks can be seen tide interval, while delay on the Gazebo side are observed for all attacks takes long time interval. In Figure 19, the Gazebo side firstly increase around 300 sec corresponds to low volume attack, after that another increase shows around 480 sec corresponds to medium level attack, and the third increase observes during 650 sec correspond to high level attack. The last increase around 800 sec in Figure 19 corresponds to low level attack effect. Also in Figure 19, on Master size the delay increases by proportion to the attack volumes on ROS middleware.

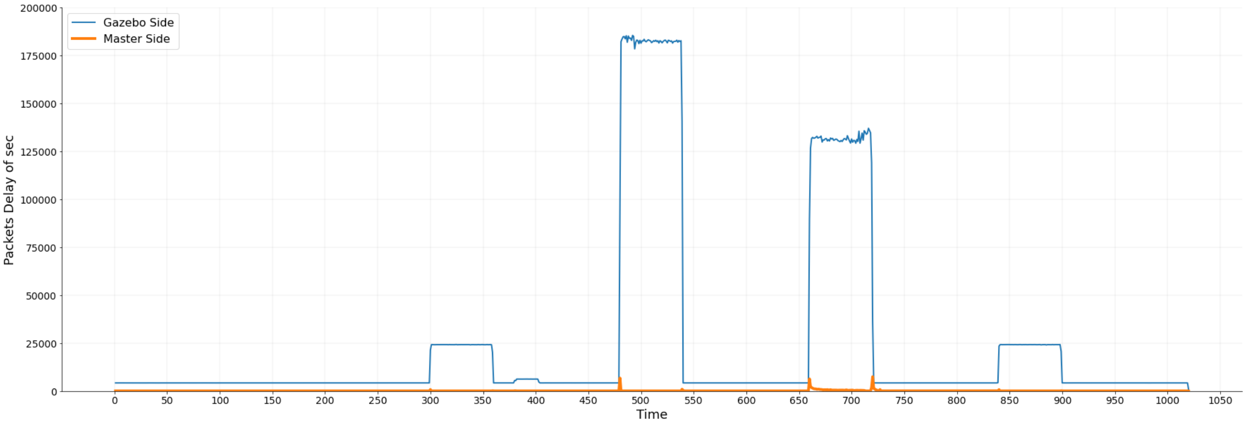

Figure 19. *Subscriber attack Delay of ROS Master and Gazebo for scenario five*

In the Figure 20, fifth scenario is evaluated for the examine the effect of repeated attacks of publisher attack. In Figure 20, each delay rises corresponds respectively low, medium, high and again low volume attacks are seen 300. sec, 500. sec, 650. sec and 850. sec. Besides Figure 20, Master side delay can be seen detailed in Figure 18. In Figure 20, the publisher attack caused delays on the Master side proportional to the size of the attack like similarly to subscriber attack. On the Gazebo side, the delays have remained nearly at the same level under the attack situations. The reason for this may be the resource usage did not return to its normal state after the first attack which also related the publisher attack mechanism described in section

4.2. In the last low-volume attack, it reached the highest delay around 850th sec due to the reason for the accumulation of packet delays. The Figure 20 shows that the effect of publisher attacks on the Gazebo side of the publisher attack affects the next network traffic, however the attack not gives same characteristics in master side.

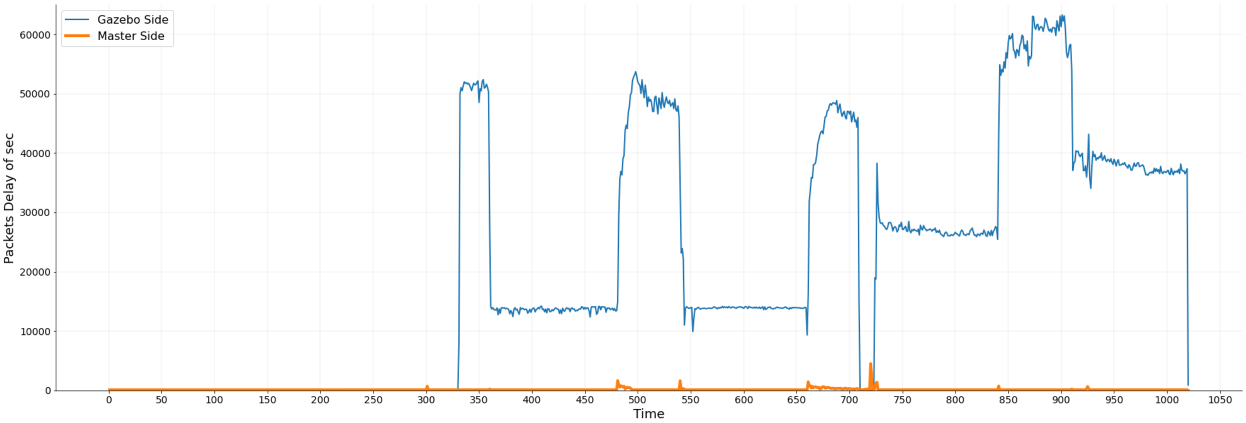

Figure 20. *Publisher attack Delay of ROS Master and Gazebo for scenario five*

In this study, the first four scenarios are evaluated to determine is the normal traffic volume of the network affect the attacks impact or not. First four scenarios are evaluated with two types of attack: Subscriber attack and Publisher attack. In the subscriber attack results are showed that the self-traffic volume is not barely affect the observation of the attack in both application layer and transport layer. However, packet losses analyses diversify by application layer and transport layer. The packet losses on application layer impact is shortly seen in the graph while the packet losses continue on the transport layer for a while. Publisher attack analyses is evaluated in both QoS parameters from application and transport layers. Publisher attack delay parameter analyses on application layer result characteristic is similar to subscriber attack impact on application layer. The self-traffic volume is not significant impact on Delay parameters onto publisher attack too. However, publisher attack on transport layer analyses gives the attack effect continues after the attack finishes for a while. And also in no traffic scenario (Scenario 1) gives highest delay values during the attack. It gives an information about while there is no traffic on the network, an attack is dominating the all communication in the network. And no traffic delay characteristic can give information about attacker characteristic since there is no another traffic then attacker. In transport layer publisher attack, by looking loss packet analyses give significant result for publisher attack because there are no other results that affect the other layers and attack type more.

In the last test scenario, the volume of the network is kept constant and the volume of the attacks is changed. In this scenario, the effect of the come one after another attacks effects are examined, and the effect of these attacks is analyzed from both the application and transport layer. First low volume, then medium volume, high volume and lastly again low volume attack are evaluated. As a result of the evaluation, it is analyzed that the effect of the attack still affects the next attack volume, even if network traffic becomes normal. It has been observed that the effect of volumetric attacks can be greater in abandonment attack situations, even if the attack volume does not increase. In the last scenario it is observed that the subscriber attack creates much more communication volume than the publisher attack. However, it is observed that the effect of the publisher attack has longer network impact after the attack with compared to subscriber attack on the transport layer. These differences could be used for the determination of these attacks. It has been observed that the effect of delay parameters is seen faster than packet loss impact, so if the earliest detection is important, delay parameter could be used in early attack detection systems. It has been seen that the delay parameter, especially for Publisher attack, can be used in attack detection based on packet loss. Although the volume of the network is small, medium or large during the attack, it has been observed that the effect of the attack is clearly visible and shows similar effects regardless of network volume. In future studies, delay and packet loss parameters can be used for the determine attacks, regardless independently of the network volume.

5. CONCLUSION

This article evaluates the impact of DoS attacks on ROS network under various circumstances. Test environment is presented in detail and the scenarios are provided to explain our evaluation. In this study, we firstly evaluate different volumes of DoS attacks that targets application and transport layer on ROS middleware. Then network traffic log collects from both application and transport layer. Finally, the network traffic logs are analyzed to understand the impact of the attacks on a different dimension on ROS middleware. The results are compared using delay and packet loss, which are quality of service measures commonly used in the literature to understand the impact of DoS attack in a network.

This study can be considered as a pioneering study to compare the effects of an attack on the ROS middleware on both the application and transport layer. As a result of the study, it was seen that the effect of these attacks can be observed from both the transport and application layers. Through the test results, the Subscriber attack causes higher levels of delay as it creates much greater communication volume than the Publisher attack. However, it has been observed that the effect of the Publisher attack in the Transport layer causes delays in a longer time interval than the Subscriber attack. It has been observed that the effect of the attacks can be different in the different layers. As a result of the analysis made at the Application Layer, that the self-communication traffic volume did not make a noticeable difference to the delay characteristic of the attack during both attacks. This may be due to the size of the attack being much higher than the traffic volume. But for all different traffic volumes, the effect of the attack is clearly seen by delays and packet losses parameters. In the transport layer, the effect of the attacks differs. From the transport layer, where the attack characteristics can be seen more clearly. Considering the evaluation criteria, it is thought that the transport layer could be a considerable layer for the detection of attacks. It has been observed that the detection of these attacks can be done by looking at both layers in future studies. Different DoS attack types are studied in the literature, in this study, the unauthorized publisher/subscriber DoS attack is focused on to see the effect of the attack on ROS middleware. Other possible attacks for the ROS layer are also open to work. Among the other QoS parameters, in this study, we focused on packet loss and delay parameters are chosen, due to the observation of DoS attacks are well observed with these parameters and also these parameters are frequently used in the literature. The results also showed that by looking at these two parameters it can give presumption about to detect volumetric attacks on ROS.

In future studies, the effect of different scenarios on ROS could be examine by increasing attack types and improve environment components. In order to increase the variety of attacks, new volumetric attacks can be created. The analysis of new volumetric attacks could give insights for robust and secure communication in the ROS system. On the other hand, adding new components to the network may increase the diversity in legitimate network traffic. Also, the studies could be conducted to examine the tests with different QoS. In addition to increasing the types of attacks, developing prevention mechanisms is one of the important fields could study.

ACKNOWLEDGEMENT

This work was supported by theVALU3S project that has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876852. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Czech Republic, Germany, Ireland, Italy, Portugal, Spain, Sweden, Turkey (TUBITAK, under contract no:119N356).
The views expressed in this work are the sole responsibility of the authors and do not necessarily reflect the views or position of the European Commission. The authors, the VALU3S Consortium, and the ECSEL JU are not responsible for the use which might be made of the information contained in here.
This work is also supported by the Scientific and Technical Research Council of Turkey (TUBITAK),

Contract No 120N800, project title: “Verification and Validation of Automated Systems' Safety and Security ".

CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

REFERENCES

[1] Internet: IFR Executive Summary World Robotics 2020 Industrial Robots, Online.https://ifr.org/img/worldrobotics/Executive_Summary_WR_2020_Industrial_Robots_1.pdf, (2020).

[2] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., and Ng, A., "ROS: an open-source Robot Operating System", ICRA Workshop on Open Source Software, Kobe, 5, (2009).

[3] Rivera, S., Lagraa, S., and State, R., "ROSploit: Cybersecurity tool for ROS", 2019 Third IEEE International Conference on Robotic Computing (IRC), Naples, 415-416, (2019).

[4] Alemzadeh, H., Chen, D., Lewis, A., Kalbarczyk, Z., Raman, J., Leveson, N., and Iyer, R., "Systems-theoretic safety assessment of robotic telesurgical systems", International Conference on Computer Safety, Reliability, and Security, Springer, Cham, 213-227, (2014).

[5] Dieber, B., Breiling, B., Taurer, S., Kacianka, S., Rass, S., and Schartner, P., "Security for the Robot Operating System", Robotics and Autonomous Systems, 98: 192-203, (2017).

[6] White, R., Christensen, D., Henrik, I., and Quigley, D., "SROS: Securing ROS over the wire, in the graph, and through the kernel", ArXiv, abs1611.07060, (2016).

[7] Narayanan, V., and Bobba, R. B., "Learning Based Anomaly Detection for Industrial Arm Applications", Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and Privacy, Toronto, 13-23, (2018).

[8] Dieber, B., White, R., Taurer, S., Breiling, B., Caiazza, G., Christensen, H., and Cortesi, A., "Penetration testing ROS", Robot Operating System (ROS), Springer, Cham, (2020).

[9] Wu, B., Chen, J., Wu, J., and Cardei, M., "A survey of attacks and countermeasures in mobile ad hoc networks", Wireless Network Security, Springer, Boston, (2007).

[10] Mirkovic, J., and Reiher, P., "A taxonomy of DDoS attack and DDoS defense mechanisms", ACM SIGCOMM Computer Communication Review, 34(2): 39-53, (2004).

[11] Specht, S., and Lee, R., "Taxonomies of distributed denial of service networks, attacks, tools and countermeasures", CE-L 2003-03, Princeton University, Princeton, NJ, (2003).

[12] Mahjabin, T., Xiao, Y., Sun, G., and Jiang, W., "A survey of distributed denial-of-service attack, prevention, and mitigation techniques", International Journal of Distributed Sensor Networks, 13(12): 1-33, (2017).

[13] Salim, M. M., Rathore, S., and Park, J. H., "Distributed denial of service attacks and its defenses in IoT: a survey", The Journal of Supercomputing, 76(7): 5320-5363, (2019).

[14] Manavi, M. T., "Defense mechanisms against distributed denial of service attacks: a survey", Computers & Electrical Engineering, 72: 26-38, (2018).

[15] Durcekova, V., Schwartz, L., and Shahmehri, N., "Sophisticated denial of service attacks aimed at application layer", 2012 ELEKTRO, IEEE, Rajecke Teplice, 55-60, (2012).

[16] Saravanan, R., Shanmuganathan, S., and Palanichamy, Y., "Behavior-based detection of application layer distributed denial of service attacks during flash events", Turkish Journal of Electrical Engineering & Computer Sciences, 24(2): 510-523, (2016).

[17] Sreeram, I., and Vuppala, V. P. K., "HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm", Applied Computing and Informatics, 15(1): 59-66, (2019).

[18] Balsa-Comerón, J., Guerrero-Higueras, Á. M., Rodríguez-Lera, F. J., Fernández-Llamas, C., and Matellán-Olivera, V., "Cybersecurity in Autonomous Systems: Hardening ROS Using Encrypted Communications and Semantic Rules", Iberian Robotics Conference, Springer, Cham, 67-78, (2018).

[19] Dieber, B., Kacianka, S., Rass, S., and Schartner, P., "Application-level security for ROS-based applications", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Daejeon, 4477-4482, (2016).

[20] Fernández Muro, B., "Securing Communications in Surgery Robots", Ph.D Thesis, Navarra University, Navarra, (2018).

[21] Huang, J., Erdogan, C., Zhang, Y., Moore, B., Luo, Q., Sundaresan, A., and Rosu, G., "ROSRV: Runtime Verification for Robots", International Conference on Runtime Verification, Springer, Cham, 247-254, (2014).

[22] Staffa, M., Mazzeo, G., and Sgaglione, L., "Hardening ROS via hardware-assisted trusted execution environment", 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), IEEE, Nanjing, 491-494, (2018).

[23] Rivera, S., Lagraa, S., Nita-Rotaru, C., Becker, S., and State, R., "ROS-Defender: SDN-Based Security Policy Enforcement for Robotic Applications", IEEE Security and Privacy Workshops (SPW), San Fransisco, 114-119, (2019).

[24] http://wiki.ros.org/rosmon. Access date: 06.12.2020

[25] Yayan, U., and Yazici, A., "Reliability-Based Multi-Robot Route Planning ", International Journal of Robotics and Automation, 34(3): 266-272, (2019).