Middleware Security for a Robotic Operating System

R.C.B. Keppetipola
Department of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Colombo, Sri Lanka
IT20227968@my.sliit.lk

O. K. Siriwardena
Department of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Colombo, Sri Lanka
IT20058050@my.sliit.lk

O.D. Abeywickrama
Department of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Colombo, Sri Lanka
IT20153540@my.sliit.lk

S.R. Serasingha Yapa
Department of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Colombo, Sri Lanka
IT20026370@my.sliit.lk

S.M.B. Harshanath
Department of Computer Science & Software Engineering
Sri Lanka Institute of Information Technology
Colombo, Sri Lanka
harshanath.s@sliit.lk

Prof. Pradeep Abeygunawardhana
Department of Electrical & Electronic Engineering
Sri Lanka Institute of Information Technology
Colombo, Sri Lanka
Pradeep.a@sliit.lk

***Abstract*— The Robotic Operating System (ROS), which was first released in 2004, has grown to be a popular platform for creating and deploying robotic applications. However, as the complexity and diversity of robotic systems keep expanding, security of ROS has come to be a much-needed concern. Two goals fall under the heading of "improving the security of robotic systems" in my research portion. The first goal attempts to strengthen ROS' security by putting into place specific security measures to solve the software's present security problems and enhance its overall security. The second goal aims to increase the relevant robot's physical security. Unauthorized access is prevented, and the robot is shielded from harm physically. By strengthening the security and dependability of robotic systems and averting potential dangers and harm to both the robot and the environment, these goals will benefit the field of robotics and society at large.**

Keywords—security, Robotic Operating System, ROS, resolving issues.

I. Introduction

The widely used open-source ROS (Robot Operating System) framework is used to create and manage robots. Security is a major issue with ROS, as it is with any software system. As the layer separating the ROS application from the underlying communication infrastructure, middleware security is a crucial component of ROS security.

Middleware security for ROS entails putting security controls in place at the middleware layer to guard against threats and weaknesses that could jeopardize the security of the entire ROS system. This entails protecting communication pathways, confirming nodes' identities, and obstructing illegal access.[2]

It is possible to provide middleware security for ROS using a variety of methods, including encryption, authentication, access control, and intrusion detection. These precautions guarantee that only authenticated and authorized nodes are able to communicate with one another, that such communication is private and secure, and that it is not intercepted or eavesdropped upon.

For the safety and security of robot systems, middleware security for ROS is essential. Developers may create more secure and durable robots that are better able to manage the difficulties of the modern world by integrating security features at the middleware layer.

With further automation coming with the move to industry 4.0, robots will become increasingly important and used. Because of its distributed nature and the potentially vast number of sensors and people it links, Industrial Control Systems (ICS) are notoriously vulnerable to cyberattacks due to the complexity and fragility of their orchestration. Because of the new vulnerabilities that have arisen as a result of this technology and its increasing degree of integration and interconnectedness, security issues have been thrust into the spotlight. Under the time constraints imposed by industry1, security has had a secondary (or even tertiary) role in the development of the Robot Operating System (ROS). Confidentiality, Integrity, and Availability (CIA), the conventional triangle of information security, will likely be prioritized differently in ICS. When considering user data in IT systems, privacy is typically more crucial than accessibility.[2]

Because of this, it is frequently the best course of action to temporarily disable a service if it is under assault. When data is lost in an IT system, backups are used to restore the lost information. Nevertheless, "restoring from the last known good backup" won't fix the damage done by an ICS (such as an actuator destroying its surrounds). Shutting down a drone or a hydro dam, to provide two examples, is not an appropriate response if they come under assault. Prioritizing the availability of the system's core functionality until it reaches a safe condition (e.g., the drone has landed or the water reservoir behind the dam has been drained) is of paramount importance. For the purpose of creating robotic systems, the Robotic Operating System (ROS) is a set of libraries that facilitates communication between (abstracted) hardware and (pure) software components. Approaches can be used. Being an open-source venture, ROS has a bustling community of thousands of programmers and over 9,000 separate packages. As robotics becomes increasingly prevalent in various industries, the need for secure and reliable robotic systems becomes critical. Robotic Operating System (ROS) is a widely used open-source middleware framework that facilitates the integration of different robotic components into a single system. Cyber-attacks on middleware can compromise the entire system, leading to physical harm, data theft, and other serious consequences. Therefore, ensuring the security of middlewarein ROS is crucial for the safe and reliable operation of roboticsystems.[3]

Even though SROS and Secure-ROS have been developed by the ROS team in recent years, the current security solutions do not offer complete protection against all vulnerabilities. ROS makes it possible to create complex robotic systems down to the component level.

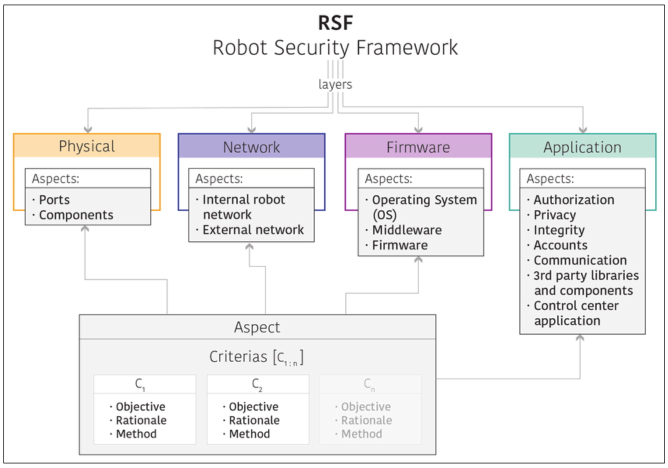

Figure 1 : Robot Security

Endpoint security refers to measures taken to prevent unauthorized access to or use of a device. In the Robot Operating System (ROS), "endpoints" include both the robots and the devices used to operate them. Robot Operating System (ROS) endpoint security is essential because of the prevalence of cyber-attacks against robots and their control devices. Information theft, system outages, and even bodily injury to people are all possible outcomes of a cyberattack on a robot [4].

Another approach to middleware security for ROS is intrusion detection, which involves monitoring the system for signs of potential security breaches. The system analyzes network traffic and system logs to identify potential threats and alerts system administrators to take appropriate actions. In the context of a robotic operating system, the term "middleware security" refers to the precautions that are taken to guarantee that the various components and nodes of the system may communicate and interact safely with one another.

II. LITERATURE REVIEW

As more and more robots are becoming internet- and network-connected, ROS's endpoint security features have become increasingly important. More and better network connections are needed for robots to function as their intelligence increases.

Unauthorized entry is a problem for ROS endpoint security. In ROS, malware poses a significant threat to endpoint security. Software designed with malicious intent is referred to as malware. ROS can be compromised by malware, which can then infect the operating system or controller of the robot. Malware has the potential to compromise the security of data, corrupt hardware, and even seize control of the robot, which could lead to unforeseeable actions. Whether in a factory or a hospital, robots are frequently deployed in settings where they must interact with humans. It's crucial in these settings that only approved people have access to the robots' command panels. Theft of confidential information, robot malfunction, or even bodily injury to people are all possible outcomes of unauthorized access. In order to stop anyone from messing with the robots and their controllers, it's important to set up access restrictions.[6]

Developers of ROS also have the responsibility of guaranteeing the safety of robots' interfaces with human-operated equipment. It's possible for thieves to intercept data or even seize control of the robot through its communication lines. Hence, encryption methods must be put in place to guarantee the safety of information passing between robots and control equipment. Additionally, software upgrades are a challenge that must be addressed by ROS's endpoint security. Software for robots and their controllers has to be updated. often to address security issues.

Authentication, encryption, access control, and auditing are the four basic pillars of a robotic operating system's middleware security. Although encryption guarantees that the data being transferred is safe and cannot be read by unauthorized parties, authentication entails validating the identities of the organizations that are participating in the conversation.

It comprises the use of security protocols and techniques to defend against potential vulnerabilities andattacks, such as the injection or eavesdropping of data in a robotic application.

Middleware security is essential for the safe and secure functioning of robotic systems, particularly within the context of Industry 4.0, where cyber assaults are becoming an increasingly serious risk to the integrity of the systems. It needs to be seen as supplementary to the implementation of safety, seeing as how a robot that is functionally safe but can also be commanded remotely is no longer safe[5].

Endpoint security for ROS is crucial since cyber-attacks against robots can cause a wide range of problems, from physical harm to the robot to stolen data and privacy violations. ROS endpoints are vulnerable to a number of different attacks, including command injection, buffer overrun, and denial of service. Thus, ROS needs solid and efficient endpoint protection [7].

III. METHODOLOGY

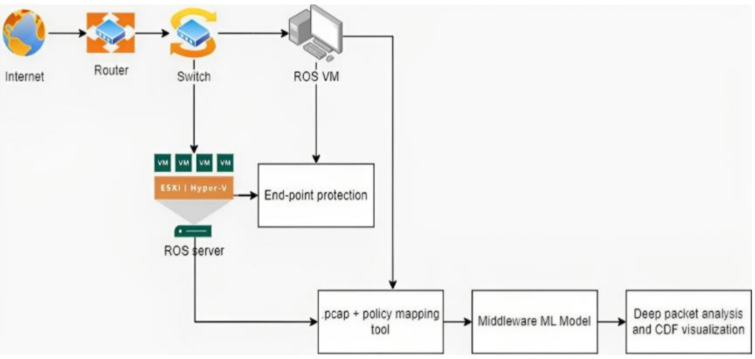

Figure 2 : Overall system diagram

A thorough literature analysis will be done in order to accomplish the first goal, which is to investigate and assess the authentication, encryption, and access control approaches currently utilized in ROS. Analyzing pertinent academic journals, conference proceedings, and other publications that cover ROS security will be part of this review's research process. The review's main objectives are to identify and assess the current authentication, encryption, and access control methods utilized in ROS as well as their shortcomings and vulnerabilities.

A risk assessment will be carried out to accomplish the second goal, which involves identifying potential safety risks and hazards in the robot's environment and functioning and evaluating the current safety measures in place. Identifying potential risks and hazards related to the environment and operation of the robot, as well as evaluating the effectiveness of the current safety precautions, are all part of the risk assessment process.

These interviews will offer insightful information about the current state of ROS security and safety, including any weaknesses in those safeguards. Suggestions will be made for enhancing the security and safety of ROS based on the findings of the literature review, risk assessment, and interviews. These suggestions will include solutions to any safety concerns and dangers found in the environment and use of the robot, as well as any restrictions and weaknesses found in the authentication, encryption, and access control mechanisms now in use.

The second generation of the Robot Operating System (ROS2) is a widely used middleware platform for creating and deploying robot software. The lack of network security safeguards in the prototype's design leaves it open to attacks. Protecting robot endpoints from targeted assaults necessitates implementing endpoint security for ROS2's middleware. This post will cover the topic of securing ROS2's middleware from endpoints. A device discovery system should be considered by enterprises so that an accurate inventory of all IoT devices on the network can be kept at all times.[7]

Each work-related device that connects to a network or the cloud should have endpoint security software installed. Protecting devices from a broad variety of threats requires endpoint security software with advanced security capabilities like malware protection, application whitelisting, access control, and so on.

Implementing security best practices is another critical part of ROS endpoint security for protecting the middleware. A minimum of fifty best practices across Azure, Amazon, GCP, and on-premises resources should be included. New resources, attack methods, vulnerabilities, and potentially harmful configuration errors should prompt regular updates to the list. In conclusion, securing middleware in ROS requires integrating security measures during the ALM process, identifying and authenticating middleware and application components, adopting IoT device security best practices, employing endpoint security software, and applying securitybest practices.

A. *Identify Security Challenges and assess Existing Mechanisms*

In this phase, we will research the literature to determine the security issues that ROS-based robotic systems must deal with and assess the existing security solutions that have been put forth for these systems. A review of academic papers, conference papers, and technical reports on middleware security for robots will serve as the foundation for the survey.

B. *Create an End-to-End Security Framework for Middleware*

Using the information from the literature research, we will create an End-to-End Security Framework for Middleware that can protect ROS-based robotic systems from beginning to end. All security-related topics, such as message encryption, access control, intrusion detection, and secure communication protocols, will be covered by the suggested framework.[8]

C. *Implementation and evaluation*

During this step, we will put the suggested security framework into use and assess its usefulness in practical situations. The suggested framework will be put to the test, and its performance in terms of security, dependability, and efficiency will be evaluated using a ROS-based robotic system.

D. *Comparison with Existing Mechanisms*

We will contrast the proposed security framework with current security practices to determine its benefits and drawbacks. Based on the evaluation results from the previous phase, a comparison will be made.

E. *Recommendations*

We will offer suggestions for enhancing the security of ROS-based robotic systems based on the results of this study. The assessments of the security issues and the outcomes of the suggested security framework's evaluation will serve as the foundation for the suggestions.

PROTETECTION AGAINST CYBER ATTACKS

In the moment, the primary objective is to conduct a comprehensive review of the existing literature to identify the security challenges that are encountered by robotic systems based on the Robot Operating System (ROS). Furthermore, the aim is to evaluate the effectiveness and suitability of the security mechanisms that have been proposed and implemented thus far for addressing these challenges in ROS- based robotic systems.

The survey will be based on a comprehensive examination of scholarly articles, conference papers, and technical reports pertaining to the topic of middleware security in the context of robotics Developing a Comprehensive [9].

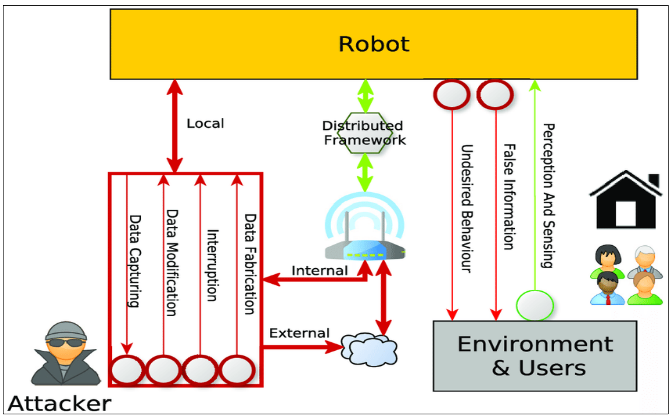

Figure 3 : Cyber Attack to Robot

Security Framework for Middleware: Based on the findings derived from the extensive review of relevant literature, an End-to-End Security Framework for Middleware will be developed with the aim of safeguarding ROS-based robotic systems throughout the entire operational process. The proposed framework will encompass various security-related subjects, including but not limited to message encryption, access control, intrusion detection, and secure communication protocols.

The implementation and evaluation phase involves the practical application of the proposed security framework and the subsequent assessment of its efficacy in real-world scenarios. The proposed framework will undergo testing to assess its performance in terms of security, dependability, and efficiency, utilizing a robotic system based on ROS. Comparison with Existing Mechanisms: In this study, we will undertake a comparative analysis between the proposed security framework and the prevailing security practices in order to ascertain the advantages and disadvantages of theformer.

A comparison will be conducted based on the evaluation results obtained in the previous phase. Recommendations: The present study will provide suggestions for augmenting the security measures of robotic systems based on the Robot Operating System (ROS). The evaluations of the security concerns and the results of the evaluation of the proposed security framework will form the basis for the recommendations [10].

ENDPOINT SECURITY FOR THE ROBOTICOPERATING SYSTEMS

The implementation of security measures in ROS2 systems commences by safeguarding the nodes comprising the network. In order to safeguard the nodes, it is important to establish measures that effectively protect them against unauthorized access, tampering, and various forms of malicious activities. In terms of ensuring the safety of ROS2 nodes, one may depend on the security capabilities offered bythe ROS2 security framework.

The aforementioned elements encompass authentication, encryption, and permission control. The Robot Operating System, now in its second iteration (ROS2), is a popular middleware platform for developing and distributing robot applications. Due to insufficient network security measures in the prototype's architecture, it is vulnerable to assault. [11]

Endpoint security for ROS2's middleware is essential for protecting robots from targeted attacks. Encrypted connections between ROS2 nodes are necessary for establishing endpoint security of the middleware [10]. In order to maintain an up-to-date inventory of all IoT devices on the network, businesses should think about implementing a device discovery system. Each work-related device that connects to a network or the cloud should have endpoint security software installed.

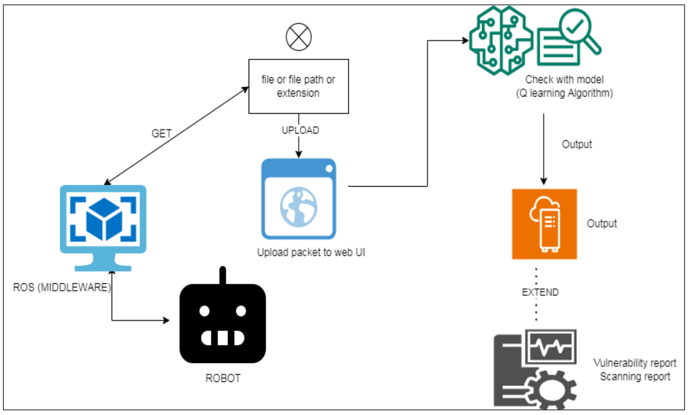

Figure 4 : Overall diagram for Endpoint Security

Protecting devices from a broad variety of threats requires endpoint security software with advanced security capabilities like malware protection, application whitelisting, access control, and so on. ROS endpoint security relies heavily on the use of security best practices to ensure the safety of the middleware. A complete and accurate inventory of all connected devices is possible with the help of security best practices for IoT. The establishment of security measures in ROS2 systems starts by safeguarding the individual nodes comprising the network. system with built-in security features such as password protection and encryption. This will help prevent attacks on the robots' last point [14]. The procedure kicks off with a slew of information gathered straight from the terminals. System logs, network traffic patterns, application behaviors, and user interactions are all examples of the types of data that fall under this umbrella.

For machine learning algorithms to function properly, this data is essential. The following stage is featuring extraction, which entails choosing pertinent properties or features from the gathered data. Predictions and the detection of abnormalities or dangers may be made by machine learning models using these characteristics. After that, we use the data to build machine learning models. These models are trained using labeled data that describes what constitutes harmful and safe behavior.

These models keep a constant eye on the actions and patterns of these nodes in real time. Data is analyzed for discrepancies with previously learnt patterns. An endpoint may be isolated, suspicious processes can be terminated, or security staff can be notified when a possible threat or anomaly is found, and the machine learning model will classify it and determine the best course of action [15].

SOLIDIFYING ROBOT OPERATING SYSTEM SECURITY

The ROS network, an advanced platform for robotic applications, is the brains of the system. In this case, a dynamic and intricate network is formed by different robotic devices and nodes exchanging commands and data. To protect this complex ROS network, we use packet capture methods. These methods follow network data packets as they go over the ROS communication channels, intercepting and capturing them. The gathered data offers insightful information about the traffic and activity on the network. The captured network data is meticulously organized and saved as PCAP (Packet Capture) files. These files serve as a comprehensive record of network activities within the ROS middleware. They encapsulate critical information about data exchange, potentially containing traces of network attacks. Our system incorporates an advanced network analysis tool into a web application with ease. This is the first line of defense; it is responsible for examining the PCAP files that have been captured and identifying any unusual network activity [16].

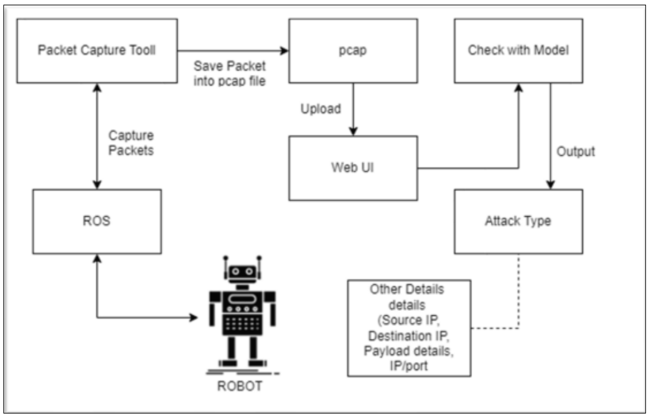
Packet Capture Tooll
Save Packet into pcap file
pcap
Check with Model
Upload
Capture Packets
Web UI
Output
ROS
Attack Type
Other Details details (Source IP, Destination IP, Payload details, IP/port
ROBOT

Figure 5 : Solidifying Robot Operating System Security

Within artificial intelligence (AI), machine learning is the study of creating models and algorithms that let computers learn from and make decisions based on data.

It is a crucial piece of technology that powers a number of applications, including recommendation systems, image, and speech recognition, and, in your case, network attack detection in the Robot Operating System (ROS). A "model" is a key point in machine learning. A model is basically an algorithm or mathematical representation that finds information, relationships, and patterns in a dataset. It is trained on past data to identify patterns and create predictions or classifications on new, unseen data, acting as the learning component of a machine learning system.

IMPROVING MIDDLEWARE VULNERABILITYMANAGEMENT.

The security of robotic systems and their applications can be greatly improved by increasing ROS's vulnerability management. Especially in areas like industrial automation, healthcare, and autonomous cars, ROS's weaknesses can represent serious threats due to its widespread use in robotics development [17]. Developers and organizations may reduce security risks, guarantee the security of robotic systems, and keep consumers and other stakeholders confident in their products by enhancing vulnerability management in ROS. As such, the ROS ecosystem as a whole must maintain a state of constant vigilance, cooperation, and a security-aware mindset.

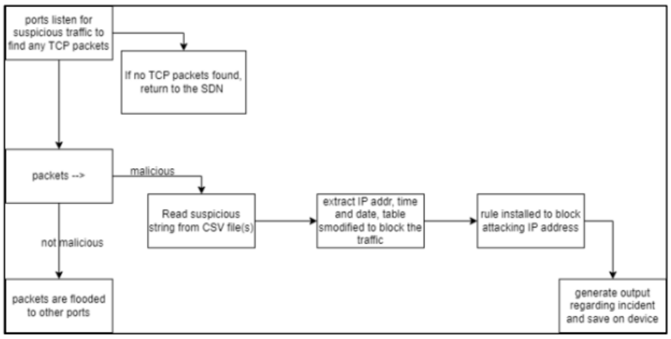
ports listen for suspicious traffic to find any TCP packets
If no TCP packets found, return to the SDN
packets -->
malicious
Read suspicious string from CSV file(s)
extract IP addr, time and date, table smodified to block the traffic
rule installed to block attacking IP address
not malicious
packets are flooded to other ports
generate output regarding incident and save on device

Figure 6 : Vulnerability Management

This picture is a schematic representation of a TCP packet. Transmission Control Protocol (TCP) packets are used to reliably and efficiently transmit data over a network. When using TCP, a connection is established between the sender and the receiver before any data is transmitted. By doing so, we can guarantee that our data will arrive intact and in the correct sequence [18].

VI. RESULT AND DESCUSSIONS

A. Results

Productivity is up since creating robotic apps now takes a fraction of the time they used to. This is because of the extensive library of open-source programs that have been developed by the extremely active and engaged community. Enhanced functionality: ROS has several tools that help boost the efficiency of robotic programs. Three examples are distributed processing, hardware abstraction, and real-time communication.

Improved adaptability: ROS provides a versatile foundation for creating new robotics software. This ranges from basic mobile robots to more sophisticated autonomous vehicles.

B. Discussions

The adoption of ROS has changed robotics. It has democratized robotic application development, allowing even novice researchers and developers to build complex robots. Thus, robotics innovation has increased. ROS has drawbacks despite its benefits. Complexity is a common complaint. New users may find it overwhelming to learn all the ROS ecosystem features and tools. ROS can also cause delay and jitter, making it unsuitable for real-time applications.

VI. CONCLUSION

In summary, within the domain of robotics, ensuring the security of middleware for a Robotic Operating System (ROS) holds significant importance. Middleware plays a crucial role in the design of the Robot Operating System

(ROS) by enabling effective communication and coordination across diverse robotic components and modules. Due to their intricate nature and interconnectedness, robotic systems are susceptible to many security vulnerabilities. ROS middleware security encompasses many measures aimed at safeguarding against unauthorized access, data breaches, manipulation, and denial of service attacks.

The privacy, security, and continuous accessibility of data and services offered by robotic systems can be ensured by the use of various security methods such as authentication, authorization, encryption, and intrusion detection. In conjunction with the consideration of software security, the inclusion of hardware security measures is vital in safeguarding the ROS middleware. The implementation of physical protections is important in order to prevent unauthorized manipulation of robot components.[21]

VII. Acknowledgment

We extend our heartfelt gratitude to Mr.Buddhika Harshanath, whose unwavering support and guidance have been instrumental in shaping the course of this research. His insightful feedback and expert perspective have greatly enriched our understanding and contributed to the depth of our findings.

Our gratitude also extends to Prof. Pradeep Abeygunawardane, whose advice and guidance has been instrumental in overcoming various challenges and obstacles. His unwavering encouragement and belief in our capabilities have been a source of motivation and empowerment.

Finally, we would like to thank all our colleagues, friends, and family members who have provided encouragement, understanding, and unwavering support throughout this journey. Your belief in us and your constant encouragement have been invaluable, and we are truly grateful for your presence in our lives.

References

[1] W. R. M. a. B. S. Vincenzo DiLuoffo, "Robot Operating System 2: The needwwewe for a holistic security approach to robotic architectures," International Journal of Advanced Robotic Systems., 2018.

[2] J. . McClean, C. J. Stull, C. R. Farrar and D. . Mascareñas, "A preliminary cyber-physical security assessment of the Robot Operating System (ROS)," Proceedings of SPIE, vol. 8741, no. , p. 874110, 2013.

[3] M. A. Rabbah, N. . Rabbah, H. . Belhadaoui and M. . Rifi, "Designing Middleware over Real Time Operating System for Mobile Robot," , 2017.[Online].Available:https://link.springer.com/chapter/10.1007/9 7 8-3-319-91337-7_37. [Accessed 5 5 2023].

[4] I. P. M. P. L. J. D.-J. Rafael R. Teixeira, "Security on ROS: analyzing and exploiting vulnerabilities of ROS-based systems," IEEE Xplore, 2023.

[5] R. S. Sean Rivera, "Securing Robots: An Integrated Approach for Security Challenges and Monitoring for the Robotic Operating System(ROS)," University of Luxembourg, Luxembourg, 2016.

[6] Q. Chen, C. Zhu, and X. Li, "Design of ROS-based middleware security architecture," Journal of Electronics & Information Technology, vol. 40, no. 1, pp. 74-81, 2018.

[7] L. D'Orazio, L. Pomante, and A. Oddi, "An Efficient and Secure Message Encryption for ROSbased Robot Networks," IEEE Transactions on Industrial Informatics, vol. 16, no. 8, pp. 5405-5414, 2020.

[8] Z. Xu, Y. Liu, B. Wei, and L. Zhu, "An Attribute-Based Access ControlMechanism for ROSBased IoT Systems," IEEE Access, vol. 8, pp. 175679-175690, 2020.

[9] Y. Zhao, S. Zhu, Z. Zhou, and J. Fan, "A Lightweight Intrusion Detection System for ROSBased Robot Networks," IEEE Access, vol. 8, pp. 205980-205992, 2020

[10] F. Pasqualetti, F. Dorfler, and F. Bullo, “Attack detection and identification in cyber-physical systems,” IEEE Trans. Automat. Contr., vol. 58, no. 11, pp. 2715–2729, 2013, doi: 10.1109/TAC.2013.2266831.

[11] “Stuxnet worm impact on industrial cyber-physical system security | IEEE Conference Publication | IEEE Xplore.” https://ieeexplore.ieee.org/document/6120048 (accessed Mar. 21, 2023).

[12] “The Impact of Dragonfly Malware on Industrial Control Systems | SANS Institute.” https://www.sans.org/white-papers/36672/ (accessed Mar. 21, 2023).

[13] “Manage endpoint security in Microsoft Intune | Microsoft Learn.” https://learn.microsoft.com/enus/mem/intune/protect/endpoint-security (accessed Mar. 21, 2023).

[14] M. S. Vardam et al., “Technical support using assistive robotics for physically challenged people,” Proc. - 1st Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2015, pp. 882–886, Jul. 2015, doi: 10.1109/ICCUBEA.2015.175.

[15] G. Bansod, N. Raval, and N. Pisharoty, “Implementation of a new lightweight encryption design for embedded security,” IEEE Trans. Inf. Forensics Secur., vol. 10, no. 1, pp. 142–151, Jan. 2015, doi: 10.1109/TIFS.2014.2365734.

[16] “Analyzing Interoperability and Security Overhead of ROS2 DDS Middleware | IEEE Conference Publication | IEEE Xplore.” https://vpn.sliit.lk/proxy/0d790d1b/https/ieeexplore.ieee.org/document/9837282 (accessed Mar. 22, 2023).

[17] D. Kienzle, N. Evans, and M. Elder, “NICE: Network introspection by collaborating endpoints,” 2013 IEEE Conf. Commun. Netw. Secur. CNS 2013, pp. 411–412, 2013, doi: 10.1109/CNS.2013.6682753.

[18] S. Lagraa, M. Cailac, S. Rivera, F. Beck, and R. State, “Real-Time Attack Detection on Robot Cameras: A Self-Driving Car Application,” Proc. - 3rd IEEE Int. Conf. Robot. Comput. IRC 2019, pp. 102–109, Mar. 2019, doi: 10.1109/IRC.2019.00023.

[19] S. Rivera, S. Lagraa, A. K. Iannillo, and R. State, “Auto-encoding robot state against sensor spoofing attacks,” Proc. - 2019 IEEE 30th Int. Symp. Softw. Reliab. Eng. Work. ISSREW 2019, pp. 252–257, Oct. 2019, doi: 10.1109/ISSREW.2019.00080.

[20] C. Nachreiner, “Anatomy of an arp poisoning attack,” Retrieved July, vol. 4, p. 2005, 2003.

[21] G. F. Lyon, Nmap network scanning: The official Nmap project guide to network discovery and security scanning. Insecure, 2009.