Case study: Remote attack to disable MiR100 safety

Sebastian Taurer, Benjamin Breiling, Stella Svrta, Bernhard Dieber

February 12, 2020

Abstract

In this abstract, we describe a case study where we remotely disabled the safety subsystem of a MiR100 industrial mobile robot. Due to several misconfigurations and negligence of standard security procedures (like changing default passwords), it is possible to retrieve, manipulate and reupload the safety program logic running on the dedicated safety PLC in the robot.

We sketch the attack vector and describe its effects and possible mitigation strategies. The vulnerability described has been acknowledged by the robot manufacturer and is being addressed.

1 Introduction

The convergence of general-purpose computing, IoT and robotics has strongly pushed the development of new robotic products in industry and other domains. Those IoT robots combine the comfortable programmability of general-purpose computer systems and their modern programming language with the manipulation capabilities of robots enhanced by their interconnected nature. However, this trend also brings along the well-known security risks of IoT [10, 11, 12] and IT.

Robot security vulnerabilities are especially grave if they concern the robot's safety systems since this can directly affect human lives. The research interest in robot cybersecurity has gradually increased in recent years internationally [2, 6, 5, 1, 9, 7]. Very recently, also the first vulnerability database specialized in robot vulnerabilities has been presented[1] [8]. While we have shown in earlier work how ROS-based software of robots can be manipulated[2] [4, 3], in this work we demonstrate an attack on a robot's safety subsystem.

In a case study, we have managed to remotely (via WiFi) disable the safety laser scanners of the MiR100[3] robot. The main reasons for this lie in relying on default passwords and misconfigurations of the internal network (i.e., it could be easily prevented). The passwords for WiFi, the configuration web-interface of the router and the safety PLC are default and thus easy to find (e.g. in manuals). This allowed us to 1. read out the factory-installed safety program, 2. modify

[1] `https://github.com/aliasrobotics/rvd`

[2] `http://bernharddieber.com/post/mir-hacking-video/`

[3] `https://www.mobile-industrial-robots.com/en/products/mir100/`

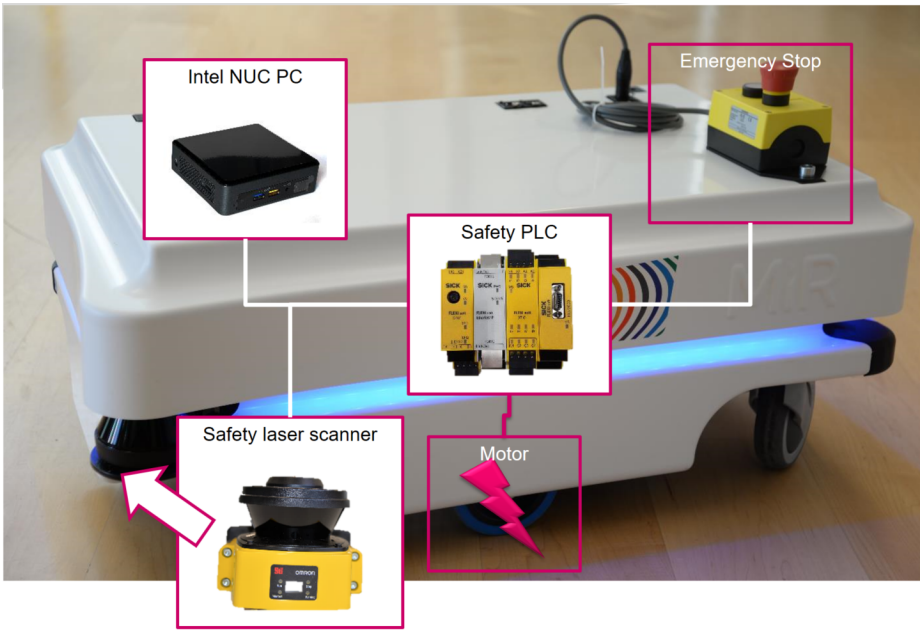

Figure 1: The relevant components of the robot: Laser scanners that secure the immediate surroundings as well as the emergency stop are connected to the safety PLC which can directly cut power to the motors in case of an emergency.

the safety program logic, 3. upload the manipulated program remotely and effectively disable the emergency stop in case an object is too close to the robot. Navigation and any other component depending on the laser scanners are not affected (thus it is hard to detect before something happens).

In the best tradition of responsible disclosure, we have provided all available information to the manufacturer. While initially unresponsive, the manufacturer has meanwhile started to address the issue.

2 Attack description

2.1 Preconditions

Preconditions for a successful attack is that the attacker can penetrate the WiFi the robot is connected to. Since the default WIFI password for all MiR robots is the same and they are hardly ever changed by system integrators, this is no hurdle for an attacker. Robots that are integrated into the manufacturing WiFi can be attacked over that channel.

All required passwords can be found in respective manuals.

2.2 MiR100 internals

In the MiR100, the two laser scanners (front and back) are used for safety purposes as well as for navigation. On the one hand the safety laser scanners are hardwired to the SICK safety PLC (data=high or low signal) and on the

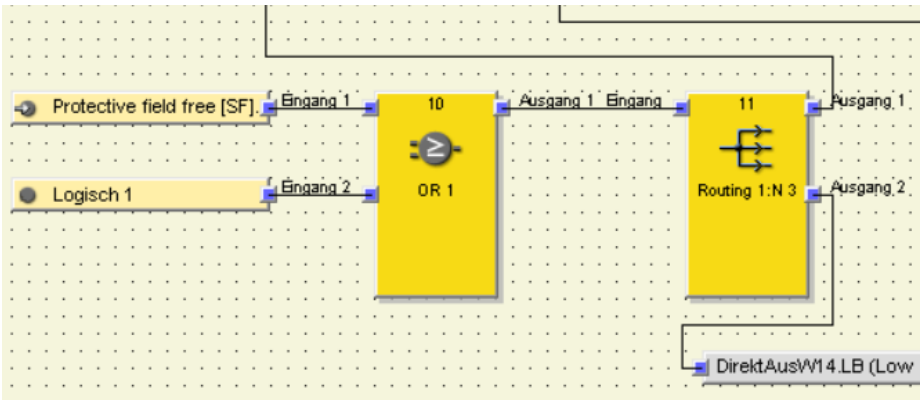

Figure 2: The manipulated PLC program. Block 11 normally takes the laser scanner (protective field free) as input. We inserted an OR block with an additional constant 1 for that signal. We could also connect the 1 directly to block 11 but for demonstration purposes we can also show the original signal in the live visualization. The same is done for block 9 (second laser scanner).

other hand they are connected via USB to the internal NUC PC which exposes their data to the ROS network to provide the measured data (data=array of distances). The safety PLC performs all safety functions (i.e., turning off the motors if an object is too close to a laser scanner). The safety system cannot be attacked via the ROS layer since there data is only consumed. Figure 1 shows the aforementioned components and their interconnections.

2.3 Summary of the procedure

The SICK safety PLC can be accessed remotely by anyone in the robot's internal network. With a simple *arp* or *nmap* scan, the IP address can be retrieved.

We used the freely available SICK Flexi Soft Designer[4] to connect to the PLC and read out the safety program logic stored on it. We altered the software in order to make the PLC ignore the input of the laser scanners (see figure 2).

On transfer of the program, a password for the PLC is required. However, the default password from the SICK manual has not been changed by the robot OEM. After a reboot of the PLC, the signals of the safety laser scanner are ignored in the PLC program and the robot will run over any obstacle in its way.

Figure 3 shows the effect of the attack. In the first sequence of pictures, the MiR with intact safety functions stops before an object (as indicated by the red lights). The second sequence shows a manipulated robot that runs over the person standing in the way.

3 Attack analysis and mitigation

The main reason why we could perform this is 1. that the safety PLC is connected to the internal network (which is not required for normal operation) and 2. that the SICK safety PLC is configured with the default password that can

[4] `https://www.sick.com/us/en/flexi-soft-designer/p/p81369`

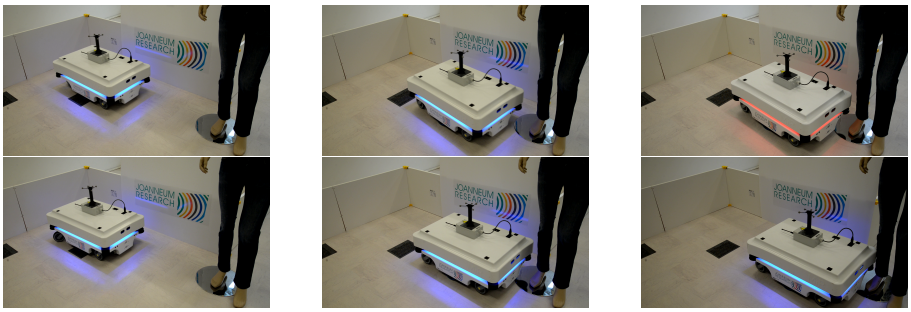

Figure 3: A MiR100 approaching an obstacle with safety intact (above) and with our manipulation. It can be seen that in the first sequence, the robot stops just before the person. In the second sequence, the robot collides.

be found in the manual of the respective software. Thus, it is easy to manipulate the software running on it.

By changing default passwords and adapting the internal routing and firewall settings in the robot, this attack could easily be prevented.

With the same attack vector, further manipulations are possible (e.g., tampering with the laser scanners' configurations).

4 Conclusion

Modern, IoT-based robots pose great potential for economic and societal profits. The companies driving the developments are typically not the established robot manufacturers but younger, start-up like companies. Their first priority is to establish their products on the market and to grow very fast. Properties like security are in such endeavours often regarded as secondary (at best) issues that can be addressed later-on.

However, history may overtake them in this regard as soon as real-world incidents start to happen. However, it is well-known that addressing security as early as possible during development is by orders of magnitude cheaper (and more secure) than adding security on-top of an existing product at later stages. Ideally, security-related activities like threat modelling and penetration testing are part of a development process.

The case study presented in this paper represents quite well the current state of security awareness in the development of IoT robots. All the insufficiencies that lead to the current vulnerability could easily be mitigated by appropriate configurations of the robot components. Now that several thousand of those robots are being used around the world, mitigation is way more expensive.

To prevent such vulnerabilities from leading to loss or harm, security must reach a higher priority during development. It stands to hope that either elevated awareness or market pull by end-users and not security incidents cause that.

References

[1] Benjamin Breiling, Bernhard Dieber, and Peter Schartner. Secure communication for the robot operating system. In *Proceedings of the 11th IEEE International Systems Conference*, pages 360–365, 2017.

[2] Tamara Denning, Cynthia Matuszek, Karl Koscher, Joshua R. Smith, and Tadayoshi Kohno. A spotlight on security and privacy risks with future household robots: Attacks and lessons. In *Proceedings of the 11th International Conference on Ubiquitous Computing*, UbiComp '09, pages 105–114, New York, NY, USA, 2009. ACM.

[3] Bernhard Dieber, Benjamin Breiling, Sebastian Taurer, Severin Kacianka, Stefan Rass, and Peter Schartner. Security for the robot operating system. *Robotics and Autonomous Systems*, 98:192–203, 2017.

[4] Bernhard Dieber, Ruffin White, Sebastian Taurer, Benjamin Breiling, Gianluca Caiazza, Henrik Christensen, and Agostino Cortesi. *Penetration Testing ROS*, pages 183–225. Springer International Publishing, Cham, 2020.

[5] Francisco Javier Rodrıguez Lera, Jesús Balsa, Fernando Casado, Camino Fernández, Francisco Martın Rico, and Vicente Matellán. Cybersecurity in autonomous systems: Evaluating the performance of hardening ros. In *Workshop on physical Agents*, page 47, 2016.

[6] J. McClean, C. Stull, C. Farrar, and D. Mascareñas. A preliminary cyber-physical security assessment of the robot operating system (ros). In *Proc. SPIE*, volume 8741, pages 874110–874110–8, 2013.

[7] Víctor Mayoral Vilches, Laura Alzola Kirschgens, Asier Bilbao Calvo, Alejandro Hernández Cordero, Rodrigo Izquierdo Pisón, David Mayoral Vilches, Aday Muñiz Rosas, Gorka Olalde Mendia, Lander Usategi San Juan, Irati Zamalloa Ugarte, et al. Introducing the robot security framework (rsf), a standardized methodology to perform security assessments in robotics. *arXiv preprint arXiv:1806.04042*, 2018.

[8] Vıctor Mayoral Vilches, Lander Usategui San Juan, Bernhard Dieber, Unai Ayucar Carbajo, and Endika Gil-Uriarte. Introducing the robot vulnerability database (rvd). In *Proceedings of the 2020 Fourth International Conference on Robotic Computing*, 2020.

[9] Ruffin White, Gianluca Caiazza, Henrik Christensen, and Agostino Cortesi. *SROS1: Using and Developing Secure ROS1 Systems*, pages 373–405. Springer International Publishing, Cham, 2019.

[10] Teng Xu, James B. Wendt, and Miodrag Potkonjak. Security of iot systems: Design challenges and opportunities. In *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '14, pages 417–423, Piscataway, NJ, USA, 2014. IEEE Press.

[11] Z. Zhang, M. C. Y. Cho, C. Wang, C. Hsu, C. Chen, and S. Shieh. Iot security: Ongoing challenges and research opportunities. In *2014 IEEE 7th*

International Conference on Service-Oriented Computing and Applications, pages 230–234, Nov 2014.

[12] K. Zhao and L. Ge. A survey on the internet of things security. In *2013 Ninth International Conference on Computational Intelligence and Security*, pages 663–667, Dec 2013.