Journal of Cyber Forensics and Advanced Threat Investigations

CFATI OPEN ACCESS

Volume X(Y), Z—U. http://dx.doi.org/xxx-xxx-xxx

Robot teardown, stripping industrial robots for good

Víctor Mayoral-Vilches[1,3], Alfonso Glera-Picón[1], Unai Ayúcar-Carbajo[1],
Stefan Rass[3], Martin Pinzger[3], Federico Maggi[2], Endika Gil-Uriarte[1]

Abstract

Building a robot requires careful selection of components that interact across networks while meeting timing deadlines. Given their complexity, as robots get damaged or security compromised, their components will increasingly require updates and replacements. Contrarily to the expectations and similar to Ford in the 1920s with cars, most robot manufacturers oppose to this. They employ planned obsolescence practices organizing dealers and system integrators into ”private networks”, providing repair parts only to ”certified” companies to discourage repairs and evade competition.

In this article we introduce and advocate for *robot teardown* as an approach to study robot hardware architectures and fuel security research. We show how systematic teardowns can help understanding the underlying hardware and demonstrate how our approach can help researchers uncovering security vulnerabilities. Our case studies show how robot teardown becomes an essential practice to security in robotics, helping us identify and report a total of 100 security flaws with 17 new CVE IDs over a period of two years. Lastly, we finalize by demonstrating how, through teardown, planned obsolescence hardware limitations can be identified and bypassed obtaining full control of the hardware, which poses both a threat to the robot manufacturers’ business model as well as a security threat.

Notes for Practice

- Following a red teaming methodology, we discuss the empirical results of three robot teardowns performed on popular industrial collaborative robots and uncover various quality and safety flaws in the process.
- We gain repairing capabilities in the robots which leads us to acquire means to mitigate security flaws early by simply extending the robotic system with off-the-shelf additional hardware elements.
- We show how robot teardown helps pinpoint security vulnerabilities across internal and external robot networks while discussing some of them.
- We show evidence of planned obsolescence practices in robotics on leading industrial collaborative robots and demonstrate how by applying minor fixes, we managed to bypass the obsolescence limitations.

Keywords
teardown, robotics, security, repair, safety

Submitted: 05/26/21 — **Accepted:** 22/07/21 — **Published:** -/-/-

[1] *Alias Robotics, Venta de la Estrella 3, Pab. 130, Vitoria, 01005 Spain, [*`victor|alfonso|unai|endika`*]@aliasrobotics.com*
[2] *Trend Micro Inc., Italy, Federico_Maggi@trendmicro.com*
[3] *Universität Klagenfurt, Universitätsstraße 65-67, 9020 Klagenfurt, Austria, [*`Stefan.Rass |Martin.Pinzger`*]@aau.at, v1mayoralv@edu.aau.at*

1. Introduction

Robotics is the art of system integration, as noted by Mayoral-Vilches et al. (2017). Building a robot requires one to carefully select components that exchange information across networks while meeting timing deadlines. In a way, a robot is a network of networks. One that comprises sensors to "read" the world, actuators to produce a physical change, and one for dedicated computational resources to process it all and respond coherently, in time, and according to its application. Roboticists often conceive the robot not as one of its parts, but as the complete system including all of its components, whether they are assembled under the same structure or physically distributed. In the case of a robotic manipulator, these robots are often presented physically distributed and include the robot arm mechanics (which generally include actuators and sensors), the human-machine interface (HMI) or teach pendant, the controller (the main compute substrate for reasoning), and any additional safety mechanism related to the robot operation. The robotic system is thereby the *composition* of all these sub-systems and networks.

Under such system integration complexity, it is not uncommon for one of the robot sub-components to fail over time, often leading to the complete system malfunction. Given the high price point of robots, it is reasonable to consider the need for repairing these machines, often replacing individual faulty components for continued operation, or simply for re-purposing them. The European Commission (EC) showed early interest on this topic in a report by Cordella, Alfieri, and Sanfelix (2019) evaluating different scoring systems for repairing and upgrading different consumer-oriented products, including robots. More recently, and as part of the Circular Economy Action Plan for Communication (European Commission) (2020), the EC has shown commitment towards establishing a new *Right to Repair* in the context of reviewing directive 2019/771. Hatta (2020) summarizes major events in the U.S. with regard the *Right to Repair* and highlights that it wasn't until 2012 that the Automotive *Right to Repair* passed in Massachussets, empowering customers with tools to fight planned obsolescence. Hatta summarizes how material obsolescence works:

- Making items difficult to repair (by raising the cost of repair, requiring special tools, etc.)
- Failing to provide information (for instance, manuals are not provided)
- Systematic obsolescence (making parts among models incompatible or making it impossible to fix newer models with parts from the older models)
- Numbering (frequently changing the model numbers to make it psychologically less attractive to use old models)
- Legal approaches (prohibiting access and modification to the internal structure of products by means of copyrights and patents)

Hatta (2020) noticed that, similar to Ford in the 1920s, most robot manufacturers follow several of these practices nowadays and organize dealers (often called distributors) or approved system integrators into private networks, providing repair parts only to *certified* companies in an attempt to discourage repairs and evade competition. Amongst the most recent examples we observe an interesting development from Teradyne, where two of its owned robotics companies (Universal Robots and Mobile Industrial Robots), follow this practice. The case of Teradyne is of special interest because its robots are advertised as collaborative, that is: designed to augment human capabilities by closely (physically) cooperating without causing any harm. Past research however hints that the lack of security measures in these robots leads to safety hazards, as concluded by Alzola Kirschgens, Zamalloa Ugarte, Gil Uriarte, Muñiz Rosas, and Mayoral-Vilches (2018); Mayoral-Vilches, Juan, Carbajo, et al. (2019); Taurer, Breiling, Svrta, and Dieber (n.d.).

Cybersecurity in robotics is still on its early stages, as demonstrated by Quarta et al. (2017); Alzola Kirschgens et al. (2018); Pogliani, Maggi, Balduzzi, Quarta, and Zanero (2020). Therefore, as in many other fields, it remains addressed mostly in disconnected silos. With most efforts concentrated in IT, hardware security in robotics has received very limited attention. Building secure robots, however, demands consideration throughout domains (hardware, firmware, OS, application, network, cloud, etc.), Mayoral-Vilches et al. (2018), and across the robot lifecycle, Mayoral-Vilches, García-Maestro, Towers, and Gil-Uriarte (2020).

The present article introduces and promotes robot teardown as a systematic *process* to repair robots, improve robot hardware and research its security. We advocate against the business priorities set in industry, which avoid repairs and planned obsolescence. Instead, we advocate for a *Right to Repair* in robotics as a means to reduce robot e-waste and recycle components, both across robots and throughout use-cases. Ultimately, we argue that, in the long run, the more researchers and practitioners will learn to systematically teardown robots, the more this practice will impact the quality assurance of hardware in robotics, putting pressure on manufacturers to produce robots with better (hardware) security measures, thereby safer. Our contributions are fourfold: first, we discuss the empirical results of three robot teardowns performed on popular industrial collaborative robots and uncover various quality, security, and safety flaws in the process. Second, we demonstrate how, as a result of a teardown, we gain repairing capabilities in the robots. This leads us to acquire means to mitigate security flaws early, by simply extending the robotic system with additional, off-the-shelf hardware elements that increase the overall cybersecurity posture with a minimal cost impact. Third, we show how teardown helps pinpoint security vulnerabilities across internal and external robot networks while discussing some of them. Fourth, we show evidence of planned obsolescence practices in robotics on leading industrial collaborative robots and demonstrate how by applying minor fixes, we managed to bypass the obsolescence limitations obtaining full control of the hardware across subsequent releases.

The content below is organized as follows: Section 2 describes the robot teardown process in three different robots and Section 3 the posterior reversing exercise to gain repairing capabilities. Section 4 argues about the obsolescence indicators encountered and demonstrates how to bypass them as a result of the results in previous sections. Finally, Section 5 summarizes our work and draws some conclusions.

2. Robot teardown

A teardown is the process of taking apart a product to understand how it is made and works. More formally, it is the approach to modeling the functional behavior and physical components of a product, as described by Younis and Tutunji (2010); Skorobogatov (2017); Tutunji (n.d.). Robot teardown is thereby the process to study robot hardware architectures through systematic disassembly to understand how the robot works and what physical sub-systems compose it.

The motivation behind teardowns was previously researched by other groups such as Kohlweiss, Auberger, Ketenci, and Ramsauer (2020); Sandborn, Myers, Barron, and McCarthy (2006). In robotics, we identify three key purposes: a) dissection and analysis to evaluate the status of a product, b) competitive benchmarking against similar products, and c) gain engineering experience and knowledge. This paper focuses on a) and c). Particularly, we show three case studies on the robots from Universal Robots (UR) and Mobile Industrial Robots (MiR). Our motivation for selecting these targets is twofold: first, these robots are arguably widely used across use cases in the professional and industrial environments, with tenths of thousands of units sold Crowe (2021) and operating in close contact with humans (as collaborative robots). Second, past research by Cerrudo and Apa (2017a, 2017b); Mayoral-Vilches et al. (2018); Zhu, Rass, Dieber, and Vilches (2021); Mayoral-Vilches, Juan, Dieber, Carbajo, and Gil-Uriarte (2019); Taurer et al. (n.d.) has shown a lack of security concern and readiness from these two manufacturers making them attractive targets for adversaries aiming to disrupt industrial processes or causing injuries as reported by Alzola Kirschgens et al. (2018). Disruption-based attacks, unfortunately, continue to be the most effective leverage used by financially driven threat actors such as DarkSide[1], just to name the most damaging and recent one (as of June 2021).

Based on common teardown practices Kohlweiss et al. (2020); Sandborn et al. (2006), we present in Figure 1 our teardown methodology for robots. To the best of our knowledge, we are the first to propose and document a teardown approach for industrial robotic products. The following subsections provide a walk-through on three case studies and discuss the most interesting findings on each one of them.

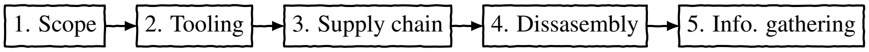


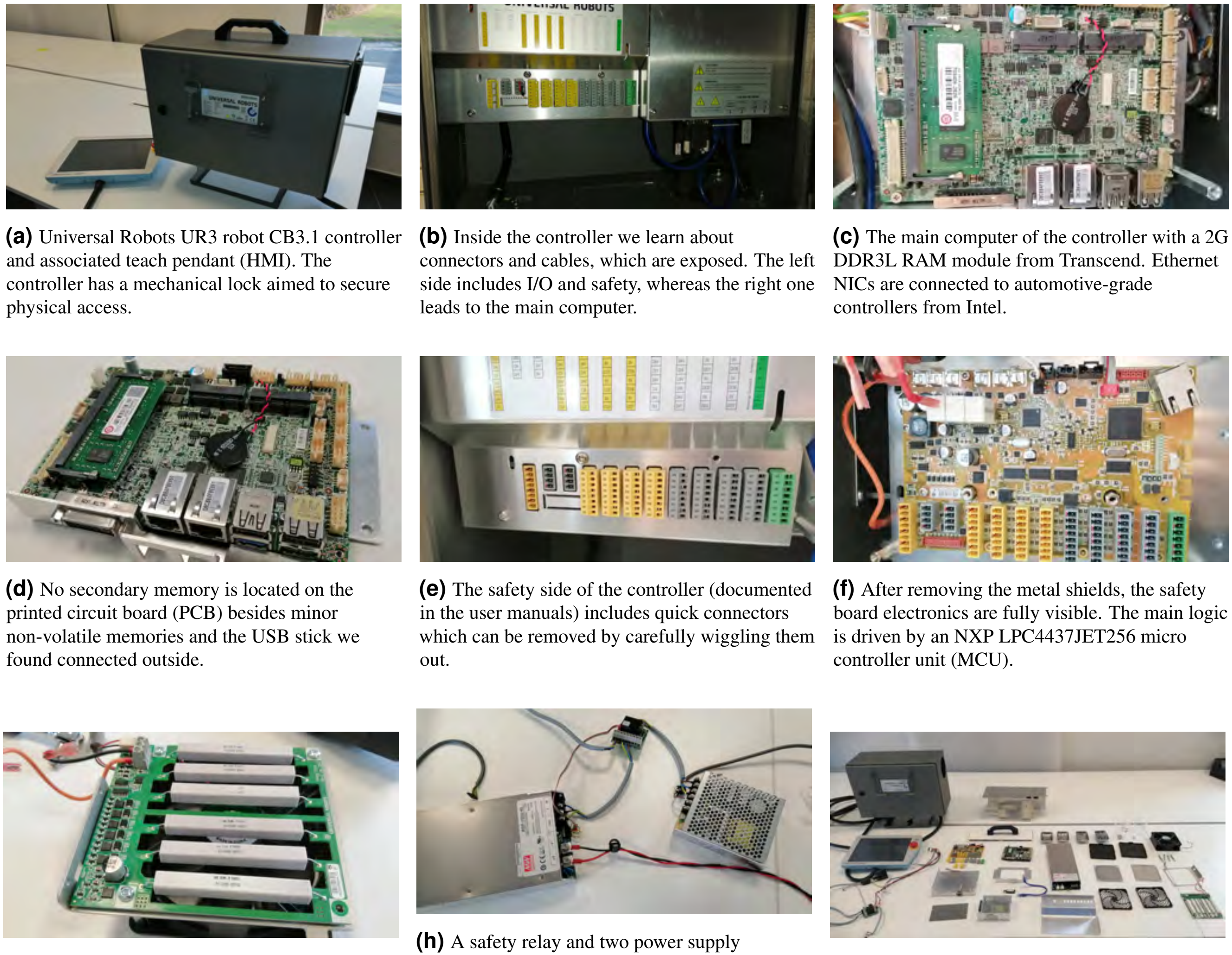
Figure 1. Our teardown methodology for robots. The process involves 5 steps: **1.** Identifies the purpose and scope of the teardown exercise. **2.** Prepares for the teardown gathering required tools for documentation and dissassembly. **3.** Examines the supply chain identifying how to acquire parts, what's the installation process and who's entitled for repairs, including costs and liabilities. **4.** Takes apart the robot, documenting each step and avoiding the damage of any component. **5.** Extracts relevant data (e.g. firmware version) from each robot component, constructs a bill of materials (BOMs) and gathers additional information by researching public resources.

2.1 Case Study 1: Teardown of an industrial collaborative robot

Figure 2 shows a selection of images obtained from the complete teardown of the UR3 CB3.1 industrial collaborative robot. Our goal is to show how a systematic teardown can lead to understanding how to obtain repairing capabilities of the complete robot, including the controller (i.e., the "brain" of the robot), teach pendant, and robot arm mechanics. We put particular emphasis in the CB3.1 controller since most safety-related electronics reside in there. The total time spent in the teardown is of five hours and 30 minutes.

An interesting observation is depicted in Figure 2f, which displays that the compute substrate in charge of implementing the safety logic is the NXP LPC4437JET256 microcontroller. While doing hardware reconnaissance we found the following excerpt within the part datasheet[2] of the corresponding microntroller:

[1] https://www.intel471.com/blog/darkside-ransomware-shut-down-revil-avaddon-cybercrime
[2] https://www.nxp.com/docs/en/data-sheet/LPC435X_3X_2X_1X.pdf



(a) Universal Robots UR3 robot CB3.1 controller and associated teach pendant (HMI). The controller has a mechanical lock aimed to secure physical access.

(b) Inside the controller we learn about connectors and cables, which are exposed. The left side includes I/O and safety, whereas the right one leads to the main computer.

(c) The main computer of the controller with a 2G DDR3L RAM module from Transcend. Ethernet NICs are connected to automotive-grade controllers from Intel.

(d) No secondary memory is located on the printed circuit board (PCB) besides minor non-volatile memories and the USB stick we found connected outside.

(e) The safety side of the controller (documented in the user manuals) includes quick connectors which can be removed by carefully wiggling them out.

(f) After removing the metal shields, the safety board electronics are fully visible. The main logic is driven by an NXP LPC4437JET256 micro controller unit (MCU).

(g) The energy-eater board. This component tends to overheat a fair bit and should generally be checked in case of failure for signs of degradation.

(h) A safety relay and two power supply units (PSUs) identified, one for the compute logic (12V) and another one to power the actuators (48V).

(i) Final figure depicting all the components contained inside of the Universal Robots UR3 CB3.1 controller, leaving aside the teach pendant.

Figure 2. UR3 collaborative robot teardown.

> **Suitability for use** — NXP Semiconductors products are ***not designed****, authorized or warranted to be suitable for use in life support, life-critical or* ***safety-critical systems*** *or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in* ***personal injury****, death or severe property or environmental damage*. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Observation 1 *The microcontroller implementing the safety logic in the UR3 CB-Series robot controller is in fact not suitable for safety-critical systems according to the silicon vendor. Confusingly, the list of applications on the first pages of the datasheet includes industrial automation or motor control, which are typical safety-critical use cases.*

In other words, this observation leads us to question the quality and reliability of the safety implementation within robots of the CB-Series by Universal Robots. Hatta (2020) indicated that vendors have historically opposed to teardowns under the argument that *closed networks of dealers guarantee quality*. However, our first observation indicates the exact opposite. Third parties with the required technical expertise might be able to identify and pinpoint hardware components that don't meet the quality standards for the safety situations the robot may have to face, leading to an overall improved scenario for end-users.

2.2 Case Study 2: Teardown of a next-gen industrial collaborative robot

Following the CB-Series, we proceeded and disassembled one of the latest releases from Universal Robots, the UR3e, an e-Series. Figure 3 depicts the complete process through selected images. The complete teardown of the robot took us a total time of three hours[3].

We observe how, while the overall outer look remains similar, the internals have suffered a significant change:

- The e-Series controller integrates a single PSU, as presented in Figure 3e, while the CB-series had two (see Figure 2h).
- While the CB-Series presented two boards containing compute, power, and safety logic (Figures 2c and 2f, respectively), the e-Series presents only one single PCB named as "SAFETYCONTROLBOARD" and depicted in Figures 3f, 3g, 3h and 3i.
- Figure 3g shows that the new PCB includes a Xilinx Artix-7 series field-programmable gate array (FPGA), widely used for implementing safety logic in a variety of automotive and control domains, and a much more reliable compute substrate for safety-related tasks than a MCU.
- Figure 3j shows that the base filter PCB—which helps interface power and RS485 communications from the controller (e-Series) to the robot arm mechanics—is similar to the one present in the CB-series. We also note that, while the arm mechanics connector changed in the e-Series (see Figure 3k), power and communications lines remain coherent (through the base filter board).
- For the most part, the electronics contained in the arm mechanics (Figure 3l) do not present relevant changes from an interoperability perspective. This facilitates re-purposing and reusing them (see Section 3).

Looking at the results of our teardown, we highlight the following observations.

Observation 2 *e-Series controllers from Universal Robots include a Xilinx Artix-7 series FPGA, widely used for implementing safety logic in a variety of automotive and control domains, as noted by Hallett, Corradi, and McNeil (2015); Gracic, Hayek, and Börcsök (2016, 2017), a much more reasonable choice from a user's safety perspective.*

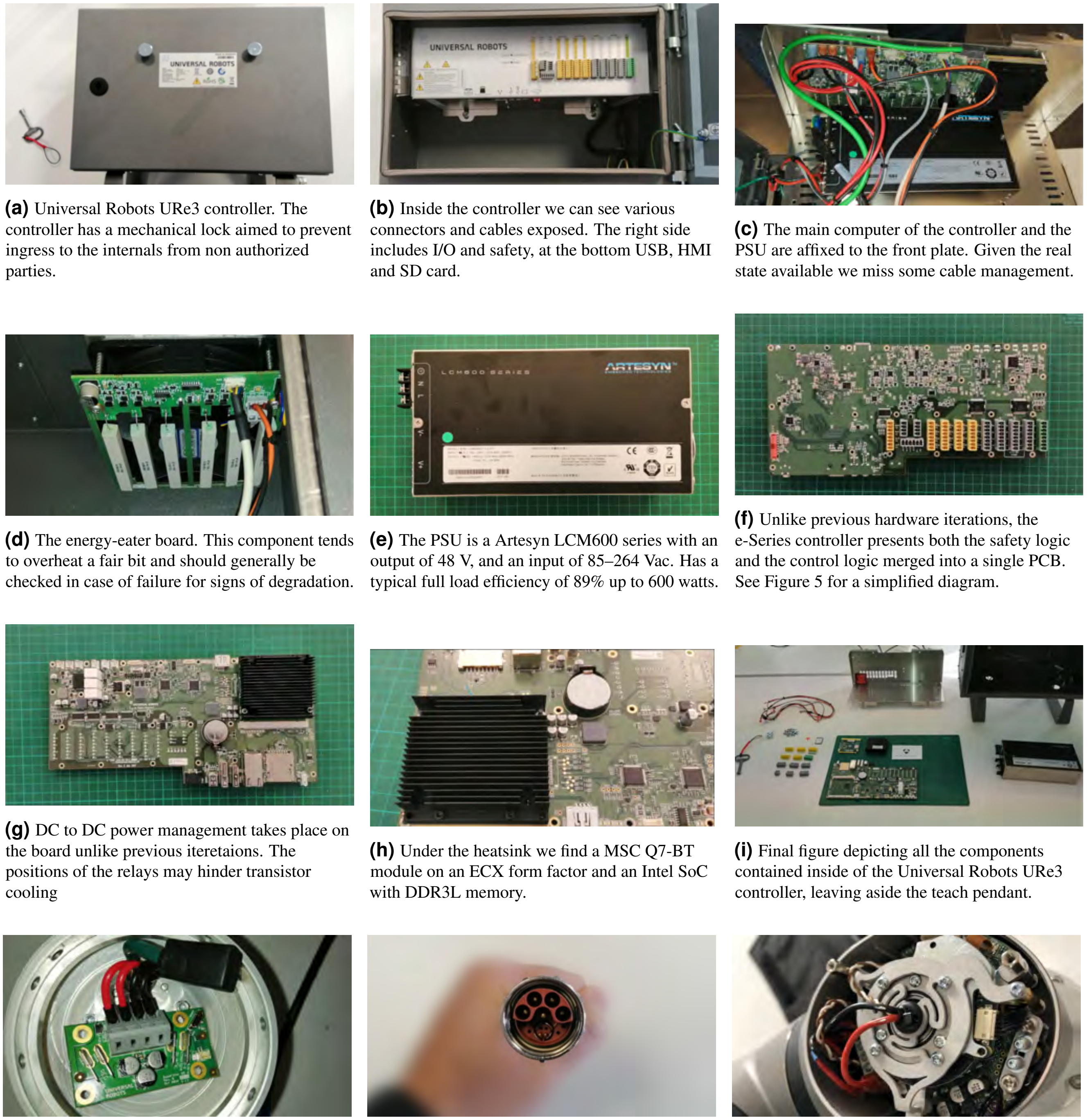
Observation 3 *While adopting different physical connectors, power and communication (RS485) lines remain coherent between CB-Series and e-Series. From the context of repairability, changing physical connectors is a clear planned obsolescense action.*

2.3 Case Study 3: Teardown of a mobile industrial robot

Figure 4 depicts the teardown process of a MiR-100, a popular mobile robot manufactured by the Danish Mobile Industrial Robots (MiR), also owned by the US Teradyne. The teardown took four hours and 20 minutes approximately. The first impression is that various components of the robot could be improved from a safety perspective, as highlighted in Figure 4g or 4i). Moreover, the teardown helped understand how this robot presents multiple (internal and external) networks and how each one of the sensors and actuators are connected across these networks, forming the data layer graph. One interesting finding resulting from the teardown is obtaining a better understanding of the robot's computational graph (the behavior itself). The robot itself is powered by Robot Operating System (ROS) (Quigley et al. (2009)) and gaining further understanding of the ROS computational graph requires understanding also its underlying hardware mapping (from which one derives the data layer graph). The teardown exercise yields exactly this, and allows to produce a data layer graph represented in the form of a hardware schematic which can then be used in combination with the computational graph to gain further understanding of the robot.

Research finding 1 *Teardown processes help determine the underlying networking architecture in a robot, from which the robot data layer graph can be inferred. Mapping the data layer graph to the computational graph (the robot behavior) is fundamental to gain better understanding of the robot and propose an appropriate security architecture.*

[3] Note that a) the robot is similar to the one studied in case study 1 and b) repeated teardowns of manipulators helped us optimize the time.



(a) Universal Robots URe3 controller. The controller has a mechanical lock aimed to prevent ingress to the internals from non authorized parties.

(b) Inside the controller we can see various connectors and cables exposed. The right side includes I/O and safety, at the bottom USB, HMI and SD card.

(c) The main computer of the controller and the PSU are affixed to the front plate. Given the real state available we miss some cable management.

(d) The energy-eater board. This component tends to overheat a fair bit and should generally be checked in case of failure for signs of degradation.

(e) The PSU is a Artesyn LCM600 series with an output of 48 V, and an input of 85–264 Vac. Has a typical full load efficiency of 89% up to 600 watts.

(f) Unlike previous hardware iterations, the e-Series controller presents both the safety logic and the control logic merged into a single PCB. See Figure 5 for a simplified diagram.

(g) DC to DC power management takes place on the board unlike previous iteretaions. The positions of the relays may hinder transistor cooling

(h) Under the heatsink we find a MSC Q7-BT module on an ECX form factor and an Intel SoC with DDR3L memory.

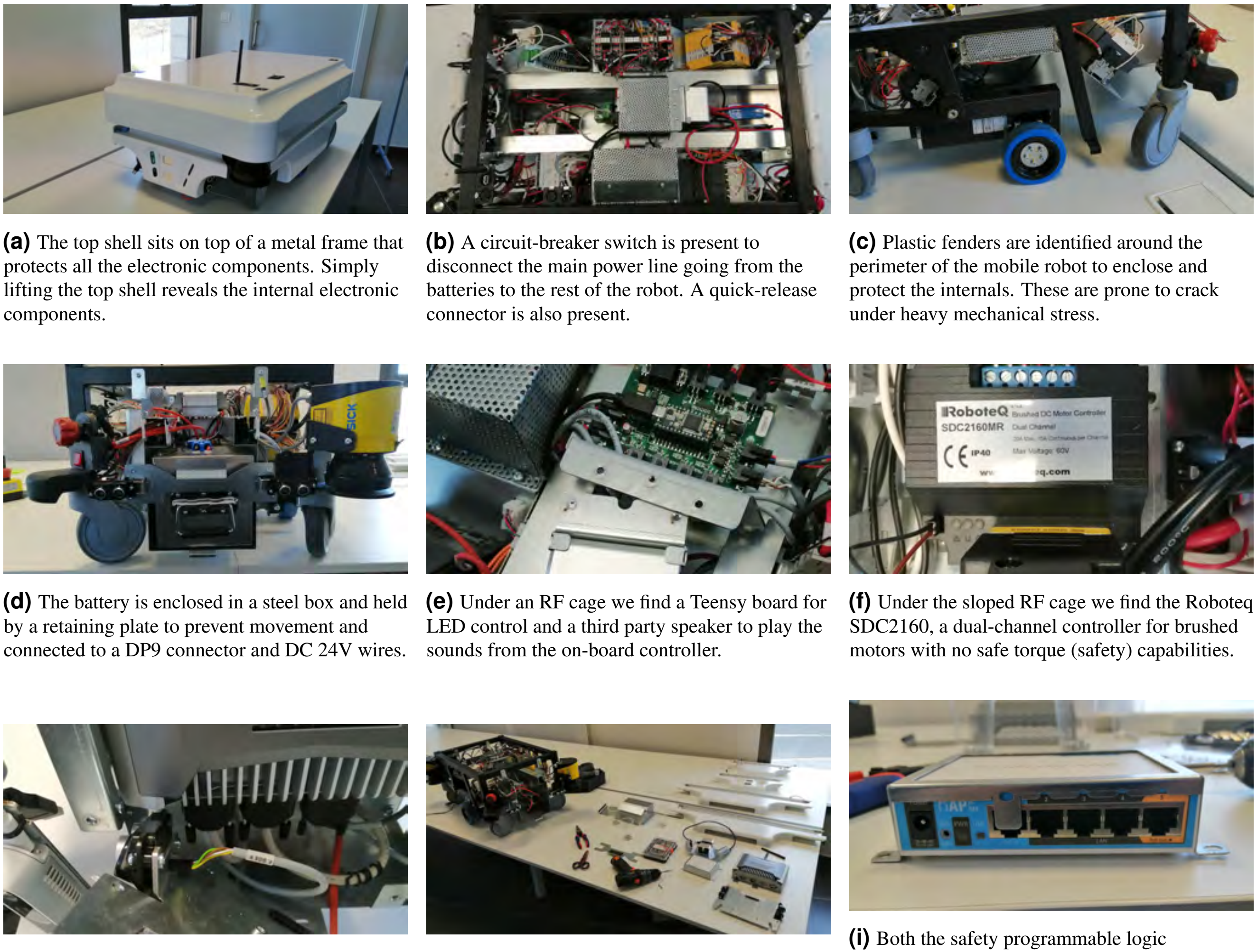
(i) Final figure depicting all the components contained inside of the Universal Robots URe3 controller, leaving aside the teach pendant.

(j) Opening the base joint of the manipulator we find the connections coming from the controller the 48V leads and the micro-USB for data.

(k) Even though the pin-out is quite similar to previous iterations of the CB3 controller, the connector itself has a different shape

(l) Opening a joint we find the harmonic drive, on top of the PCB for communication and power distribution. And the release solenoid.

Figure 3. UR3e collaborative robot teardown



(a) The top shell sits on top of a metal frame that protects all the electronic components. Simply lifting the top shell reveals the internal electronic components.

(b) A circuit-breaker switch is present to disconnect the main power line going from the batteries to the rest of the robot. A quick-release connector is also present.

(c) Plastic fenders are identified around the perimeter of the mobile robot to enclose and protect the internals. These are prone to crack under heavy mechanical stress.

(d) The battery is enclosed in a steel box and held by a retaining plate to prevent movement and connected to a DP9 connector and DC 24V wires.

(e) Under an RF cage we find a Teensy board for LED control and a third party speaker to play the sounds from the on-board controller.

(f) Under the sloped RF cage we find the Roboteq SDC2160, a dual-channel controller for brushed motors with no safe torque (safety) capabilities.

(g) The on-board controller is the embedded and “ruggedized” EC70A-SU from DFI which features an Intel processor and a Ubuntu 16.04 file system.

(h) Final figure depicting the main components contained inside of the MiR100 robot, alongside the tools used for the teardown.

(i) Both the safety programmable logic controller (PLC) and the on-board controller are connected to a Mikrotik hAP ac. A 2.4/5GHz dual-band omnidirectional access point.

Figure 4. MiR-100 mobile industrial robot teardown.

3. Teardown-enabled security research

The previous section highlighted how teardown helped identify quality and safety issues in robots, as well as obtaining a better understanding of their architectures by matching each robot's data layer graph with their corresponding computational graph. Beyond this, we argue that robot teardown is also key for security research in robotics. Recall that safety and cybersecurity are very related and influence one another, as noted by Mayoral-Vilches, Pinzger, Rass, Dieber, and Gil-Uriarte (2020); Alzola Kirschgens et al. (2018); Mayoral-Vilches, Juan, Carbajo, et al. (2019).

Teardown, as a process, is an essential part of a hardware reverse engineering task, and brings useful lessons and insights for the design of current and future robot systems. Generally, teardown supports Kerckhoffs' principle in revealing all the details and weaknesses of a security system, excluding volatile secrets such as keys or credentials that are stored in memory and most likely disappear naturally once the power supply is taken away (with the exception of keys stored in permanent memory, which is generally discouraged, and would be discovered along the teardown process). Overall, the history of proprietary systems violating Kerckhoffs' principle by pursuing "security through obscurity" is rich of failure cases (with the military domain as the sole exception), as a vast amount of related work demonstrates.

Reverse engineering has always been invaluable to discover vulnerabilities and develop remedies in many domains: network security (Guha and Mukherjee (1997)), access control (Wang, Wang, Zhang, and Li (2008)), embedded systems (McLoughlin (2008); Rajendran, Sam, Sinanoglu, and Karri (2013)), software engineering (Lin, Zhang, and Xu (2010); Treude, Filho, Storey, and Salois (2011)), or the internet of things (Tellez, El-Tawab, and Heydari (2016)). By promoting systematic teardown we want to extend this successful concept to the analysis of abandoned robots.

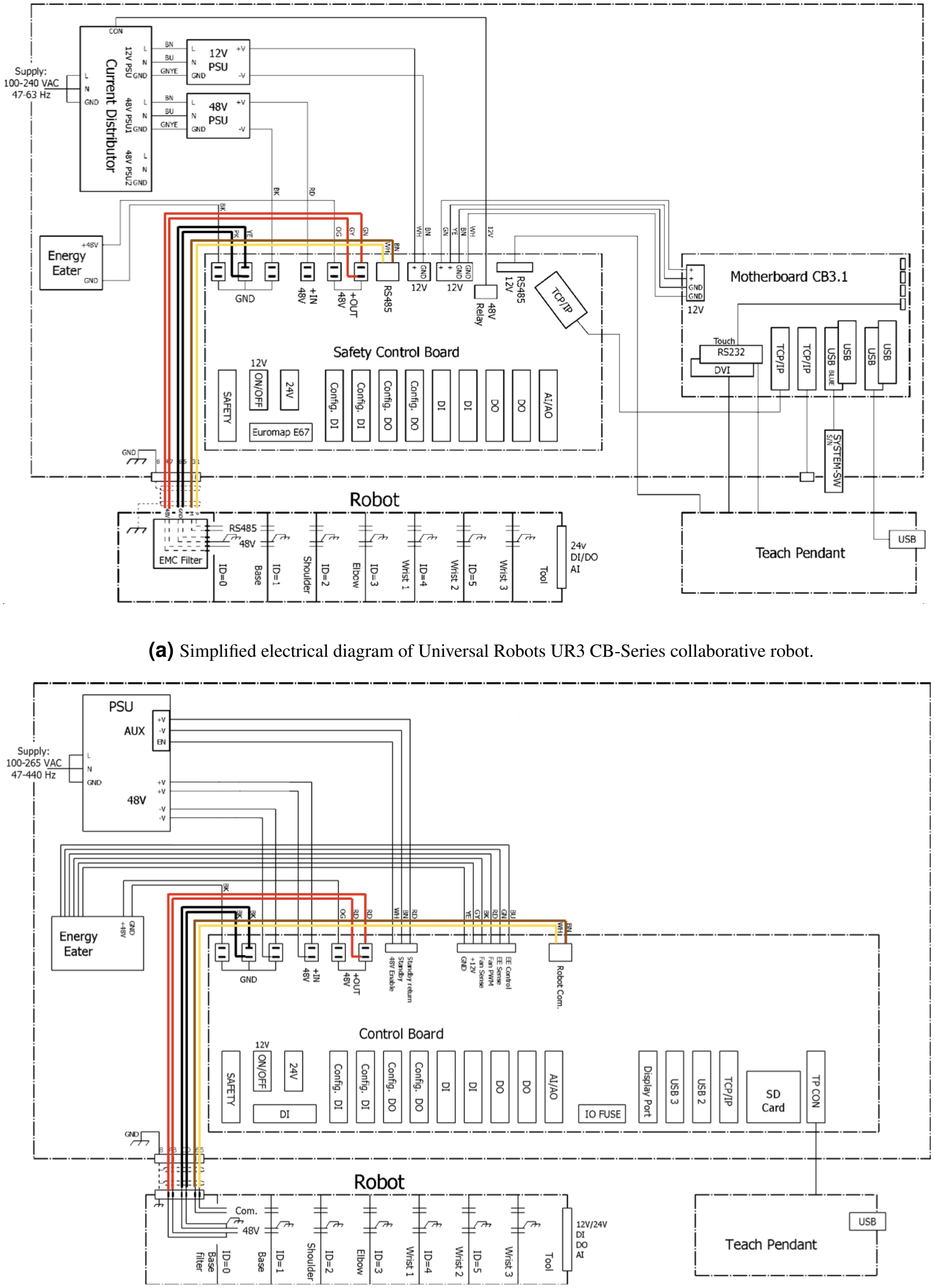
Particularly, and as part of this research, our group identified more than 100 security flaws across the three robots described above over a period of two years. Most of the flaws were cataloged as vulnerabilities and 17 obtained new Common Vulnerabilities and Exposures (CVE) IDs all of which are publicly disclosed at the Robot Vulnerability Database (RVD), proposed by Mayoral-Vilches, Juan, Dieber, et al. (2019). Table 1 introduces some of the selected security vulnerabilities found. The information obtained through teardown helps pinpoint flaws across the multiple (internal and external) robot networks. In most cases, these robots present few or no security measures, allowing adversaries to easily exploit the flaws of internal components (e.g., RVD#2558, RVD#2561 or RVD#2562), so compromising the robot behavior or taking full control of it.

We advocate for robot teardowns as a means to improve security in robotics and encourage manufacturers, integrators and end-users to carefully consider the underlying hardware architecture to protect their robotic systems. Similarly, we encourage teardowns as a tool to mitigate outstanding security flaws. Proper knowledge of the hardware helps determine which additional elements can help mitigate security issues when the manufacturer does not react. As an example, our group introduced an additional commercial off-the-shelf hardware firewall within MiR's internal network, between the main controller and the safety PLC, by SICK AG, mitigating RVD#2558 without having to modify any parts of the firmware. This modification could enable users and system integrators frustrated with MiR's security policies to secure their robots directly.

CVE ID	RVD ID	Description	Report
CVE-2019-19626	RVD#1408	Bash scripts (magic UR files) get launched automatically with root privileges and without validation or sanitizing	`https://github.com/aliasrobotics/RVD/issues/1408`
CVE-2020-10290	RVD#1495	Universal Robots URCaps execute with unbounded privileges	`https://github.com/aliasrobotics/RVD/issues/1495`
CVE-2020-10267	RVD#1489	Unprotected intelectual property in Universal Robots controller CB 3.1 across firmware versions	`https://github.com/aliasrobotics/RVD/issues/1489`
CVE-2020-10266	RVD#1487	No integrity checks on UR+ platform artifacts when installed in the robot	`https://github.com/aliasrobotics/RVD/issues/1487`
CVE-2020-10265	RVD#1443	UR dashboard server enables unauthenticated remote control of core robot functions	`https://github.com/aliasrobotics/RVD/issues/1443`
CVE-2020-10264	RVD#1444	RTDE Interface allows unauthenticated reading of robot data and unauthenticated writing of registers and outputs	`https://github.com/aliasrobotics/RVD/issues/1444`
CVE-2020-10278	RVD#2561	Unprotected BIOS allows user to boot from live OS image	`https://github.com/aliasrobotics/RVD/issues/2561`
CVE-2020-10270	RVD#2557	Hardcoded Credentials on MiRX00 Control Dashboard	`https://github.com/aliasrobotics/RVD/issues/2557`
CVE-2020-10279	RVD#2569	Insecure operating system defaults in MiR robots	`https://github.com/aliasrobotics/RVD/issues/2569`
CVE-2020-10276	RVD#2558	Default credentials on SICK PLC allows disabling safety features	`https://github.com/aliasrobotics/RVD/issues/2558`
CVE-2020-10273	RVD#2560	Unprotected intellectual property in Mobile Industrial Robots (MiR) controllers	`https://github.com/aliasrobotics/RVD/issues/2560`
CVE-2020-10277	RVD#2562	Booting from a live image leads to exfiltration of sensible information and privilege escalation	`https://github.com/aliasrobotics/RVD/issues/2562`
CVE-2020-10269	RVD#2566	Hardcoded Credentials on MiRX00 wireless Access Point	`https://github.com/aliasrobotics/RVD/issues/2566`
CVE-2020-10275	RVD#2565	Weak token generation for the REST API	`https://github.com/aliasrobotics/RVD/issues/2565`
CVE-2020-10274	RVD#2556	MiR REST API allows for data exfiltration by unauthorized attackers (e.g. indoor maps)	`https://github.com/aliasrobotics/RVD/issues/2556`
CVE-2020-10271	RVD#2555	MiR ROS computational graph is exposed to all network interfaces, including poorly secured wireless networks and open wired ones	`https://github.com/aliasrobotics/RVD/issues/2555`
CVE-2020-10272	RVD#2554	MiR ROS computational graph presents no authentication mechanisms	`https://github.com/aliasrobotics/RVD/issues/2554`

Table 1. The 17 novel (new CVE IDs) vulnerabilities encountered during a period of two years in the robots of Teradyne and as a result of an initial hardware teardown. All security issues were responsibly disclosed. For a full list of the more than 100 security flaws, we kindly refer readers to the Robot Vulnerability Database Mayoral-Vilches, Juan, Dieber, et al. (2019).

Research finding 2 *Teardown helps pinpoint security flaws across the multiple internal and external robot networks.*



(a) Simplified electrical diagram of Universal Robots UR3 CB-Series collaborative robot.

(b) Simplified electrical diagram of Universal Robots UR3e e-Series collaborative robot.

Figure 5. Simplified electrical diagrams of Universal Robots UR3 CB-Series (5a) and UR3e e-Series (5b) collaborative robots. Matching the results of our teardown (depicted in Figures 3j and 3l), we highlight in red the high voltage power lines, in black the GND, and use brown and yellow (instead of *white*) for the RS485 communication bus lines.

4. Lessons learned

Through this work we learned about the underlying hardware architectures that popular collaborative robots deploy today, researching security in the process. As a result of the teardown case studies described in Section 2, our group also identified several of the planned obsolescence indicators previously introduced in Section 1. Planned obsolescence was particularly evident in the robots from Universal Robots. To further illustrate this, Figures 5a and 5b depict the simplified electrical diagrams of the UR3 and UR3e robots. From an electrical point of view, these two robots present a similar layout for interfacing with the robot arm.

While we appreciate certain changes in the electronics, given the teardown results, we find no real reason why backwards or forward compatibility between controllers and robotic arms should not be possible. This would mean that existing customers with UR3 robots could repair and replace parts in either the controller or the robotic arm, without being forced to pay the premium price of buying a complete new set including both.

Unsurprisingly, we observe that the manufacturer introduced subtle changes meant to make this particular intent harder. One of such actions is depicted in Figure 3k, which shows the replacement of the controller-to-arm connector, which we can only justify with attempts to exercise obsolescence practices. Another of such actions includes the obscurity around the changes introduced in the UR3e robot arm itself. These changes can be summarized with the addition of an extra 6-axis force-torque sensor at the end of the robot. The *exact* same result can be achieved in UR3 robot arms by adding commercial off-the-shelf robot components, gaining such capabilities.

5. Conclusions

In this article we presented robot teardowns as an approach to study robot hardware architectures, obtain repairing capabilities and research its security. We discuss the empirical results of three robot teardowns and the findings affecting quality and safety throughout the process. We then discuss how teardown is a relevant tool for security research in robotics which helps pinpoint security flaws early across the multiple internal and external networks in a robot. Moreover, we introduce our security findings and propose mitigations powered by the hardware know-how and repairing capabilities acquired. Ultimately, we research planned obsolescence practices in the robots from Teradyne and propose actions that could be taken to bypass obsolescence.

Our results show evidence that robot teardowns can help the robotics industry and supply chain by improving significantly quality, safety and security. Our findings extrapolate to most of the robots manufactured by Teradyne and its subsidiaries. We show concern for the currently growing trend in robotics to create private networks of certified groups, a common practice shown by manufacturers like MiR or UR, both owned by Teradyne. This difficulties system integration, repairability and ultimately security. We advocate for a '*Right to Repair*' in robotics and encourage end-users to reflect their needs into their supply chains and into the original upstream robot manufacturers.

Declaration of Conflicting Interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Alzola Kirschgens, L., Zamalloa Ugarte, I., Gil Uriarte, E., Muñiz Rosas, A., & Mayoral-Vilches, V. (2018, June). Robot hazards: from safety to security. *ArXiv e-prints*.

Cerrudo, C., & Apa, L. (2017a). *Hacking robots before skynet* (Tech. Rep.). Retrieved from `https://ioactive.com/wp-content/uploads/2018/05/Hacking-Robots-Before-Skynet-Paper_Final.pdf`

Cerrudo, C., & Apa, L. (2017b). *Hacking robots before skynet: Technical appendix* (Tech. Rep.). Retrieved from `https://ioactive.com/pdfs/Hacking-Robots-Before-Skynet-Technical-Appendix.pdf`

Cordella, M., Alfieri, F., & Sanfelix, J. (2019). *Analysis and development of a scoring system for repair and upgrade of products-final report.* Publications Office of the European Union Luxembourg.

Crowe, S. (2021). *Teradyne's robotics portfolio grows revenue 33% in q1.* Retrieved 2021-05-02, from `https://www.therobotreport.com/teradyne-robotics-portfolio-revenue-33-q1/`

for Communication (European Commission), D.-G. (2020). *Circular economy action plan, for a cleaner and more competitive europe.* Publications Office of the European Union Luxembourg. doi: 10.2779/05068

Gracic, E., Hayek, A., & Börcsök, J. (2016). Implementation of a fault-tolerant system using safety-related xilinx tools conforming to the standard iec 61508. In *2016 international conference on system reliability and science (icsrs)* (pp. 78–83).

Gracic, E., Hayek, A., & Börcsök, J. (2017). Evaluation of fpga design tools for safety systems with on-chip redundancy referring to the standard iec 61508. In *2017 2nd international conference on system reliability and safety (icsrs)* (pp. 386–390).

Guha, B., & Mukherjee, B. (1997, July). Network security via reverse engineering of TCP code: vulnerability analysis and proposed solutions. *IEEE Network*, *11*(4), 40–48. (Conference Name: IEEE Network) doi: 10.1109/65.598458

Hallett, E., Corradi, G., & McNeil, S. (2015). Xilinx reduces risk and increases efficiency for iec61508 and iso26262 certified safety applications. *Xilinx White Paper*.

Hatta, M. (2020). The right to repair, the right to tinker, and the right to innovate. *Annals of Business Administrative Science*, 0200604a.

Kohlweiss, A., Auberger, E., Ketenci, A., & Ramsauer, C. (2020). Integration of a teardown approach at graz university of technology´ s lead factory. *Procedia Manufacturing*, *45*, 240–245.

Lin, Z., Zhang, X., & Xu, D. (2010, March). Automatic reverse engineering of data structures from binary execution. In *Proceedings of the 11th Annual Information Security Symposium* (p. 1). West Lafayette, IN: CERIAS - Purdue University.

Mayoral-Vilches, V., Alzola Kirschgens, L., Bilbao Calvo, A., Hernández Cordero, A., Izquierdo Pisón, R., Mayoral Vilches, D., … Peter, A. (2018, June). Introducing the robot security framework (rsf), a standardized methodology to perform security assessments in robotics. *ArXiv e-prints*.

Mayoral-Vilches, V., García-Maestro, N., Towers, M., & Gil-Uriarte, E. (2020). Devsecops in robotics. *arXiv preprint arXiv:2003.10402*.

Mayoral-Vilches, V., Hernández, A., Kojcev, R., Muguruza, I., Zamalloa, I., Bilbao, A., & Usategi, L. (2017). The shift in the robotics paradigm—the hardware robot operating system (h-ros); an infrastructure to create interoperable robot components. In *Adaptive hardware and systems (ahs), 2017 nasa/esa conference on* (pp. 229–236).

Mayoral-Vilches, V., Juan, L. U. S., Carbajo, U. A., Campo, R., de Cámara, X. S., Urzelai, O., … Gil-Uriarte, E. (2019). Industrial robot ransomware: Akerbeltz. *arXiv preprint arXiv:1912.07714*.

Mayoral-Vilches, V., Juan, L. U. S., Dieber, B., Carbajo, U. A., & Gil-Uriarte, E. (2019). Introducing the robot vulnerability database (rvd). *arXiv preprint arXiv:1912.11299*.

Mayoral-Vilches, V., Mendia, G. O., Baskaran, X. P., Cordero, A. H., Juan, L. U. S., Gil-Uriarte, E., … Kirschgens, L. A. (2018). aztarna, a footprinting tool for robots. *arXiv preprint arXiv:1812.09490*.

Mayoral-Vilches, V., Pinzger, M., Rass, S., Dieber, B., & Gil-Uriarte, E. (2020). Can ros be used securely in industry? red teaming ros-industrial. *arXiv preprint arXiv:2009.08211*.

McLoughlin, I. (2008, December). Secure Embedded Systems: The Threat of Reverse Engineering. In *2008 14th IEEE International Conference on Parallel and Distributed Systems* (pp. 729–736). (ISSN: 1521-9097) doi: 10.1109/ICPADS.2008.126

Pogliani, M., Maggi, F., Balduzzi, M., Quarta, D., & Zanero, S. (2020, October). Detecting Insecure Code Patterns in Industrial Robot Programs. In *Proceedings of the 15th acm asia conference on computer and communications security (asia ccs '20).* Taipei, TW.

Quarta, D., Pogliani, M., Polino, M., Maggi, F., Zanchettin, A. M., & Zanero, S. (2017, May). An Experimental Security Analysis of an Industrial Robot Controller. In *Proceedings of the 38th ieee symposium on security and privacy.* San Jose, CA.

Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., … Ng, A. (2009, May). Ros: an open-source robot operating system. In *Proc. of the ieee intl. conf. on robotics and automation (icra) workshop on open source robotics.* Kobe, Japan.

Rajendran, J., Sam, M., Sinanoglu, O., & Karri, R. (2013, November). Security analysis of integrated circuit camouflaging. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (pp. 709–720). New York, NY, USA: Association for Computing Machinery. Retrieved 2021-05-10, from `https://doi.org/10.1145/2508859.2516656` doi: 10.1145/2508859.2516656

Sandborn, P., Myers, J., Barron, T., & McCarthy, M. (2006). Using teardown analysis as a vehicle to teach electronic systems manufacturing cost modeling. In *Proceedings of the international electronics packaging education conference (at the ectc).*

Skorobogatov, S. (2017). Deep dip teardown of tubeless insulin pump. *arXiv preprint arXiv:1709.06026*.

Taurer, S., Breiling, B., Svrta, S., & Dieber, B. (n.d.). Case study: remote attack to disable mir100 safety..

Tellez, M., El-Tawab, S., & Heydari, M. H. (2016, December). IoT security attacks using reverse engineering methods on WSN applications. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)* (pp. 182–187). doi: 10.1109/WF-IoT.2016.7845429

Treude, C., Filho, F. F., Storey, M.-A., & Salois, M. (2011, October). An Exploratory Study of Software Reverse Engineering in a Security Context. In *2011 18th Working Conference on Reverse Engineering* (pp. 184–188). (ISSN: 2375-5369) doi: 10.1109/WCRE.2011.30

Tutunji, T. (n.d.). Reverse engineering: Electronics.

Wang, R., Wang, X., Zhang, K., & Li, Z. (2008, October). Towards automatic reverse engineering of software security configurations. In *Proceedings of the 15th ACM conference on Computer and communications security* (pp. 245–256). New York, NY, USA: Association for Computing Machinery. Retrieved 2021-05-10, from `https://doi.org/10.1145/1455770.1455802` doi: 10.1145/1455770.1455802

Younis, M. B., & Tutunji, T. A. (2010). Reverse engineering in mechatronics education. In *7th international symposium on mechatronics and its applications* (pp. 1–5).

Zhu, Q., Rass, S., Dieber, B., & Vilches, V. M. (2021). Cybersecurity in robotics: Challenges, quantitative modeling, and practice. *arXiv preprint arXiv:2103.05789*.