A Security Analysis of CNC Machines in Industry 4.0

Marco Balduzzi[1], Francesco Sortino[2], Fabio Castello[2], and Leandro Pierguidi[2]

[1] Trend Micro Inc, 225 East John Carpenter Freeway, Irving, Texas (USA)
[2] Celada SpA, via Cesare Battisti 156, Cologno Monzese, Milan (Italy)

Abstract. Computer numerical control (CNC) machines are extensively used in production plants and are considered a crucial asset for organizations worldwide. These machines require unique controllers that differ from those used in other types of machine tools in terms of software architecture, protocols, and design, so to meet the high precision and accuracy demands of their applications. The growing adoption of network-enabled systems in the industrial domain, driven by Industry 4.0, has resulted in an increased use of CNC machines. These machines have evolved from traditional mechanical machines to full-fledged systems with multiple networking services for smart connectivity. This study investigates the risks associated with this technological development. Using actual machine installations, we conducted the first empirical evaluation of the privacy and security implications of Industry 4.0 in the CNC domain. Our findings revealed that malicious users could conduct five types of attacks: compromise, denial-of-service, damage, hijacking, and theft. We reported our findings to the affected vendors and proposed mitigations to manufacturers, integrators and end-users. Our work aims to provide an opportunity to increase awareness in a domain where security does not appear to be a priority at present.

1 Introduction

In the past decade, there has been a significant rise in the popularity of network-enabled systems, even for devices that were historically not designed to offer such capabilities. This trend has been particularly evident in the industrial domain, where various types of network-enabled systems are widely used to support modern manufacturing processes.

The development of devices such as industrial gateways, computer numerical controls (CNCs), industrial robots, and autonomous vehicles for logistics has led to new industrial models that follow the general paradigm of Industry 4.0, driving manufacturing companies towards networked shop floors.While connecting modern machine tools to wide networks, including the Internet, presents an important opportunity for creating new business intelligence through the collection and analysis of production data, it also poses potential threats to the security and privacy of organizations.

CNC machines play a fundamental role in the manufacturing industry because they are the building blocks of the mechanical processing of pieces. In a manufacturing line, a variety of systems cooperate, such as robots or other support systems (like control servers), but CNCs are responsible for the mechanical processing of the pieces through drillers, lathes, or cutters. Industrial robots, on the other hand, are used for auxiliary operations such as material handling, palletizing, or as soldering stations.

CNC machines require unique systems that differ from other machine tools, not only in terms of software architecture and protocols but also in their overall design, to meet the specific demands of precision and accuracy required by their applications. While under the hood, CNC machines still rely on well-established mechanical automation routines, they are also equipped with unique solutions specific to their domain, such as advanced software algorithms and specialized hardware components. These domain-specific functionalities set modern CNCs apart from traditional machine tools and enable them to achieve higher levels efficiency in manufacturing processes.

For this reason, we believe that CNC machines are a key element in analyzing the security posture of the manufacturing ecosystem. As far as we know, we are the first to conduct a comprehensive analysis of the security issues related to this specific technology and demonstrate potential vulnerabilities in practice.

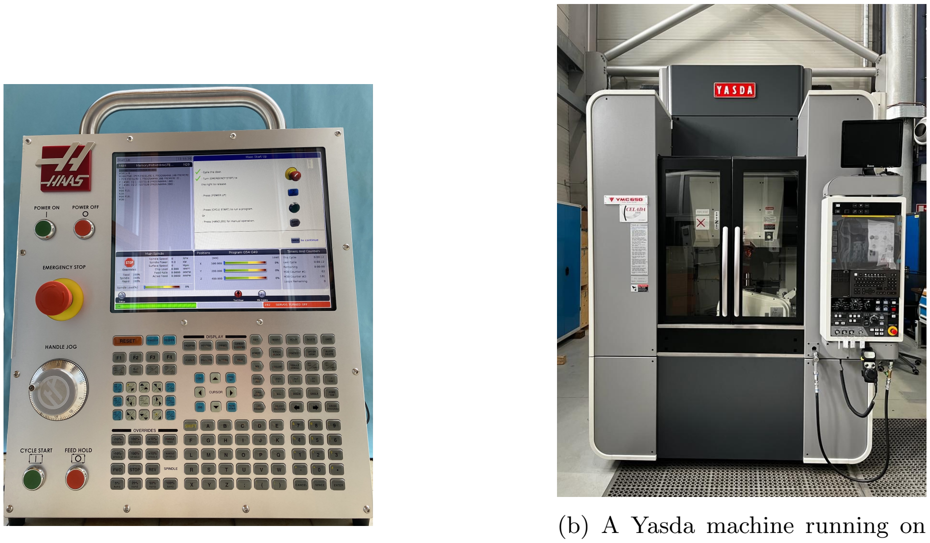
In short, the contributions of our work consist of the following:

- We investigate the security and privacy of CNC machines in Industry 4.0. To the best of our knowledge, we are the first to conduct a depth empirical analysis in this direction.
- We conduct an extensive security assessment of the technologies offered by modern CNCs by making use of the controllers provided by four large representative vendors.
- We perform threat modelling and report problems resulting in five attack classes: compromise, damage, denial-of-service, hijacking and theft.
- We communicate our findings to the affected vendors, we propose mitigations, and do our best to raise awareness in this domain.

2 Background

A CNC machine is a machine tool developed to transform the geometry of raw material through machining, a process through which a material (be it metal, polymer, or otherwise) is cut until it reaches the desired geometry through a *controlled process.* This process is carried out through the aid of cutting tools and is achieved using a controller, which, together with the mechanics of the machinery, constitutes a numerically controlled machine tool. The main benefit of this addition lies in the possibility of the machine to operate process phases in an unattended way and to use the computing power of the controller to create complex geometries with high degrees of precision.

CNC machines are programmed in G-code (RS-274[1]). This language resembles Basic programming: it is presented as a series of instructions initialized by a letter address, which follow one another on successive lines separated by paragraph breaks; each of these lines is called block. Each letter address specifies



(a) A Haas controller simulator.

(b) A Yasda machine running on Fanuc controller.

Fig. 1: Examples of simulator and machine.

the type of movement or function called by the user in that part of the program. Over the years, concepts that we now consider basic in programming languages, such as loops, macros, and object programming, have also been introduced in machine language, and numerous examples of conversational or guided languages have been included to facilitate CNC operations.

While G-code is, still, the standard for programming CNC machines, engineers nowadays tend to rely on CAM software[3] to translate architectural drawings (of the parts to be produced) into software programs. Such programs are then ran on controller simulators before being deployed in production lines. Figure 1a shows one of these simulators, which can be either physical (like in the photo) or software (e.g., in form of virtual machine). Despite this difference, controller simulators implement the same logic of a real-world CNC machine (ref. Figure 1b) – in fact, the software running on such simulators is normally the same as the one on the machine, despite the hardware peripherals being virtualized e.g. the motors used to move the machine's axes.

3 Approach

The manufacturing and deployment of a CNC machine can be modeled as a supply chain process, where a controller manufacturer produces and sells controllers to multiple machine manufacturers. The machine manufacturers, using

[3] Computer Aided Manufacturing

the controllers, develop CNC machines such as lathes, and make them available to resellers, integrators, and end-users.

There are two considerations to make: firstly, any security issues or vulnerabilities introduced by the controller manufacturer at the beginning of the supply chain will be propagated throughout the entire chain, along with any technologies or software used by the controller. Therefore, by examining the controller, we can gain a wider perspective on the adoption of such technologies and any related issues throughout the supply chain.

Secondly, the number of controller manufacturers on the market is much smaller than the number of machine manufacturers, with a single controller typically being used to build dozens of CNC machines. This is important for our goal of evaluating the security of CNCs, as it means we can focus on a smaller number of manufacturers that represent a significant portion of the market.

Our investigation begins by identifying a set of representative, large controller manufacturers on the market. We proceed by selecting those players that have a worldwide reach, are on the market since tens of years, are widely known in the industrial domain, or have developed technologies widely used in this industry. All selected manufacturers develop controllers used on machines we have access to[4]. This is important for us because we want to conduct an empirical study, showing that our concerns have practical implications. Table 1 provides a summary of the selected manufacturers and their respective controllers and machines that we used for testing.

Our analysis consists of the following process:

- We conduct threat modelling, by presenting the scenarios in which a miscreant would be able to target a CNC machine and discussing the impact of such attacks.
- We identify the technologies introduced in the CNC realm to adhere to Industry 4.0. They encompass protocols and services used to connect the machines to smart environments, for example to share the production information with centralized systems for better management and cost reduction. They also enable remote management, for example, for an operator to change the executed program or configure the tooling.
- We conduct a first coarse-grained security assessment, for example using vulnerability scanners to identify potential known vulnerabilities or misconfigurations in such services. Note that the focus of our research is on domain-specific technologies, i.e. we ignore those problems related to generic software (like Windows services).
- We then go deep into the CNC technologies previously identified, by analyzing the risks of abuses and conducting practical attacks on the controllers. For this, we develop attack tools that leverage the weaknesses that we identified. We make use of both proprietary documentation and APIs we were given access to.

[4] The machines are located in different facilities: in Celada, MADE Competence Center, or the Department of Mechanical Engineering of the Polytechnic University of Milan.

Vendor	**Haas**	**Okuma**	**Heidenhain**	**Fanuc**
Country	US	Japan	Germany	Japan
Year of establishment	1983	1898	1889	1972
Estimated size	More than US$1B revenue and 1,300 employees (2018)	US$1.41B revenue and 3,812 employees (2020)	US$1.3B revenue and 8,600 employees (2020)	US$4.18B revenue and 8,260 employees (2020)
Market	Controllers and machines for all markets	Controllers and machines for all markets	Controllers	Controllers and simple machines
Simulator	100.19.100.1123	OSP-P300S	TNC 640 Programming Station 340595 V.10.00.04	Not used
Controllers	100.20.000.1110	P300MA-H	TNC 640	31iB5 iHMI and 32i-B
Machines	Super Mini Mill	Genos M460V-5AX6	Hartford 5A-65E	Yasda YMC 430+RT10 and Star SR-32JII
Types	3-axis vertical machining center	5-axis vertical machining center	5-axis vertical machining center	5-axis vertical micro machining center and Swiss lathe

Table 1: A summary of the selected manufacturers and their respective controllers and machines used for testing.

- We collect evidence of our concerns and collaborate with the affected vendors in suggesting mitigations.

3.1 Threat Modelling

CNC machines are commonly installed in manufacturing networks. These networks, often referred as OT networks, are standalone networks that traditionally were not in communication with corporate (IT) networks. However, in modern factory plants, CNC machines communicate with external servers for enabling remote machine programming or process monitoring. These machines are, for example located in corporate networks reachable via industrial gateways or mobile networks. Mobile operators offer connectivity to CNC machines via Internet while industrial gateways act as bridges between OT and IT networks. To confirm these trends, in the preliminary phase of our research, we conducted an interview with experts on the fields (e.g. suppliers and installers of machines) who confirmed these claims.

We model the attacker as following:

- A remote attacker who has access to the OT network. This attacker could be an insider with direct access to the OT network where the CNC machine is installed, or an attacker with a presence in an enterprise with missing or wrongly configured network segmentation that exposes the CNC machine.
- A remote attacker with access to the IT network. The attacker gains access to the CNC machine by pivoting from the IT network, potentially exploiting misconfigurations or vulnerabilities in the industrial gateway connecting the IT and OT networks. Previous research has shown that such devices are

vulnerable to several types of attacks [2]. Alternatively, the attacker could pivot from the server that communicates with the CNC machine.
- An Internet-based attacker. In this scenario, the attacker conducts the attack from the Internet. Unfortunately, CNC machines are sometimes left exposed to the Internet for remote monitoring or due to misconfiguration. We conducted an analysis of this type using a large-scale scanner (ZMap) and found evidence of exposed machines. However, we did not connect to these machines for ethical reasons.
- A remote attacker who communicates with the machine operator. In this scenario, the attacker social-engineers the operator, for example, via email, persuading him to install a CNC add-in, as we discuss later.

In this threat model, we should also consider the possibility of an attacker with physical access to the machine. However, for the purposes of our research, we chose to focus solely on remote attackers and did not include this particular scenario.

An attacker who fits within our threat model would be capable of carrying out all five attack classes outlined in the rest of the paper.

3.2 CNC Technologies and Related Problems

All the controllers we considered provide various technologies that can integrate CNCs into modern digital shop floors. These technologies enable automatic data exchange with acquisition systems, enterprise resource planning (ERP) systems, CAM software, digital twin solutions, and tool management systems. Since these technologies are typically proprietary and designed specifically for CNCs, they require a thorough and specialized analysis to fully understand their security implications.

For example, Haas Connect[5] is a cloud service offered by Haas to monitor a machine remotely. With Haas Connect, an engineer can monitor the production information of the machine, knowing how many parts are produced over time, or being informed if any alert occurs. Many of these technologies are included by default in the controller, while others are offered on demand and need to be purchased in addition. However, we observed that most of the customers prefer purchasing machines equipped with all technologies for many reasons like the fiscal incentives offered by several countries on buying these “smart technologies” or the clear advantages in having machines that can be centrally managed and monitored. In our research, we decided to focus on those technologies that are included by-default in the installations (second column), with the addition of THINC-API for the reasons explained later. OPC-UA was not taken into consideration because rarely available.

MTConnect[6] is an effort to standardize the different protocols used in the industrial domain to collect machinery data. The goal is indeed to provide guidelines for converting old and proprietary information to a common language; this

[5] https://www.haascnc.com/productivity/control/haas-connect.html

[6] https://www.mtconnect.org

Vendor	Default Technologies	Optional Technologies
Haas	MTConnect, Haas Connect, Ethernet Q Commands	NaN
Okuma	NaN	THINC-API , MTConnect
Heidenhain	RPC and LSV2 (DNC)	OPC-UA
Fanuc	Focas	OPC-UA , MTConnect

Table 2: A summary of Industry 4.0 technologies adopted by manufacturers.

will help organizations to handle machine tools from different brands in an easier form. Along with our evaluation, we confirmed that 3 of the tested vendors support MTConnect, in particular Haas provides such feature on all default installations. In our analysis, we investigated the data that an attacker could infer (or leak) from a machine exposing MTConnect over the network. A common scenario is, for example, the number of parts that are produced, together with the associated program. In other cases, an attacker can infer the source code of the executed program by repeatedly querying the MTConnect agent installed on the machine as we show later.

Despite the standardization effort around MTConnect, proprietary protocols are confirmed to be the majority, with one of these being Haas's Ethernet Q Commands[7]. With this protocol, a user can query information from a controller (for example the machine's model, the tooling configuration, or the number of produced parts) or set (program) variables needed for a program to execute. In the following Listing, few examples are given:

```
?100: Query the Machine's Serial Number
?Q402: Query the Parts Counter #1 (number of produced parts)
?Q600 10000: Read the value of variable 10000
?E10000 123: Write the value 123 into the variable 10000
```

This service is useful in making a machine reachable remotely and enables manufacturing automation; however, it may also expose the machine to potential threats. This is, indeed, the case suggested by our analysis. In fact, even if the documentation reported that only a limited range of registers could be written, namely those ones related to program variables (i.e., 10000-10999), this was not the case. As we describe later in the paper, our experiment confirmed that such a lack of access control allows a miscreant to conduct attacks like denial-of-service, hijacking, or damage.

Heidenhain offers so-called DNC interface[8], which is implemented with two protocols: RPC and LSV2. The first is a proprietary protocol operating on TCP/19003. Heidenhain uses the generic name of RPC (remote procedure call) for a protocol allowing a remote peer to call a remote interface's method on the CNC. The second is a standardized protocol used by certain vendors. While

[7] https://www.haascnc.com/service/troubleshooting-and-how-to/how-to/machine-data-collection—ngc.html

[8] https://www.heidenhain.com/products/digital-shop-floor/connected-machining

it is not as famous as other technologies, it is used and documented to a certain extent. PyLSV2 is, for example, a Python library for implementing a LSV2 compatible client.

In our evaluation, we obtained access to the RemoTools library provided by the manufacturer to the integrators in order to develop interfaces for the controller. A miscreant having access to this library is facilitated in implementing a malicious client for hijacking the operation of the CNC machine, or stealing confidential data. Note that the same attacks could be developed with public libraries as well, for example for LSV2. The controller offers the possibility to enable network authentication on the DNC interface for both RPC and LSV2. The authentication is implemented in form of SSH tunnelling, which is very convenient because the controller runs on top of Linux. This option, which needs to be voluntarily enabled by the integrator or the end-user, is a good solution to the problems that we identified and that we discuss later in the paper.

Fanuc offers an equivalent technology called Focas[9]. Even though Focas offers a restricted set of remote-call possibilities compared with the other vendors (that is, a limited number of management features), our experiments showed that a miscreant can still conduct attacks like damage, DoS, and hijacking. This is an important issue because, unfortunately, authentication was introduced only recently (in 2020) and only as a non-default option — according to our communications with the vendor. This new version allows to configure an eight-digit code to be used as authentication token. This is achieved by setting the controller's global parameter 10344 to the desired code. By default, this value is set to 0 (no authentication).

Okuma stands out from the controller market for one interesting feature: the modularity of its controller. In fact, while the vendor offers in its simplest form a limited controller, it also provides a mechanism (called THINC-API) to highly customize its functionalities. With this technology, anyone can implement an add-in that – once installed – runs in the context of the controller, in the form of extension. Applications developed with THINC-API are commonly offered by integrators and resellers to their customers, and can be made available to 3rd-parties via the Okuma's app store[10] for easier distribution.

Given the prevalence of this technology, we conducted a dedicated assessment in the hope to better understand the potential impact of this technology despite not being provided as default option. Unfortunately, our analysis highlighted that simple security mechanisms that are nowadays very common like *resource access control* are not yet supported. As a result, if a miscreant manages to install a malicious application, she will be able to access all controller's information and to tamper with its behavior. There are several paths that a miscreant can take for such installation, for example by compromising the machine or using social engineering techniques. A malicious user could also upload the application to the app store, for example by hiding the malicious functionalities around legitimate ones, and lure her victim to download and install it. Note that we did not conduct

[9] https://www.fanuc.eu/it/en/cnc/development-software/focas-development-libraries

[10] https://www.myokuma.com/

this experiment for legal reasons. In our experiments, we managed to compromise the controller under test via a well-known system vulnerability (MS10-61) so as to install our application without notice. The malicious application we developed for testing mimicked a bot reaching out to the attacker via a call-back, and waiting for commands to be prompted to the backdoored CNC.

4 Findings

Our research reported issues common to many of the controllers under exam. We provide the summary of our findings and discuss their security implications.

First of all, the controllers we analyzed are equipped with either obsolete and legacy software, or software encompassing a large number of known vulnerabilities. Although this issue is well-understood in the ICS realm, and we were not surprised to run into obsolete software, we would have expected that machine tools like CNCs – that can easily cost a million dollar – would come with auto-updating mechanisms or, at least, mechanisms to inform the end-user of a need for an update. This is especially true in the context of Industry 4.0, in which machines tend to be normally connected to the network.

Second, several networking technologies do not support authentication, or do *not* have authentication enabled by default. In particular, only DNC and Focas have support for authentication, while MTConnect, Ethernet Q and THINC-API not have (note that THINC-API is a corner case because is exposed only locally). This issue is very severe because offers to any malicious user the possibility to abuse of the unauthenticated services.

Third, resource access control is lacking on most of the architectures of the controllers: A user (or a process) is often given full access to any system's resource, including its file-system or memory locations. For example, an application written on top of THINC-API will have full access permission to any system's resource including the internal controller configurations; with Ethernet Q, a remote user can write to memory locations mapped outside of the running process.

Fourth, the monitoring services expose a large amount of information. On one side, this is expected because those services have been, as said, introduced to make CNC machines compliant with Industry 4.0 paradigm. However, the information can be abused by a miscreant, especially given that authentication is often not available. In our experiments, we confirmed that all analyzed controllers suffer from data leakage problems resulting in confidential information being exposed to 3rd parties (e.g. programs code).

4.1 Impact

Overall, as depicted in Table 3, our evaluation identified 18 attacks (or attack variations) that we grouped into five attack classes: compromise, damage, denial-of-service, hijacking, and theft[11].

[11] When an attack is reported multiple times is because it consists of variations of the same attack. For example, “change tool geometry” can be leveraged to achieve dam-

Attack Class	**Attack Name**	**Haas**	**Okuma**	**Heidenhain**	**Fanuc**	**Total**
Compromise	RCE	✓	✓	✓		3
Damage	Disable feed hold	✓				1
	Disable single step	✓		✓		2
	Increase tool life	✓	✓	✓		3
	Increase tool load	✓	✓		✓	3
	Change tool geometry	✓	✓	✓	✓	4
	Decrease tool life	✓	✓	✓		3
DoS	Decrease tool load	✓	✓		✓	3
	Change tool geometry	✓	✓	✓	✓	4
	DoS via parametric program	✓	✓	✓	✓	4
	Trigger custom alarms	✓		✓		2
	Ransomware	✓	✓	✓		3
Hijacking	Change tool geometry	✓	✓	✓	✓	4
	Hijack parametric program	✓	✓	✓	✓	4
	Program rewrite		✓	✓	✓	3
Theft	Leak production information	✓	✓	✓	✓	4
	Leak program code		✓	✓	✓	3
	Screenshot			✓		1
	Total	15	14	15	10	

Table 3: Summary of the attacks identified in our research.

Among the different controllers that we tested, we observed a consistency in the number of problems: Haas, Okuma and Heidenhain yielded a similar amount of issues (15), with Fanuc having 10 attacks confirmed. This is a symptom that security does not seem to be a priority for controller manufacturers. This, together with the possibility of CNC machines being misconfigured and exposed to corporate networks, or worse to the Internet, creates serious and compelling problems.

Considering the same table on a line-by-line basis, the scenario is not better. Among all attacks, only two are confirmed to apply to a single vendor only (i.e., disable feed hold and theft via screenshot). On the other hand, six attacks are confirmed on all vendors.

Features like the configuration of the geometry of the installed tools, or the modification of the variables used by a parametric program with values supplied via network are automation-facing options, needed when dealing with complex automation and unsupervised process. Although these requirements are nowadays more common in manufacturing, vendors do not seem to take into account unwanted consequences of these features, thus raising concerns about security.

Compromise The first class of attacks consists of issues that result in a compromise of the CNC machine. While the focus of our research is limited to domain-specific problems, we also conducted a general assessment of the security posture of the controllers under analysis, including the simulators. For

age, denial-of-service, or hijacking; this depends on which geometries are changed, the type of machine and the manufacturing process. Vice-versa, distinct attacks can conduct to the same goal. For example, an attacker can take control of the production of an exposed CNC by hijacking a parametric program, by modifying the geometry of a tool to introduce a micro-defect, or by changing the executed program.

this, we used standard vulnerability assessment tools like Nessus with the aid of manual analysis and inspection.

Our experiments confirmed that several CNCs were prone to compromise at different levels including obsolete software or operating systems, weak OEM passwords or service credentials, enabled jumpers that allowed firmware extraction. Considering that our tests were conducted on CNCs ready to be delivered to the end-users, this reveals a general lack of awareness with respect to security.

Damage This class of attacks consists in damaging either the machine (or part of the machine, such as the tool or the spindle), or the part in production. CNCs are costly machines, with prices ranging from a few thousand to millions of US dollars, so damage is an important issue. Not only is the damage to be considered in terms of breakage of machinery components, and therefore the economic burdens on the end-user, but some interventions to replace the damaged elements also require procurement of complex assemblies, with logistic times usually on the order of weeks or months. Furthermore, the replacement interventions of these components require days of work and a phase of zeroing of the geometries of the machine tool (for example, the setup of the axes), thus introducing, in addition to the monetary cost, an impediment in terms of use of the machine for varying times.

We identified five attacks that could lead to damage. Due to the lack of space, this paper will present two of them.

Feed hold is a functionality that enables an operator to pause the execution of a machine, by stopping the feed axes, for example, to inspect the part in production during a program run. In our experiments, we confirmed that one vendor, Haas, is vulnerable to an attack in which a malicious user can remotely disable the feed hold while being used: an operator pressing the pause button of the machine will not be able to pause the manufacturing. For this vendor, the attack involves abusing the lack of authentication and access control on Ethernet Q to set the global variable 3004 to 7.

Another attack consists of tampering with the geometry of the tools. Each tool used by a CNC needs to be measured in any of its fundamental geometric quantity, depending on the type of machine and manufacturing process. A correct measurement is a must in computing the quotes for working a part within tolerance. In addition to that, any manufacturing process consumes the tool, for example, by reducing the overall geometry of the cutting edge. To address this need, a parameter called *wear* is used as a form of compensation. For example, in the case of a vertical milling machine used to drill holes in a raw part, a negative wear causes the column to crash into the part with damage on the tool or the spindle. Unfortunately, we found that this attack successfully works in all its variations and on all manufacturing controllers, including simulators and real-world installations as we demonstrate in Section 5.

Denial-of-Service Miscreants are often interested in sabotaging the operations of a targeted organization, such as a competitor or a generic victim they can profit from, for example, by demanding a ransom to restore the normal functionalities. With DoS, we mean all attacks aimed at disrupting the manufacturing process, for example, by stopping the machines from operating, or at slowing down the production with the end goal of reducing the efficiency of the industrial process.

We identified six attacks leading to DoS. One of these consists of lowering the load parameter associated with a tool, in order to slow down the production. This attack works because the controller automatically tunes the spindle's speed according to the capacity of the tool installed on the machine.

Another way of causing DoS is triggering alarms so as to block the current execution and request the intervention of the operator. Unfortunately, our evaluation reported that two vendors permit generating software alarms remotely. Although this feature can, to a certain extent, make sense in the development of a program for CNC applications, for example, for a program to trigger an alarm in certain conditions, it is arguable whether it would make sense to offer this option through a remote network call.

We also confirm the possibility to ransom the machines under test by compromising and installing an add-in that locks the HMI (Okuma), or by encrypting the G-code programs exposed via network shares, which were by default unprotected on Haas and Heidenhain.

Hijacking With hijacking we refer to the possibility for a miscreant to either introduce a micro-defect in the manufacturing process, or to replace the program in execution with one of her choice. In our experiments, we confirmed that all vendors were vulnerable to a change of a tool geometry aimed at introducing a micro-defect. With this, an adversary can take control of the manufacturing process to introduce very small micro-defects that might pass the QA process. These would eventually result in big financial or reputational losses for the victimized manufacturer.

Another option for hijacking the production is to alter the logic of a parametric program. By substituting the values of the memory variables used by a parametric program, an attacker can influence the final outcome. An example of this attack is the production of components "in sizes", in which the difference in geometry is often controlled by the selection (that is, activation) of specific program blocks for manufacturing the size or configuration of the work geometry in a parametric way. The modification of these values leads to the introduction of defects or to the production of wrong sizes compared to what is set by the operator on the HMI. All controllers were affected by this issue.

Finally, on three of the four controllers, we managed to replace the executed program with one of our choice without requiring any operator intervention or notice.

Theft Theft is a major concern in the manufacturing world. Production includes sensitive information that a manufacturing process produces and that an adversary is interested in monitoring or stealing. In our evaluation, we confirmed that all tested vendors expose such private information to varying degrees. The information we confirmed being exposed within the tested machines includes how many parts are produced, the name of the program associated with each production, the name of the machine, its serial number and related controller version, the active screen or menu on the HMI, the tool number, and part program comments.

Program files constitute a highly sensitive intellectual property because they specify the movements that a machine has to perform to conduct the machining. If an adversary manages to get access to these files, she could reproduce the part on her side or learn all the details behind the manufacturing, as in the case of an adversarial competitor. Theft of program files becomes of even greater concern in consideration that programs developed in G-code are not compiled. In our work, we managed to leak the content of the executed program on three controllers. In all the cases, we performed the attack via network, that is, without the need to bypass any security mechanism like brute-forcing an authentication procedure. In the case of Okuma, the MTConnect service exposes by default the block line currently executed, thus enabling an attacker to poll the daemon to reconstruct the code. For Heidenhain, its DNC interface is by default unauthenticated and a user can therefore remotely dump the executed program (via RPC or LSV2). Similarly, Fanuc exposes such data via Focas.

Finally, the DNC interface of Heidenhain can be abused to take screenshots of the operator's HMI. This enables a miscreant to spy on the manufacturing process, potentially accessing information such as the part program code, the tools list, or the machine configurations in an even more simplified way.

5 Use Cases

In this Section, we provide few a examples among the many attacks that we conducted on our real-world CNC installations, showing how we implemented them and discussing their practical impact[12].

The first experiment consists of abusing the Ethernet Q Commands interface of Haas to conduct three attacks: introducing a micro-defect in the manufacturing process (hijacking), performing a DoS, and damaging a tool. These attacks are possible because Ethernet Q Commands allows for altering the geometry of a tool remotely. As previously mentioned, the controller exposes this interface by default and does not provide authentication nor resource access control.

We conducted these attacks on a Haas Super Mini Mill machine – shown in operation during our experiment in Figure 2. For this experiment, we developed a program that instructed the machine to engrave four *equal traces* in a part of raw metal. The engraving was supposed to be 5.05 mm deep, as measured in Figure 3a. The result of the manufacturing cycle is shown in Figure 4. The part on the left shows the correct execution of the manufacturing, with four traces of the same depth.

[12] An extensive list of use cases are provided in our technical report [4].

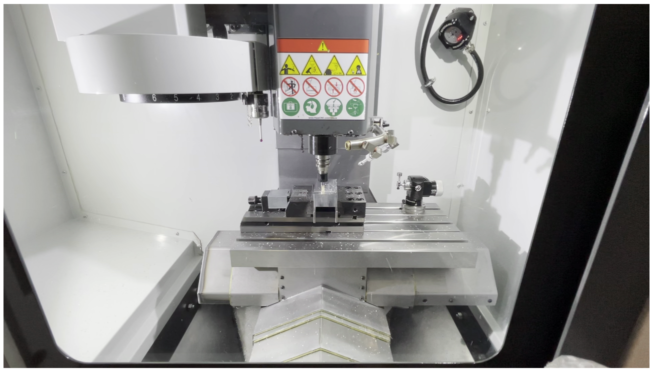


Fig. 2: The Haas Super Mini Mill engraving the first trace.

(a) Correct engraving.

(b) Defective engraving.

Fig. 3: Example of hijacking attack.

At this point, we ran our attacks by altering the wear parameter three consecutive times. First, we set a wear of +0.25 mm on tool number 1 to introduce a micro-defect:

```
$ echo "?E2201 0.25" | nc <IP> 5000
```

Then, we set the same wear to +5.50 mm, which is more than the original depth of the engraving. Finally, we set the wear to -10 mm.

The result of our attacks is shown in the right part of Figure 4. This part shows only two engravings instead of four. The first engraving is the reference one and corresponds to the normal execution of the machine. The second engraving has a depth of only 4.80 mm as measured in Figure 3b, i.e. with an error of 0.25 mm as per attack.

The other two engravings were not made because: In one, the machine operated above the plane of the raw part due to the wear being higher than the depth (5.50 mm > 5.05 mm); in the other, the machine crashed the tool against

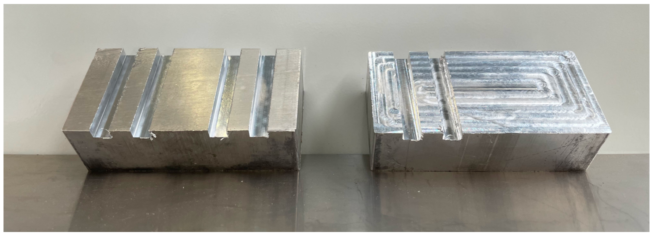


Fig. 4: The correct process on the left, and that of our confirmed attacks on the right.

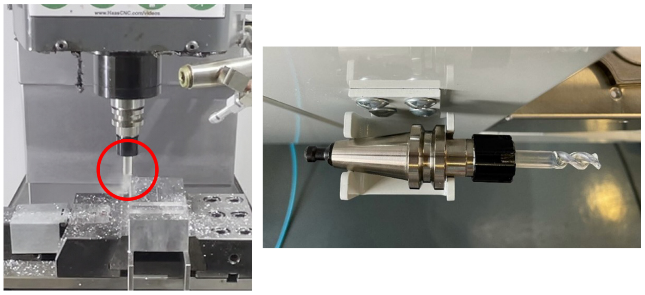


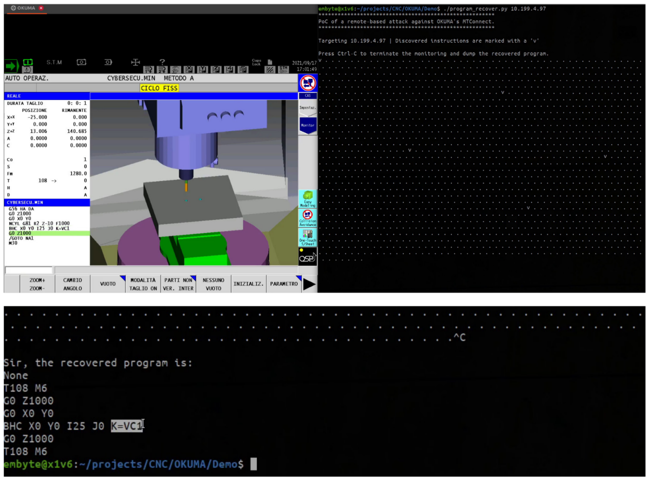
Fig. 5: The 3D-printed plastic tool for our damaging experiment, which crashed against the raw material (left), and a detail thereof (right).

the raw part because of the negative wear (-10 mm). For this last attack, we printed a plastic tool with a 3D printer, which we voluntarily broke against the raw part during the attack as shown in Figure 5.

With this single experiment, we demonstrated how an attacker can remotely alter the geometry of tools to conduct attacks with three goals: hijacking the production to insert a micro-defect, making the machine operate above the plane of the material (DoS), and damaging the production's tool or part.

The next experiment shows how to leak the program code running on the machine. Three tested controllers were affected by this issue. In the case of Okuma, the agent reports several useful information related to the manufacturing process like the number of installed tools or the position of the axes. The problem lies with the fact that the same agent reports both the name of the executed program and the code block (i.e., the instruction) currently executed on the machine. As result, a miscreant can pool the service to fetch the executed instructions shown in Figure 6. This is a severe issue because it required nothing more than connecting to the exposed service for conducting the attack. We communicated this issue to Okuma, which promptly acknowledged and fixed it.

One important consequence of being able to dump the executed program is the act of reverse-engineer it, which is fairly easy with G-code. This, leads to the next use case: parametric program hijacking. As we discussed previously, it



```
AUTO OPERAZ.          CYBERSECU.MIN   METODO A
REALE
DURATA TAGLIO              0: 0: 1
          POSIZIONE      RIMANENTE
X+X        -25.000          0.000
Y+Y          0.000          0.000
Z+Z         13.006        140.685
A            0.0000         0.0000
C            0.0000         0.0000

Co                              1
S                               0
Fm                         1280.0
T             108  ->           0
H                               A
D                               A
CYBERSECU.MIN
 G56 HA DA
 G0 Z1000
 G0 X0 Y0
 NCYL G81 R2 Z-10 F1000
 BHC X0 Y0 I25 J0 K=VC1
 G0 Z1000
 /GOTO NA1
 M30
```

```
embyte@x1v6:~/projects/CNC/OKUMA/Demo$ ./program_recover.py 10.199.4.97
*******************************************************
PoC of a remote-based attack against OKUMA's MTConnect.
*******************************************************

Targeting 10.199.4.97 | Discovered instructions are marked with a 'v'

Press Ctrl-C to terminate the monitoring and dump the recovered program.
v . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . v . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . v . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . v . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .^C

Sir, the recovered program is:
None
T108 M6
G0 Z1000
G0 X0 Y0
BHC X0 Y0 I25 J0 K=VC1
G0 Z1000
T108 M6
embyte@x1v6:~/projects/CNC/OKUMA/Demo$
```

Fig. 6: The dump of the executed program's source code via an unauthenticated and exposed MTConnect agent.

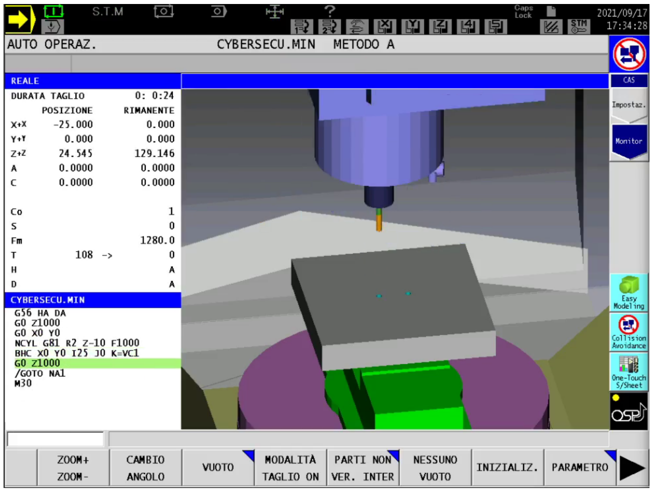
is a common practice of developers to use variables to dynamically change the execution flow of a program (as in a sort of conditional IF). In our example, we have a program that is supposed to drill K holes, where K is controlled by the variable VC1, as we highlighted in the instruction block of Figure 6. In this use case, K holds a value of 2 and the machine drills two holes, as shown in Figure 7.

At this point, an attacker that understands the program can remotely replace the content of the variable with an arbitrary value (such as 25) in order to hijack the production. This would alter the production to suit the attacker's needs, slowing down the production, or damaging it. Figure 8 shows this example in practice. All tested controller are affected by this issue.

6 Responsible Disclosure and Mitigations

In conducting this research, we wanted to raise awareness in a domain in which security didn't, yet, seem to considered an important driver. With this goal in mind, we underwent an important disclosure process and communicated our findings in a timely and responsible manner with the vendors of the tested controllers. This process was not easy, and required strong commitment on our side in engaging with the right peers and educating them on the importance of the issues that we identified. The large amount of demo material that we collected during our experiments helped in this direction.

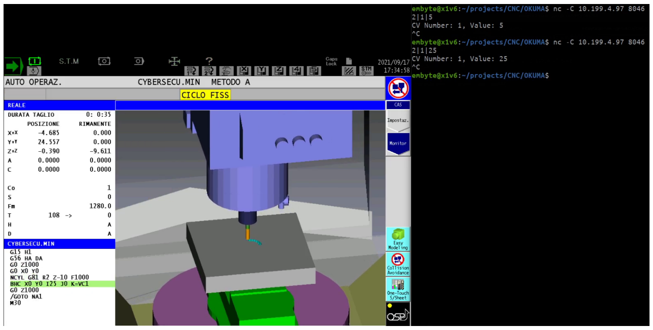
Fortunately, all vendors acknowledged our concerns and most of them have addressed, to various degrees, our findings in a reasonable time frame. More importantly, all of them have expressed interest in our research and have decided to improve either their documentation or their communication efforts with the



```
2021/09/17
17:34:28
AUTO OPERAZ.            CYBERSECU.MIN   METODO A
REALE
DURATA TAGLIO           0: 0:24
      POSIZIONE       RIMANENTE
X+X    -25.000            0.000
Y+Y      0.000            0.000
Z+Z     24.545          129.146
A       0.0000           0.0000
C       0.0000           0.0000

Co                            1
S                             0
Fm                       1280.0
T           108  ->           0
H                             A
D                             A
CYBERSECU.MIN
 G56 HA DA
 G0 Z1000
 G0 X0 Y0
 NCYL G81 R2 Z-10 F1000
 BHC X0 Y0 I25 J0 K=VC1
 G0 Z1000
 /GOTO NA1
 M30
```

Fig. 7: A parametric program executing two holes as per legitimate operation.



```
2021/09/17
17:34:58
AUTO OPERAZ.            CYBERSECU.MIN   METODO A
                        CICLO FISS
REALE
DURATA TAGLIO           0: 0:35
      POSIZIONE       RIMANENTE
X+X     -4.685            0.000
Y+Y     24.557            0.000
Z+Z     -0.390           -9.611
A       0.0000           0.0000
C       0.0000           0.0000

Co                            1
S                             0
Fm                       1280.0
T           108  ->           0
H                             A
D                             A
CYBERSECU.MIN
 G15 H1
 G56 HA DA
 G0 Z1000
 G0 X0 Y0
 NCYL G81 R2 Z-10 F1000
 BHC X0 Y0 I25 J0 K=VC1
 G0 Z1000
 /GOTO NA1
 M30
```

```
embyte@x1v6:~/projects/CNC/OKUMA$ nc -C 10.199.4.97 8046
2|1|5
CV Number: 1, Value: 5
^C
embyte@x1v6:~/projects/CNC/OKUMA$ nc -C 10.199.4.97 8046
2|1|25
CV Number: 1, Value: 25
^C
embyte@x1v6:~/projects/CNC/OKUMA$
```

Fig. 8: The same parametric program executing 25 holes after hijacking.

machine manufacturers, with the final goal of offering to the end-users more secure solutions.

Table 4 provides a short summary of this process. CISA's ICS-CERT extended invaluable help and support during our discussion with the vendors, for which we are grateful.

We also propose mitigation strategies for both manufacturers and end-users / integrators. With regards to controller manufacturers, we recommend adding support for authentication on all services and enforcing authentication by default. Additionally, we encourage manufacturers to adopt appropriate authorization schemes in the design of their systems, such as privilege separation and access management.

For integrators and end-users, we suggest the following mitigation strategies: Use of context-aware IPS/IDSs that regularly keep up with newer industrial

Vendor	Issues (and CVEs)	Contact Date	Ack Date	Feedback
Haas	Abuse of Ethernet Q Commands (CVE-2022-2474, CVE-2022-2475, CVE-2022-41636). RCE via Java JMX. Firmware extraction via enabled boot jumper.	17/11/21 (direct). 13/01/22 (CERT)	20/07/22	Issues acknowledged and public advisory released. The simulator won't be fixed because out of scope.
Okuma	RCE via CVE-2010-2729. Abuse of THINC-API. Code leak via MTConnect.	19/11/21 (direct)	25/11/21	Issues acknowledged and MTConnect fixed. THINC-API won't be fixed due to performance reasons.
Heidenhain	Abuse of DNC (CVE-2022-41648). Weak OEM password. Multiple known vulnerabilities.	04/02/22 (direct). 01/03/22 (CERT)	10/05/22	Issues acknowledged and public advisory released.
Fanuc	Abuse of Focas.	07/03/22 (direct). 29/03/22 (CERT)	27/04/22	Issues acknowledged and documentation enhanced. Added support for authentication.

Table 4: A summary of our responsible disclosure process with the vendors.

protocols. Correct network segmentation should be implemented to isolate CNC machines from other network assets. Consider modern CNC machines as part of an organization's IT assets and follow the same patch management procedures as any other equipment, such as desktop computers or servers. In our research, we also collaborated with a vendor to add support for proprietary CNC protocols

7 Related Work

While previous work has addressed the security of smart manufacturing technologies, including CNCs to a limited extent, our extensive evaluation of the CNC domain using both controller simulators and real-world machines sets our research apart as the first of its kind.

Quarta et al.[10] conducted a security analysis of an industrial robot. By using a real-world industrial robot, the authors analyzed its architecture and evaluated the associated risks. However, this paper differs from our work in the following ways: firstly, our work focuses on the overall ecosystem of computer numerical controls while this paper focuses on a single robot and its implementation; secondly, our work includes the analysis of CNC machines which differ significantly from industrial robots in terms of design, architecture, and implementation of both software and protocols; thirdly, manufacturers of industrial robots such as ABB, do not typically offer CNC solutions (and vice versa), highlighting the substantial differences between these two types of machine tools.

In a follow-up study, Pogliani et al. [9] explored the security risks associated with bad practices in code development for modern industrial robots. The authors proposed a static-code analysis tool to detect security vulnerabilities in robot code and used it to show that certain implementations of programs found online were effectively vulnerable to different classes of attacks. This work differs in focus from ours. Additionally, the programming languages used in industrial robots (e.g. RAPID and KRL) are quite different from those ones in the CNC domain (G-code, M-code, proprietary macros). Maggi et al. [7] investigated how smart factory floors are exposed to potential security threats in Industry 4.0. They reported security issues at different levels including abusing industrial add-ins or compromising digital twins in software simulators. Their research explored the risks of the industrial ecosystem as a whole, showing that modern smart installations give rise to a larger attack surface, compared with previous generations of industrial facilities. This work touches on the security of the different systems without going vertical on a single category. In addition, the problems identified related with common OS functionalities rather than domain-specific features. Balduzzi et al. [2] looked at industrial gateways used in smart factories to enable communication between modern and legacy devices. The authors reported issues in which translations occurred for example from Modbus TCP to RTU. Niedermaier et al. [8] showed how PLCs can be influenced by packet flooding. The authors conducted an experiment with 16 devices from six vendors, and demonstrated that all except for one device are susceptible to network flooding attacks. Maggi et al. [6] looked at the radio protocols used to remotely control industrial machinery. Their research indicated that multiple vendors were prone to the same class of problems: the ability for a miscreant to arbitrarily generate fake radio messages and sabotage the operation of industrial plants. Similar problems were reported by Balduzzi et al. [3] who conducted a security analysis of a radio protocol standard used in the maritime industry for monitoring and tracking logistics and passenger ships.

In a work closer to ours, Chen et al. [5] discussed the hypothetical risks associated with CNC machines, reporting issues related to a CNC's terminal. The authors proposed mitigation strategies like the adoption of cryptographic schemes for data protection, or industrial gateways for proper network segmentation and access control. Similarly, Tu et al. [11] proposed a trusted security framework for CNC machines. Although these works sit in the same domain of research as ours, they provide different research methodologies and contributions. Our work is closer to the real-world implementations of CNC machines, in having conducted an empirical evaluation of the security boundaries of the technologies put in place by controller manufacturers according to the needs dictated by Industry 4.0.

8 Conclusions

Our research explored the risks associated with the adoption of Industry 4.0 in CNC machines. These machines underwent a shift from standalone systems to network-enabled ones that resemble full-fledged machines more closely than they do mechanical devices. As a result, end-users are left dealing with sophisticated systems that, if not correctly configured or poorly designed, might open the door to abuse.

In our research, we explored technologies specific to the CNC domain and conducted an extensive security evaluation. We implemented PoC attacks on real-world installations, demonstrating that our concerns have practical implications, and identified important issues that are common among all controllers under test.

In addition to publishing our findings in this research paper, we also created demo material to educate the community about the security risks in the CNC domain. Our responsible disclosure process prompted interest from the affected manufacturers, who acknowledged our findings. Our aim is to raise awareness in a field that we believe will gain more attention in the future.

References

1. Interchangeable variable block data format for positioning, contouring, and contouring/positioning numerically controlled machines. Electronic Industries Association (1979)
2. Balduzzi, M., Bongiorni, L., Flores, R., Lin, P., Perine, C., Vosseler, R.: Lost in translation: When industrial protocol translation goes wrong. Trend Micro (2020), `https://www.madlab.it/papers/wp-lost-in-translation-when-industrial-protocol-translation-goes-wrong.pdf`
3. Balduzzi, M., Pasta, A., Wilhoit, K.: A security evaluation of ais automated identification system. In: Proceedings of the 30th annual computer security applications conference (2014)
4. Balduzzi, M., Sortino, F., Castello, F., Pierguidi, L.: The security risks faced by cnc machines in industry 4.0. Trend Micro (2022), `https://www.madlab.it/papers/cnc.pdf`
5. Chen, X., Wang, Z., Yang, S.: Research on information security protection of industrial internet oriented cnc system. In: 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC) (2022)
6. Maggi, F., Balduzzi, M., Andersson, J., Lin, P., Hilt, S., Urano, A., Vosseler, R.: A security evaluation of industrial radio remote controllers. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. pp. 133–153. Springer (2019)
7. Maggi, F., Balduzzi, M., Vosseler, R., Rösler, M., Quadrini, W., Tavola, G., Pogliani, M., Quarta, D., Zanero, S.: Smart factory security: A case study on a modular smart manufacturing system. Procedia Computer Science (2021)
8. Niedermaier, M., Malchow, J.O., Fischer, F., Marzin, D., Merli, D., Roth, V., Von Bodisco, A.: You snooze, you lose: Measuring plc cycle times under attacks. In: WOOT@ USENIX Security Symposium (2018)
9. Pogliani, M., Maggi, F., Balduzzi, M., Quarta, D., Zanero, S.: Detecting insecure code patterns in industrial robot programs. In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. pp. 759–771 (2020)
10. Quarta, D., Pogliani, M., Polino, M., Maggi, F., Zanchettin, A.M., Zanero, S.: An experimental security analysis of an industrial robot controller. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 268–286. IEEE (2017)
11. Tu, S., Liu, G., Lin, Q., Lin, L., Sun, Z.: Security framework based on trusted computing for industrial control systems of cnc machines. In: International Journal of Performability Engineering (2017)