Robotik Sistem Güvenliğinin Doğrulaması için ROS Tabanlı Saldırı Aracı
ROS Based Attack Tool for Verification of Robotic System Security

Yunus Sabri Kırca, Elif Değirmenci, Ahmet Yazıcı, Metin Özkan

Computer Engineering Department
Eskişehir Osmangazi University, Eskişehir, Türkiye
yunussabri.kirca@ogu.edu.tr, edegirmenci@ogu.edu.tr,
ayazici@ogu.edu.tr, meozkan@ogu.edu.tr

Özetçe

Robot İşletim Sistemi (ROS) sektör ve akademide yaygın olarak kullanılan bir ara katman yazılımıdır. Artmakta olan kullanımına rağmen kendi içinde kapsamlı bir siber güvenlik önlemi barındırmamaktadır. Açık kaynak ve ücretsiz bir yazılım olması ROS'un tercih edilirliğini arttırsa da güvenlik konusunda kullanıcıların kendi çözümlerini üretmesi gerekmektedir. Bu çalışmada, önerilen araç kolay testler gerçekleştirmeye odaklı bir ROS saldırı aracıdır. Bu sayede, ROS'un zayıf yönlerinden biri olan güvenlik yöntemleri geliştirme süreci hızlanmaktadır. Modüler yapı sayesinde istenilen saldırı yöntemleri entegre edilebilir. Dolayısıyla ihtiyaca göre daha kapsamlı hale getirilebilir. Saldırı aracında saldırı senaryoları tanımlanır. ROS geliştiricileri, robotik sistemlerinin bir saldırı altında nasıl tepkiler verdiğini ve güvenlik önlemlerinin ne kadar başarılı çalıştıklarını tanımlanan saldırı senaryolarına göre her seferinde aynı tutarlılıkta gerçekleştirebilir. Tutarlı gözlemlenebilir test sonuçları ve testlerin otomatik olarak gerçekleşmesi sayesinde zaman kazancı sağlanmaktadır.

Abstract

Robot Operating System (ROS) is a middleware that is widely used in industry and academia. Despite its increasing use, it lacks cyber security measurements. Although it is open source and free software, it increases the preference for ROS, but users need to produce their solutions for security. In this study, the proposed tool is a ROS attack tool focused on performing easy validation tests. The process of developing security methods, which is one of the weaknesses of ROS, is accelerated. Thanks to the modular structure, desired attack methods can be integrated. Therefore, it can be more comprehensive as needed. Attack scenarios can be defined in the attack tool. ROS developers can perform the same consistency every time, according to the defined attack scenarios, how their robotic systems react under an attack, and how well the security measures work. Time is saved thanks to consistent observable test results and automatic testing.

1. Introduction

Developing technology and rapid increase in production and consumption have made robotics an indispensable need for the industry. Naturally, the importance of robotic systems in human life is constantly increasing and human-robot interaction is increasing day by day. This also makes it easier for malicious people to access robotic systems [1,2]. Therefore, robotic systems need to be developed more safely. When the robotic system is taken over, not only financial losses occur but also physical damage, injuries, and loss of life may occur.

Robot Operating System [3,4] (ROS) is the most widely used robotic Internet of Things (IoT) middleware in industrial and academic fields. However, ROS has many known vulnerabilities and no security features [5-7]. Although a possible attack can cause very serious material and moral problems, security studies in the field of robotics are insufficient.

Robotics and cybersecurity are very different fields, and usually, someone who works in one of these fields falls short in the other. For this reason, testing robotic systems for safety is relatively more difficult. The simplest method of performing security testing is to perform attacks on the system and observe the system. Tests can be performed with a variety of attack tools without the need to write any code. Very few tools have been developed for ROS. Since ROS is a structure that runs on a network, a tool that will attack ROS should also contain features for network attacks only. Given such situations, there is a huge lack of a general hacking tool for performing robotic system security tests. Also, some penetration testing experience is required for attacking robotic systems. More comprehensive attack tools need to be developed in terms of both ease of use and testing.

Penetration tests are performed to test the security of a system in terms of cybersecurity. The use of various Validation and Verification [8] (V&V) methods both produce more stable results and speed up testing processes. Runtime Verification [9] (RV) is a V&V method used to test a system in real-time. In the RV method, the defined rules are tested continuously. It is an easy-to-use method as only error states are defined and there is no need to define every expected event. In many cases, RV methods can continue to be used as a control mechanism even after testing has ended. In fact, RV methods are further developed and structures that can interfere with the system

under the name Runtime Enforcement [10] (RE) are being developed. In addition to the verification process that takes place at the RV, the RE can also intervene in the system as defined. For check the verify the systems at runtime, some of the ROS specific RV tools are available. ROSMonitoring [11,12] and ROSRV [13,14] are the widely known of these. In this study, we use ROSMonitoring for ease of use and fast installation.

In this study, a ROS based attack tool is proposed. With this tool, a system can be attacked, and it can be observed how secure this system is against attacks or how it reacts to attacks. Moreover, this tool has features for RV and testing. In order to use it with RV tools such as ROSMonitoring, it broadcasts the attack information over the system. It is in a structure that is open to adding new attack methods by the user. Time-based attack scenarios can be defined, and attacks can be made automatically. Literally, it is a tool developed to test the security of ROS systems. This tool is still work in progress and more attack methods can be added in the future.

In the following section security attack tools are given in two subsections, ROS-specific attack tools and general security tools. In the third section, the proposed attack tool and the proposed RV method is given in details. In the fourth section, experimental environment and test results are given. And in the last section conclusion and future study is presented.

2. Security Attack Tools

The proposed attack tool has some features from other attack tools as well as having its features. Since ROS is a robotic IoT middleware, it must use network communication. Various network attacks also have effects on ROS. In the literature, there are various attack tools for security aspects. In the following subsections firstly ROS specific attack tools are detailed, then general security tools are explained. Some of the attack tool's attacks are inspired by those tools.

2.1. ROS Specific Attack Tools

In the literature, there are mainly three ROS-specific attack tools. All required ROS to be installed on the device to be attacked. Some of the most popular attack tools for ROS are ROSChaos [15], ROSPenTO [16] and ROSPloit [17]. ROSChaos is a penetration testing tool that is especially designed for exploiting ROS Master API. It can be very destructive to ROS systems without any security enhancements [18]. On the contrary, ROSPenTO is another penetration testing tool that can perform various attacks with minimal communication with the ROS Master API [18]. Which includes isolating services, nodes, injecting false data, etc. ROSPloit, [19] is a tool that can perform both reconnaissances and attack a little more extensively than other tools. It can perform a comprehensive ROS scan thanks to the NMAP [20] (Network Mapper) library it basically uses. It can also perform many attacks such as Denial of Service (DoS), Man in the Middle (MiTM), unauthorized data access and unauthorized data publishing.

2.2. General Security Tools

Since network attacks have an impact on ROS, it would be very insufficient for security to limit attack tools to tools developed only for ROS. For this reason, some features of common tools used for network security have also been included or may be added to the proposed attack tool. The modular nature of the attack tool allows users to add any attacks they want. Some attack methods from tools such as NMAP, Hping3 [21], Metasploit Framework [22], have been added as examples. NMAP is a network scanner tool that is used to discover hosts and services on a computer network by sending packets and analyzing the responses. Hping3 is an open-source packet generator and analyzer with lots of other capabilities. DoS attacks also can be performed by generating tons of network packets. Metasploit Framework is a tool for developing and executing exploit code against a remote target machine. It has a wide variety of attack scenarios. Some attacks that may have an impact on ROS are used in attack tool, especially by performing volumetric attacks on the network.

3. Proposed Tool and Method

In this study, a proposed tool for robotic system security and a test method using this tool are proposed.

3.1. ROS Based Attack Tool

The main aspect of the proposed tool is performing long-lasting and varied attacks to the test security of a ROS system. Continually running multiple attack tests can be tedious and time-consuming. This attack tool has been developed to observe the effects of cyber-attacks on ROS and to test attack and prevention method for ROS, if any.

The tool has several unique abilities that other ROS attack tools don't have. It, act like part of ROS mechanism and publish its data about attacks. This information can be used with RV tools like ROSMonitoring or ROSRV for testing, marked data collecting, etc. Also, attacks are time-based programmable so, which attack, when will be executed, for how much time, on which target, with how much attack volume, etc. like parameters can be defined by the users. Attacks and auxiliary scripts are stored in file folders separated which makes importing, exporting, and modifying attacks like operations are easy.

3.2. Model Based RV Method

The proposed method is given to perform an example test of the use of the ROS Attack tool. The method that can be seen in Figure 1. roughly tests an extreme network use case on the network. To perform this method requires two types of data. The first of these is the real-time bandwidth values used by the devices used on the network. The other one is the activity level of the system. Having the devices in the system in an idle or a running state will increase the bandwidth usage. A smart structure can also be developed for the secure bandwidth upper limit, but in this study, estimated values are given for a simple test. Finally, with given parameters and defined rules on ROSMonitoring, RV operation can be performed for this method.

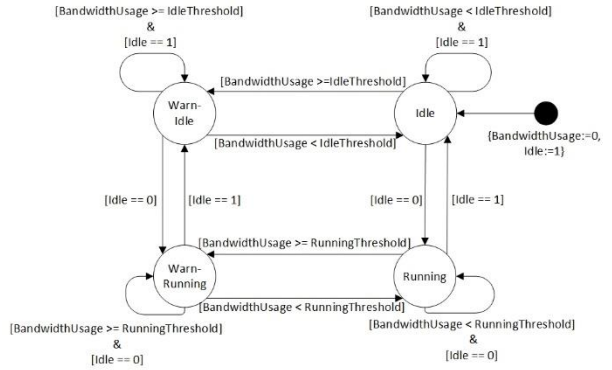

Figure 1: Model-Based RV.

With the proposed method, it can be said that an anomaly network usage can be detected in the system. By performing various attacks, their effects on the system can be observed. However, it can be difficult to observe test results, given that data is flowing rapidly and in large quantities. While different types of attacks are carried out in one scenario, the data can be mixed. At this point, many different attacks can be carried out by using the proposed attack tool together with a tool such as ROSMonitoring. Attacks can be observed much more efficiently with specific attack periods and labeling of generated data.

4. Experimental Results

ROS-based attack tool and Model-based RV method are proposed in this study. We evaluate an experimental study with this attack tool and RV method.

4.1. Environment Setup

The state of meeting the desired needs of a robotic system is checked by performing tests. Tests in the field of robotics are primarily carried out in simulation environments in order not to cause any damage. Correct performance of the developed software, problems that may arise in terms of safety like the probability of accidents, etc. situations are tested in simulation environments. Cyber-attacks also can cause safety issues. However, attacks on the virtual environment and the real robot can cause very different results. If the simulation environment runs on a single device, the desired effect may not be obtained from network-based attacks. In order to avoid this situation, while modeling the devices in a real environment, the devices that control each device can be separated from each other virtually or physically as seen in Figure 2.

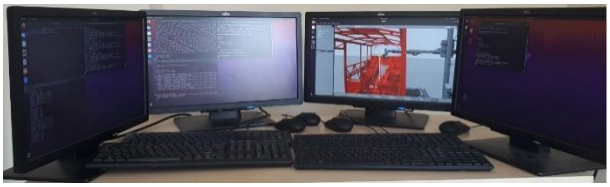

Figure 2: Physical test environment.

The test environment's logical data flow is given in Figure 3. In this study's setup, robotic platform is a ROS and GAZEBO [23] simulation platform. Verification System is a device that ROSMonitoring is running with proposed method. A switch that Robotic Platform communicates, is reflecting network data to Verification System. As last ROS Attack tool runs on Attacker System device. Attacker System perform attacks to any Robotic Platform device also Publishes Attack State or any other attack information to Verification System.

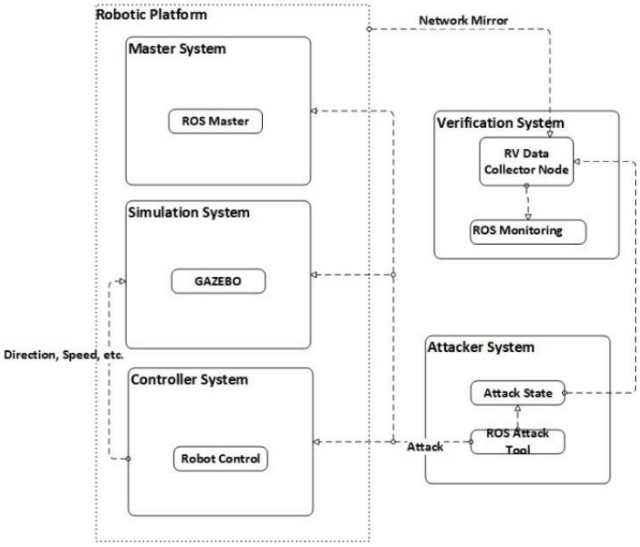

Figure 3: Data stream of test environment.

4.2. Test Results

The attack scenario was carried out as given in Table 1. While there is no attack at idle, an attack occurs during DoS that will cause some traffic in the system. The unit of time is seconds and can be compared with Figure 4. Communication takes place according to the data flow chart given in Figure 3. In addition, the Model-Based RV model given in Figure 1 observes an unexpected traffic volume in the system according to the network bandwidth usage.

Table 1: Step by step attack scenario.

System and Attack State	***Duration***
Idle State, No Attack	60s
Idle State, DoS Attack	120s
Idle State, No Attack	60s
Running State, No Attack	60s
Running State, DoS Attack	120s
Running State, No Attack	60s
Idle State, No Attack	60s

Model result from the RV model and the attack status from the attack tool are combined on ROSMonitoring to verify model's reliability. If the results returned from the model and the results returned from the attack tool do not match, they are given as red dots on the graph in Figure 4. Confirmed points are not red. These points seem to be inaccurate as there is some delay between the attack start and end moments of the attack tool and the observation of the effects of these situations on the system.

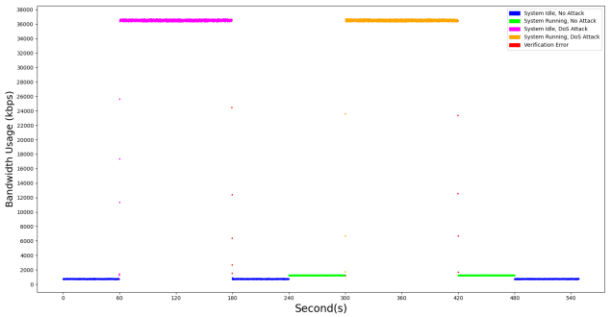

Figure 4: Verification results with bandwidth usage.

Due to Figure 4, different states those given in Figure 1, are marked as different colors. Model-based RV method is verified for every points except red ones.

5. Conclusion

In this study, an attack tool and a model-based RV method are proposed. The proposed attack tool has been developed to verify the security of ROS-based robotic systems. It will be able to stay up-to-date in terms of attacks, as it allows users to add attack methods as they wish. By defining time-based attack scenarios, high comparability data can be produced thanks to attacks that are carried out completely similarly by different defense mechanisms. The attack tool can also attack open points on the robotic system with various reconnaissance methods. By performing the same attack scenario each time at different points in the system, the reactions to the attacks can be observed. The proposed attack tool saves time in the development of robotic system security studies, thanks to its easy use with GUI and automatic execution of many operations.

In order to carry out the test of the attack tool, a method had to be proposed in this study. The model-based RV method monitors the used bandwidth of the robotic system in real-time and determines the situation according to the determined threshold values of the traffic. By comparing the result of the method and the attack status broadcast from the attack tool, the system's ability to produce correct results has been observed.

As future work, various RV methods can be added to the proposed attack tool, as well as new attack features. As it was developed to help verify robotic system security on the basis of the proposed attack tool, it may contain defensive methods in the future, making it more comprehensive.

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 program and Austria, Czech Republic, Germany, Ireland, Italy, Portugal, Spain, Sweden, Turkey.

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 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 ".

References

[1] B. Dieber, B. Breiling, S. Taurer, S. Kacianka, S. Rass, and P. Schartner, "Security for the robot operating system," *Robotics and Autonomous Systems*, vol. 98, pp. 192–203, 2017.

[2] N. DeMarinis, S. Tellex, V. P. Kemerlis, G. Konidaris and R. Fonseca, "Scanning the Internet for ROS: A View of Security in Robotics Research," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8514-8521, doi: 10.1109/ICRA.2019.8794451.

[3] "ROS Smach," Jul. 18, 2022. http://wiki.ros.org/smach (accessed Jul. 18, 2022).

[4] A. Koubâa, *Robot Operating System (ROS).*, vol. 1. Springer, 2017.

[5] J.-P. A. Yaacoub, H. N. Noura, O. Salman, and A. Chehab, "Robotics cyber security: Vulnerabilities, attacks, countermeasures, and recommendations," *International Journal of Information Security*, pp. 1–44, 2021.

[6] R. R. Teixeira, I. P. Maurell, and P. L. J. Drews, "Security on ROS: analyzing and exploiting vulnerabilities of ROS-based systems," in *2020 Latin American robotics symposium (LARS), 2020 Brazilian symposium on robotics (SBR) and 2020 workshop on robotics in education (WRE)*, 2020, pp. 1–6.

[7] S.-Y. Jeong *et al.*, "A study on ros vulnerabilities and countermeasure," in *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 2017, pp. 147–148.

[8] J. O. Grady, *System validation and verification*, vol. 12. CRC Press, 1997.

[9] E. Bartocci, Y. Falcone, A. Francalanza, and G. Reger, "Introduction to runtime verification," in *Lectures on Runtime Verification*, Springer, 2018, pp. 1–33.

[10] S. Pinisetty, P. S. Roop, S. Smyth, N. Allen, S. Tripakis, and R. von Hanxleden, "Runtime enforcement of cyber-physical systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 5s, pp. 1–25, 2017.

[11] A. Ferrando, R. C. Cardoso, M. Fisher, D. Ancona, L. Franceschini, and V. Mascardi, "ROSMonitoring: a runtime verification framework for ROS," in *Annual Conference Towards Autonomous Robotic Systems*, 2020, pp. 387–399.

[12] "ROSMonitoring: A Runtime Verification Framework for ROS" https://github.com/autonomy-and-verification-uol/ROSMonitoring (accessed July 29, 2022)

[13] J. Huang *et al.*, "ROSRV: Runtime verification for robots," in *International Conference on Runtime Verification*, 2014, pp. 247–254.

[14] "ROSRV: Runtime Verification for Robot Operating System" https://github.com/Formal-Systems-Laboratory/ROSRV (accessed July 29, 2022)

[15] "Audit and test security vulnerabilities or simply wreak chaos upon your ROS system" https://github.com/ruffsl/roschaos (accessed July 29, 2022)

[16] "The Robot Operating System (ROS) penetration testing tool (ROSPenTo) can send XML remote procedure calls (XMLRPC) to the ROS-Master and to ROS-Nodes." https://github.com/jr-robotics/ROSPenTo (accessed July 29, 2022)

[17] “ROSploit is a collection of tools and scripts designed to aid in the exploitation of ROS(Robotic operating system) robots.” https://github.com/seanrivera/rosploit (accessed July 29, 2022)
[18] B. Dieber *et al.*, “Penetration testing ROS,” in *Robot operating system (ROS)*, Springer, 2020, pp. 183–225.
[19] S. Rivera, S. Lagraa, and R. State, “Rosploit: Cybersecurity tool for ROS,” in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, 2019, pp. 415–416.
[20] “Open source utility for network discovery and security auditing.” https://github.com/nmap/nmap (accessed July 29, 2022)
[21] “A network tool able to send custom TCP/IP packets and to display target replies like ping do with ICMP replies.” https://github.com/antirez/hping (accessed July 29, 2022)
[22] “Very powerful tool which can be used by cybercriminals as well as ethical hackers to probe systematic vulnerabilities on networks and servers.” https://github.com/rapid7/metasploit-framework (accessed July 29, 2022)
[23] “Gazebo,” Jul. 18, 2022. https://gazebosim.org/ (accessed July 29, 2022)