Security Analysis of Networked 3D Printers and their Deployments

Matt McCormack, Sanjay Chandrasekaran, Tianlong Yu, Guyue Lui,
Sandra DeVincent Wolf, Vyas Sekar

March 6, 2020

CMU-CyLab-20-001

CyLab
Carnegie Mellon University
Pittsburgh, PA 15213

Security Analysis of Networked 3D Printers and their Deployments

Matt McCormack, Sanjay Chandrasekaran, Tianlong Yu, Guyue Lui, Sandra DeVincent Wolf, Vyas Sekar
Carnegie Mellon University

Abstract

Networked 3D printers are an emerging trend, enabling agile manufacturing. However, they are simultaneously increasing the security threats to manufacturing by creating new ways for attackers to cause physical hazards, steal proprietary data, create defective parts, or halt operations. Prior work has given limited attention to the security implications of adding these machines to a network. In this work, we present C3PO, an open-source network security analysis toolfor systematically identifying security threats to networked 3D printers. C3PO's design is guided by industry standards and best practices. It identifies potential vulnerabilities in data transfer, the printing application, availability, and exposed network services. Furthermore, C3PO analyzes the security implications of a 3D printer's network deployment, such as an attacker compromising a camera to modify printing instructions "on-the-wire." We use C3PO to analyze 13 networked 3D printers and 5 real-world manufacturing network deployments. We identified 33 network security trends in networked 3D printers such as a susceptibility to low-rate denial of service attacks (all 13), transmitting unencrypted data (12/13), and being deployed on publicly accessible networks (2/5). We leverage these findings to provide recommendations on securing networked 3D printers and their deployments.

1 Introduction

Additive manufacturing (i.e., 3D printing) is a key enabler of agile manufacturing [8, 20, 26]. Aviation and other safety-critical domains desire to utilize 3D printing [9, 17, 20, 54].

While there is significant potential for impact (e.g., excitement surrounding the advent of a "Fourth Industrial Revolution" [26]), there are also security concerns (e.g., [13, 29, 32, 37, 57]). For example, networked manufacturing machines in a Boeing airplane production facility were infected with malware, stopping production [40]. These cyber vulnerabilities have high monetary costs, many escalating to more than $1M in damages [27].

Indeed, prior work (illustrated in Table1) demonstrated that an attacker can create defective parts by modifying CAD files (on a PC prior to being sent to a 3D printer) or the printer firmware [6, 63, 68] to have the correct appearance but different physical properties [6, 25, 78]. Additionally, networked 3D printers create new vectors for performing known attacks, such as stealing data [70, 84] and halting operations [33, 69], that have been performed on other manufacturing devices.

Most of these prior efforts focus on directly tampering with the control PC or the printer firmware. However, as these deployments are increasingly interconnected, we should also be concerned about threats from *network attackers* (e.g., physical hazards, denial of service, etc.). Unfortunately, there are currently no tools for identifying if a 3D printer is susceptible to these types of attacks. Existing tools lack: (1) coverage of multiple categories of vulnerabilities (e.g., identify out of date services but not availability vulnerabilities), (2) support for multiple vendors/protocols[1], and (3) ignore other devices on the network (i.e., the network deployment). These limitations highlight the need for a security analysis tool that can identify potential vulnerabilities across 3D printer vendors while also analyzing the security of the network deployment.

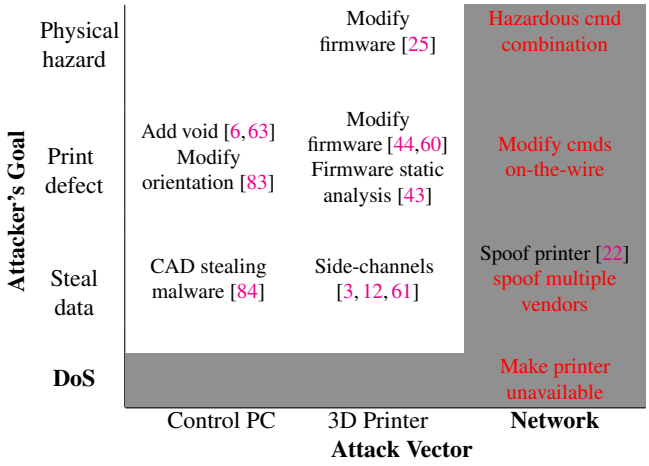


Table 1: Prior work on 3D printer attacks, characterized by attacker goal and the attack vector in the 3D printing workflow. The shaded cells are new contributions we make, and we demonstrate the attacks in red on commercial networked 3D printers.

To this end, we designed and implemented an open-source security analysis tool, Connected 3D Printer Observer (C3PO) [7], to systematically identify potential security vulnerabilities on networked 3D printers guided by key recommendations from industry standards [28] and best practices [19,48]. C3PO is composed of two parts:

- The first part identifies machine-specific vulnerabilities on standalone 3D printers (i.e., the device in isolation).
- The second part searches for potential multistage attack paths[2] that allow an attacker to cause a physical haz-

[1] Currently, there are 50+ vendors [1], and our survey of 9 vendors identified 7 protocols for transferring instructions to a networked 3D printer.

[2] An example of a multistage attack is an attacker compromising an IIoT camera and using it to launch a DoS attack against a 3D printer.

ard, create defective parts, steal data, or halt operations. To achieve this, we demonstrate the practical application of attack graphing tools for identifying intermediate nodes (e.g., IIoT cameras) that impact the security of a networked 3D printer.

We used C3PO to analyze 13 networked 3D printers, representing 9 vendors, across the spectrum of costs and printing processes. Additionally, we used C3PO to analyze five real-world 3D printer network deployments, covering multiple network sizes and complexities. Each network deployment was analyzed with 19 scenarios that assumed the presence of different vulnerabilities (e.g., default credentials on IIoT cameras, PCs running Windows 95, etc.).

Findings: Our key findings are:

- *Standalone:* With respect to individual 3D printers, we identified 33 vulnerabilities that enabled us to perform four attacks (such as driving the printer into a part, modifying printing instructions "on-the-wire", and DoS). Table 1 summarizes our findings and contrasts to prior work. All 13 networked 3D printers were vulnerable to simple DoS attacks (e.g., SYN flood), some requiring a power-cycle to recover. Most (12 of 13) did not encrypt data in transit. 4 of 13 allowed network inputs that crashed the machine. Finally, 4 of 13 were vulnerable to a published exploit (such as WannaCry [42]).
- *Deployments:* Many networked 3D printers were unnecessarily placed on publicly accessible networks and could be remotely accessed via IP. Deployments contained a significant proportion (>41%) of embedded devices (e.g., IIoT cameras) that could be used as potential launchpads for future attacks.

Based on our findings we derive insights and recommendations for defending networked 3D printers. Specifically, we envision "bolting-on" security after networked 3D printers are deployed using software-defined networking and network function virtualization to implement agile and specific security functions for each device. Thereby enabling the employment of flexible defenses that can be tailored to the specific needs of each machine.

Disclosure and Impact: We have disclosed our findings with all of the device vendors, and some have requested additional analysis of their new 3D printers to improve their product's security. Since our initial pilot studies, our tool has been requested by manufacturing center administrators and used to understand and improve their security posture. We are also in active conversations with facility operators and device vendors to implement our recommendations into practice.

2 Background and Motivation

We provide an overview of the 3D printing workflow and define our attacker goals. Additionally, we discuss prior work to motivate the need for a security analysis tool.

2.1 Background on 3D Printing Workflow

Additive manufacturing, often referred to as 3D printing, creates a physical object by sequentially joining layers of deposited material. This process enables fabricating structures that are not possible with traditional manufacturing methods [78]. The future of manufacturing relies on 3D printing as it reduces the cost of building complex parts, allows rapid design iteration, and enables on-demand production [8].

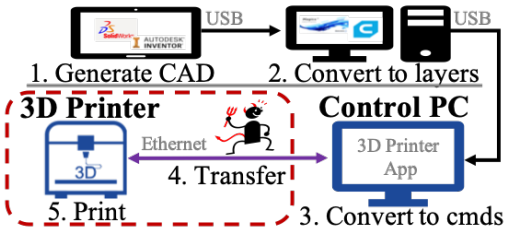


Figure 1: A general 3D printing workflow. Our work focuses on analyzing the security inside the red dashed box.

Workflow: The 3D printing workflow (shown in Figure 1) consists of the following five steps (where the first three steps can either be performed on the same host or multiple hosts).

1. *Generate CAD representation.* Create a digital representation, often as a stereolithography file (*.STL).
2. *Convert to layer representation.* Divide the digital representation into vertical layers.
3. *Convert to printing commands.* Generate machine-specific commands for each layer (e.g., G-code [66]).[3]
4. *Transfer commands over the network.* Printing commands are placed in a file and sent to the 3D printer.
5. *3D Print.* The commands in the file are either executed immediately or after a user action (e.g., a button push).

In 3D printer deployments, we noted that one control PC would be used with multiple networked 3D printers. Additionally, the operating model for 3D printers differs from many IoT devices (e.g., [4]) in three ways: (1) 3D printers lack mobile apps,[4] (2) the majority of network traffic remains on the local network, and (3) all networked 3D printers exposed at least one listening TCP-based service.

Our work evaluates the security vulnerabilities related to connecting a 3D printer to a network (i.e., red box in Fig 1).

Attack Goals: Based on prior work (e.g., [49, 76–78]), we envision an attacker with one of the following goals:

- **Causing physical hazards** [35]. All networked 3D printers have components that can pose a safety risk (e.g., high-power lasers, high-temperatures heaters, etc.). An attacker could manipulate these to cause a physical hazard, such as starting a fire by commanding the heater to its maximum

[3] G-code was used by 3 of the 13 networked 3D printers analyzed.

[4] Some vendors are beginning to release mobile apps for remote monitoring.

value while turning off safety features (e.g., the cooling fan) and driving the hot printer nozzle into the part (Figure 4).
- **Creating defective parts** [6]. A network attacker could send malicious commands to a 3D printer causing its software to crash midway through printing (Figure 5), forcing a multi-hour printing operation to be repeated.
- **Stealing proprietary data** [84]. Large manufacturing centers are composed of multiple networked 3D printers. Often new printing tasks are sent to the first available networked 3D printer. An attacker could advertise fake 3D printers in order to steal designs and create forgeries.
- **Halting printing operations** [33]. An attacker can send thousands of status requests to a networked 3D printer to overwhelm it's ability to respond to legitimate requests (Figure 6). Thereby prohibiting legitimate users from sending new files, resulting in a loss of productivity and potentially costing thousands of dollars [69].

2.2 Prior Work and Motivation

Prior work: We group prior attacks (e.g., [78]) based upon the attacker's goal and the attack vector (shown in Table 1). We highlight three main attack vectors: (1) the control PC (prior to printing instructions being sent over the network), (2) the network, and (3) the networked 3D printer. We noted that prior work has given limited attention to security risks arising from the network.

Motivation: As such, most demonstrated attacks have ignored the network as an attack vector. Some modified STL files at the control PC before they were sent over the network (e.g., [6, 63]). Others assumed physical access to allow modifying the printer's firmware (e.g., [25, 60]). Network security analysis of 3D printers has been limited to a single vendor and only identified data transfer vulnerabilities–missing availability vulnerabilities [22]. Furthermore, most of the prior work does not identify multiple types of vulnerabilities and does not scale to multiple vendors/protocols. Moreover, the 3D printer's network deployments have been ignored, missing potential multistage attacks (e.g., those leveraging other devices on the network). We revisit prior work in §7.

Threat model and scope: We limit our attacker to only accessing the 3D printer over the network (i.e., no physical access). As a starting point, we do not consider attackers who are seeking to be stealthy or evade countermeasures. An attacker can start with network access (e.g., insider threat) or gain it by compromising a device on the network. For example, an attacker could gain access to a control PC using an e-mail with a malicious link [6], an IIoT device using default credentials [5], or a networking switch using unpatched vulnerabilities [67]. Thus, the security of a networked 3D printer is impacted by its network deployment.

The combination of a 3D printer's individual vulnerabilities and its network deployment creates an array of possible attack paths for causing a physical hazard, creating defective parts, stealing data, or halting operations. Our C3PO security analysis toolaims to be a generic tool for identify susceptibility to these types of security risks from a connected device's network API. Additionally, it informs the design of "bolt-on" network defenses.

3 C3PO Tool Design

In this section, we first describe three requirements of our tool for identifying network vulnerabilities in 3D printers. Then, we discuss why existing tools cannot meet these requirements. Finally, we present our design of C3PO [7], an open-source security analysis tool for networked 3D printer and their deployments.

3.1 Tool Requirements

We identified three requirements for our security analysis tool:
- **R1: Increased coverage of vulnerabilities.** The tool should cover multiple vulnerabilities as often combinations of vulnerabilities are required for an attack to succeed (e.g., a broadcast query and a lack of encryption could be combined to spoof a printer and steal data).
- **R2: Protocol-agnostic.** The tool should not be designed for a specific 3D printer (or protocol, i.e., G-code), but support multiple vendors, including those using a closed-source, proprietary protocol.[5]
- **R3: Addressing complex deployment models.** The tool should be able to analyze complex deployment models and consider the security impacts from other devices which could be leveraged by attackers to achieve their goals.

Existing Tools: We are not aware of 3D printer specific network analysis tools. While many generic network security tools exist, they do not meet all of the above requirements. Existing IoT tools can only detect a small set of vulnerabilities. Additionally, some are device-specific (e.g., IoT Security Checker [15] and IoT Vulnerability Scanner [55] focus on login credentials, PENTOS [71] focuses on wireless security attributes). While others are protocol specific (e.g., PRET [45], OWASP Nettacker [82]).

To achieve high coverage, be protocol-agnostic, and address complex deployment models for networked 3D printers, we develop C3PO. It leverages existing tools (i.e., Nessus [65], Mutiny [62], and hping [58]) and adds modules specific to networked 3D printers. We discuss our detailed design next.

3.2 C3PO Overview

At a high level, C3PO consists of two stages. First, an individual 3D printer analysis for identifying machine-specific vulnerabilities in a standalone 3D printer (§4.2). Second, a

[5] In our survey, 5 of 9 vendors used distinct proprietary protocols.

network deployment analysis for identifying potential multi-stage attack paths through a 3D printer's network deployment using attack graphing (§5.2). We discuss the first stage and then show how its results are fed into the second stage to aid analyzing the network deployment.

3.2.1 Standalone 3D Printer Security Analysis

Table 2: Security assessment categories for networked devices from industry standards: IEC 62443-4-2 Foundational Requirements (IEC FR) [28], and best practices: 2018 OWASP IoT Top 10 (OWASP) [48] and IoT Security Foundation (IoTSF) [19].

Category	Attributes	Reference	In C3PO
1: Data Transfer	Lack Authentication	IEC FR 1 & 3 OWASP #1	X
	Lack Encryption	IEC FR 3 & 4 OWASP #7 IoTSF - G	✓
	Command Actuators	IEC FR 2	✓
	Broadcast Advertisement	IEC FR 5	✓
2: Printing Application	Management Commands	IEC FR 2	✓
	Insecure applications	IEC FR 2 & 3 OWASP #3 IoTSF - E	✓
	Insecure updates	IEC FR 3 OWASP #4 IoTSF - J	✓
	Lack of device management	IEC FR 1-3 & 6 OWASP #8 IoTSF - F, K & L	X
	Insecure default settings	IEC FR 7 OWASP #9	X
3: Availability	Robust to DoS	IEC FR 3 & 7	✓
4: Network Services	Outdated libraries	OWASP #5 IoTSF - D	✓
	Network access	IEC FR 1 & 5 OWASP #2 IoTSF - H	✓
5: Not Applicable	Insufficient privacy	OWASP #6 IoTSF - A	X
	Lack of physical hardening	IEC FR 3 & 7 OWASP #10 IoTSF-B	X

To provide coverage of vulnerabilities (R1), we ensure our tool identified network security attributes described in security standards [28] and best practices [19, 48] for networked devices (shown in Table 2). After pruning categories that were not applicable to the manufacturing domain (e.g., privacy) or could not be evaluated with only network access (e.g., physical hardening), we grouped the resulting attributes into four categories: (1) data transfer, (2) printing application, (3) availability, and (4) exposed network services. These are mapped to four corresponding modules in C3PO's first stage as shown in Figure 2, with the following workflow.

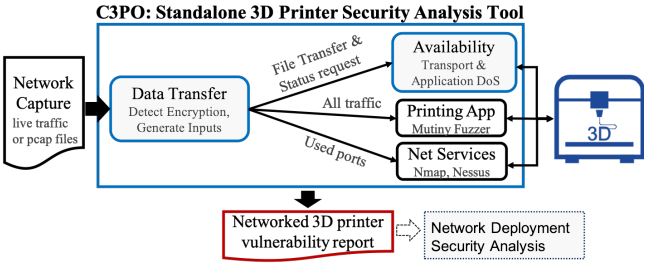


Figure 2: Overview of C3PO's networked 3D printer vulnerability analysis tool. Blue (Shaded) boxes represent our additions, and black ones are existing tools.

First, we feed a network capture to C3PO, which could be either live network traffic from the control PC or stored traces (e.g., pcap files). We don't assume any prior knowledge (e.g., protocol format, printer vendor) about the capture (R2). The network capture is analyzed by the *Data Transfer* module which determines whether encryption is used and generates a specific input to each of the following modules (as denoted in Figure 2). The *Availability* module takes possible printing commands (e.g., file transfer, status requests), which replicates these commands to test for both network and application layer availability issues. The *Printing Application* module is fed the entire network capture, where a mutational fuzzer [62] creates a large set of potentially malicious inputs for the printing application. The *Network Services* module uses existing tools to scan exposed network services [38] and identify known vulnerabilities [65]. A list of ports used in the network capture is sent to it for identifying potentially unused network services. Finally, C3PO collects the results from all of the modules and generates a vulnerability report for the networked 3D printer under test. Next, we discuss these key modules in detail.

Data Transfer Module: As many networked 3D printers use a closed-source, proprietary format to encode their printing commands, it is challenging to differentiate encryption from packed binary data. To overcome this challenge, we leverage prior work (e.g., [72, 73]) to determine if the data is encrypted by using three tests on the data: (1) calculate the entropy per byte, (2) perform a chi-squared test for a uniform distribution, and (3) calculate the serial correlation coefficient. We performed all three test on a per-packet level for the complete network capture. We separated out data with identifiable file headers (e.g., Gzip, JPEG, etc.). We infer that encryption is used if there is high entropy (>6.75 bits),[6] the chi-squared test results in a probability of a uniform distribution (p-value >0.01), and low serial correlation (<0.3).[7] If the data exchanged in both directions passes these tests we consider encryption to be used. If it only passes in a single direction, we consider the commands to possibly be encrypted but not sent over an encrypted channel.

[6] Test files of random string values had a maximum entropy of 6.65 bits.
[7] Files with >128 random bytes had a maximum serial correlation of 0.29.

Availability Module: Identifies DoS at two network layers.

- Transport layer. Analyzes the underlying network layer capabilities of the 3D printer, not sending any data to the printing application. Specifically, we test with a SYN flood (using hping [58]) and TCP connection exhaustion (e.g., multiple TCP sessions).
- Application layer. In order to remain protocol-agnostic, we use the input network capture as the input for generating all test cases. Specifically, we perform a stress test (e.g., sending multiple, concurrent status requests) and partial data exchange (e.g., only send the first 100 Bytes of a printing file, then keep the connection open indefinitely).

We used repeated messages (assumed to be status requests) for the stress test and the stream where the largest amount of data is sent from the control PC to 3D printer for the partial data exchange (assuming this to be the printing command file). Additionally, we run a slowloris [14] variant of the stress test to identify low-rate DoS vulnerabilities.

Printing Application Module: To identify potential vulnerabilities within the networked 3D printer's application software, we used an existing mutational fuzzer that generates inputs that match the protocol format found in the network capture [62]. This allows C3PO to leverage network fuzzing without having to know the protocol format (e.g., required for [50]) or requiring access to the control PC application (e.g., IoTFuzzer [11]). Fuzzed inputs were transmitted for 30 minutes while a benign status request message was used to identify an input that caused the application to crash.

3.2.2 Network Deployment Security Analysis

Identifying vulnerabilities for a standalone 3D printer is the first step, but does not convey the complete security picture because there may be other vulnerable devices (e.g., IIoT cameras, sensors, etc) in a manufacturing network. For example, a manufacturing deployment might add an internet connected camera on the 3D printer's subnet in order to remotely monitor the printing process. Thus, it is important to identify how these devices could be used by an attacker against networked 3D printers (R3).

The network deployment analysis component of C3PO addresses this problem. Its goal is to create an attack graph which identifies all possible attack paths to the networked 3D printer. Two inputs are required to achieve this goal. First, C3PO needs to automatically identify all of the other devices and their network connections to the networked 3D printer. Second, C3PO needs to identify each device's vulnerabilities in order to find all possible attack paths. However, this is challenging due to the wide array of devices, the complex network deployments, and the lack of models for attacks in the manufacturing domain. To address these challenges, our network deployment component includes three modules as shown in Figure 3.

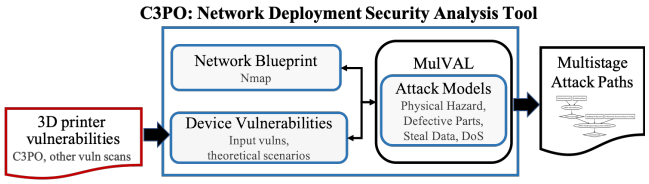


Figure 3: Overview of C3PO's network deployment analysis tool, extending prior attack graphing tools, blue (Shaded) boxes represent our additions.

Network Blueprint Module: Creates a network topology, listing all of the devices a single hop from our key assets (i.e., networked 3D printers and control PCs).

Device Vulnerabilities Module: Includes vulnerabilities for both networked 3D printers and other devices. The networked 3D printer's vulnerabilities can be provided by C3PO's individual 3D printer analysis. For other devices, we can either apply known vulnerabilities (e.g., from a vulnerability scan) or incorporate theoretical scenarios (for example, scenarios could be based upon common vulnerabilities for devices, such as IIoT cameras having default credentials [5]).

Attack Models Module: Takes the outputs from the previous two modules to create an attack graph. It consists of the set of networked 3D printers to evaluate, the attacker goals, the attacker's starting location (i.e., same local network or a remote network), and a mapping of vulnerabilities to attacks. With these inputs and models, we extends the attack graphing tool MulVAL [47] to perform a "what-if" analysis. It uses vulnerabilities from our theoretical scenarios (e.g., what attack paths exist if all IIoT cameras on the network have default credentials) to generate an attack graph, showing how an attacker can cause a physical hazard, create defective parts, steal data, or halt operations on the networked 3D printer(s) being evaluated.

4 3D Printer Evaluations

In this section, we present our findings from running C3PO on 13 networked 3D printers. In total, we identified 33 vulnerabilities covering each of the categories identified by industry standards and best practices (Table 2): insecure data transfer, availability, insecure printing application, and insecure network services.

4.1 3D Printers Evaluated

The 13 networked 3D printers evaluated ranged from low-cost, desktop polymer machines to $1M+, industrial metal 3D printers as shown in Table 3. The networked 3D printers represent two classes: (1) desktop and (2) industrial. The desktop machines generally print a polymer material and have a lower cost (<$5,000). While the industrial machines print

either polymers or metals, require significant space, and have higher costs (>$15,000). The desktop machines selected were among the top 10 sold on Amazon, and the industrial models were from the top industrial vendors. During our evaluation we transferred printing instructions for the same design file[8] to each networked 3D printer and observed six different protocols (five of which were proprietary).

Table 3: Networked 3D printers evaluated.

	3D Printer	Approx. Cost (US$)	Year Released	Material	Printing Protocol
Desktop	Machine A	300	2015	Polymer	G-code
	Machine B$^{\oplus}$	1,400	2019	Polymer	G-code
	Machine C	1,500	2014	Polymer	proprietary
	Machine D	2,850	2015	Polymer	proprietary
	Machine E	4,200	2016	Polymer	compressed G-code
Industrial	Machine F*	17,000	2017	Polymer	compressed proprietary
	Machine G*$^{\diamond}$	18,900	2008	Polymer	compressed proprietary
	Machine H*$^{\diamond}$	31,900	2007	Polymer	compressed proprietary
	Machine I$^{\diamond}$	50,000	2007	Polymer	STL
	Machine J$^{\oplus}$	150,000	2016	Metal	proprietary
	Machine K$^{\dagger\diamond}$	600,000	2010	Metal	proprietary
	Machine L*	750,000	2011	Polymer	compressed proprietary
	Machine M†	~1,000,000	2014	Metal	proprietary

$^{\oplus}$: Machines are the first model released by a new vendor
$^{\diamond}$: Machines in operational use but no longer supported by the vendor
*: Machines F, G, H & L are produced by the same vendor
†: Machines K & M are produced by the same vendor

4.2 Findings

We group our findings based upon the logical network layer where the vulnerability manifests.

4.2.1 Transport Layer

Our findings in the transport layer revealed an implicit assumption by 3D printers that the network is non-adversarial.

Observation 1: ***12 of 13 networked 3D printers did not use encryption when transferring data (i.e., a local attacker could steal data).***

Encryption: None of the networked 3D printers encrypted data in both directions (i.e., to and from the 3D printer, reference Table 4). Most data transfers exhibit an entropy of <5.48 bits and a serial correlation of >0.38 in one direction.[9] Additionally, only two had a majority of their packets pass a chi-squared test for their data being a uniform distribution, which encrypted data should pass. Only one may be encrypting the printing commands file prior to sending it over an unencrypted channel. This allows an attacker to potentially view file meta-data (e.g., filenames, length, etc.). As most did not utilize encryption, known file headers could be identified (e.g., Gzip, JPEG, etc.).

To put this in context of other IoT markets, we also ran C3PO on 11 commodity IoT devices (e.g., Amazon Alexa, D-Link camera, etc.). Among these IoT devices, 6 out of the 11 utilized encryption when transferring data. We manually confirmed that 5 of these IoT devices utilized TLS for data exchange. This suggests that networked 3D printers are behind the state of the art with respect to encrypting data being sent over the network. This is particularly surprising for the industrial networked 3D printers, as it creates a risk that proprietary data could be stolen.

Table 4: Test results for identifying encrypted data transfer on individual devices' network traffic.

Device	Used Encryp	Entropy/byte (Sent / Recv)	Serial Correlation (Sent / Recv)	Percent Packets $\chi^2 > 0.01$
Machine A	No	4.26 / 5.16	0.65 / 0.47	0.5%
Machine B	No	4.16 / 5.01	0.39 / 0.58	0%
Machine C	No	6.64 / 4.70	0.49 / 0.48	0%
Machine D	Possible	7.88 / 5.29	0.10 / 0.50	0.2%
Machine E	No	6.87 / 5.14	0.29 / 0.44	0.3%
Machine F	No	5.54 / 5.48	0.58 / 0.62	0%
Machine G	No	3.26 / 5.36	0.54 / 0.59	0%
Machine H	No	6.37 / 5.32	0.47 / 0.59	0%
Machine I	No	6.75 / 6.50	0.49 / 0.48	63.7%
Machine J	No	4.13 / 2.87	0.35 / 0.46	41.1%
Machine K	Not analyzed, machine. configured as a NAS server*			
Machine L	No	4.61 / 5.34	0.57 / 0.42	0%
Machine M	Possible	7.99 / 5.39	0.02 / 0.38	93.1%
IoT A	No	5.92 / 5.93	0.09 / 0.16	35.7%
IoT B	No	5.94 / 6.43	0.08 / 0.26	0.8%
IoT C	No	5.60 / 7.31	0.51 / 0.32	18.5%
IoT D	Yes	7.78 / 7.78	0.17 / 0.17	12.6%
IoT E	No	5.04 / 4.86	0.37 / 0.43	0%
IoT F	Yes	7.93 / 7.96	0.08 / 0.06	97.7%
IoT G†	Yes	7.93 / 7.99	0.12 / 0.001	97.7%
IoT H†	Yes	7.99 / 7.99	0.006 / 0.02	91.5%
IoT I†	Yes	7.52 / 7.74	0.33 / 0.24	74.8%
IoT J†	Yes	7.92 / 7.76	0.08 / 0.17	77.3%

*: Not specific to networked 3D printers
†: Manually verified to be using TLS

Observation 2: ***12 of 13 networked 3D printers were vulnerable to transport layer denial of service attacks (e.g., SYN flood crashes the 3D printer, requiring a power-cycle).***

We analyzed two transport layer denial of service issues: SYN flood and TCP connection exhaustion. This is an area that prior work has not explored for networked 3D printers, and allows many of the DoS attacks we demonstrated.

SYN Flood: During a SYN flood, 9 of the 13 networked 3D printers analyzed were unavailable on the network. Ad-

[8] A CAD file for a small boat [18]
[9] Encrypted data has an entropy of >6.75 bits and <0.3 serial correlation.

ditionally, two (one desktop and one industrial) were still unavailable after the attack and required a power-cycle to regain network connectivity.

TCP connection exhaustion: Most networked 3D printers were designed assuming a small number of simultaneous clients (∼20). However, an attacker could create a temporary DoS condition with less than 4,000 connections on 10 of the 13 networked 3D printers. In general, the industrial printers allowed a smaller number of connections (6-65) and were thus easier to DoS; while the desktop printers allowed significantly more connections (960-4,000). Of the networked 3D printers vulnerable to TCP connection exhaustion, four did not implement a timeout for inactive TCP connections (one desktop and three industrial). Thus an attacker could slowly create a large number of connections and render the 3D printer unavailable without sending any data (e.g., only need to send SYN and ACK packets for each connection).

The more robust machines generally allowed each host a limited number of connections (10-135), less than its maximum capacity. An attacker could delay network operations (e.g., increase the time required to send printing commands over the network) but could not render the networked 3D printer unavailable.

Table 5: Network-layer DoS vulnerabilities.

3D Printer	SYN Flood	Maximum TCP Connections	TCP Timeout
Machine A	●*	1	6 seconds
Machine B	●*	1,194	340 seconds
Machine C	○	295	47 seconds
Machine D	○	998	None
Machine E	○	4,000	30 seconds
Machine F	●	33	150 seconds
Machine G	●	10	60 seconds
Machine H	●	33	150 seconds
Machine I	●	1	None
Machine J	●*	6	None
Machine K	●	4,095	60 seconds
Machine L	○	65	None
Machine M	○	10	10 seconds

○: No impact ●: DoS

* 3D printer required power cycling after DoS attack

4.2.2 Application Layer

Moving up to the application layer, we noted an assumed trust between the control PC and the 3D printer. This is evidenced by the lack of authentication between the control PC and the 3D printer. When coupled with other aspects of the 3D printer's network operations (as discussed below), this creates significant vulnerabilities.

Observation 3: ***12 of 13 networked 3D printers did not authenticate the control PC (e.g., any host on the network could send commands that the 3D printer would execute).***

Authentication: Given proprietary protocols, C3PO can only guess if the connection is authenticated by analyzing the initial data packets sent after a TCP handshake. If similar packets are repeated, it assumes this is a data exchange (e.g., status request) and no authentication occurred. We manually validated each 3D printer's network traffic to determine if authentication was utilized. Only one desktop 3D printer appeared to be using authentication and one additional desktop printer supported authenticating a subset of commands.

Observation 4: ***3 of 13 networked 3D printers execute unauthenticated commands received over the network (e.g., attacker can drive the print head into a part, Figure 4).***

Network APIs: Each networked 3D printer exposes a set of network APIs to support printing operations that fall into three categories: (1) receive files, (2) basic management (e.g., pause, abort, etc.), and (3) direct actuator commands (shown in Table 6).

Table 6: Test results for identifying encrypted data transfer on individual devices' network traffic.

Device	Receive Files	Management Commands	Actuator Commands
Machine A	✓	✓	✓
Machine B	✓	✓	
Machine C	✓*	✓†	
Machine D	✓*		
Machine E	✓	✓	✓†
Machine F	✓*		
Machine G	✓*	✓	
Machine H	✓		
Machine I	✓*		
Machine J	✓	✓	✓
Machine K	✓*		
Machine L	✓*		
Machine M	✓*		

*: Required UI action †: Required authentication

All networked 3D printers were able to receive files (e.g., printing commands) over the network, though each vendor utilized a different protocol (e.g., compressed G-code over HTTP to proprietary command format over a Windows communication foundation protocol). Additionally, some networked 3D printers provide an API for issuing management commands (e.g., pause the current printing job). An attacker could maliciously use these exposed management commands (e.g., adding delays by pausing or restarting the current printing task). Finally, a small number networked 3D printers provided direct access to 3D printer actuators (e.g., moving the print head), which could be used to cause more serious problems. Next, we show one such example.

Execute Actuator Commands: One networked 3D printer blindly executed actuator commands sent over the network,

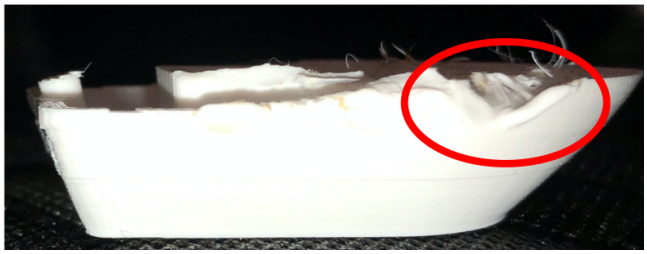
directly executing G-code commands as they arrive. This creates vulnerabilities because of two factors. First, the networked 3D printer does not authenticate the control PC sending the commands; it accepts commands from any network host. Second, the machine does not limit commands that will be executed while it is printing (e.g., moving the print nozzle into the part).

Thus, in the middle of printing a part, an attacker can send malicious actuator commands (e.g., increase heater temperature, drive print nozzle into part, etc.). The printer will perform the malicious command at its current location in the print file. This creates a safety risk that allows an attacker to cause the networked 3D printer to create an object different from what was specified (e.g., a defective part due to damage from the printer nozzle, shown in Figure 4).

To demonstrate this vulnerability, we emulated a network attacker with a goal of creating a defective part. We connected to the networked 3D printer by simply opening a TCP connection. Next, we waited until the printing task was about half-way done and sent a single command to the 3D printer.[10] This drives the 220°C printing nozzle into the part (Figure 4b), melting the plastic and creating a defective part (Figure 4c).

(a) Normal printing operation. Note separation between nozzle and boat.

(b) Attack: print nozzle impacts boat.



(c) Permanent damage to part after attack.

Figure 4: A network API that executes actuator commands while printing allows an attacker to create defects.

Observation 5: ***4 of 13 networked 3D printers had printing applications that lacked input filtering (e.g., a malformed input could crash the networked 3D printer's firmware).***

Lack of Input Filtering: While most networked 3D printers generate unique filenames at the 3D printer, one of the industrial machines had the client (control PC application) generate unique file names. The networked 3D printer would blindly save received files with their provided filename. If two files with the same filename but differing data were received, the printer firmware would crash (error message shown in Figure 5). The crash would persist across reboots, and could only be cleared by starting the machine in a "safe-mode" (where the printing application is not started) and deleting the file.

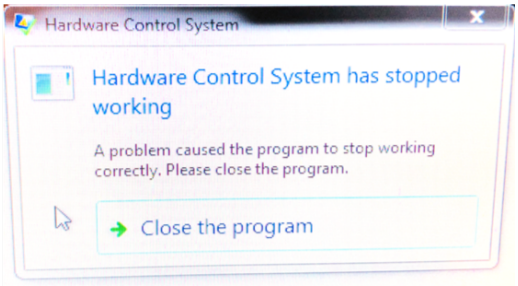


Figure 5: Error message after sending two different files with the same filename to an industrial 3D printer.

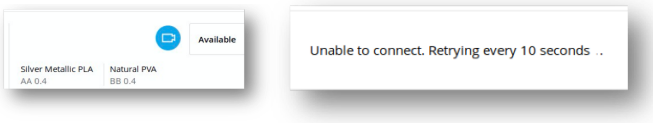
Malicious Inputs: While some 3D printer applications did not crash when given a fuzzed input, three desktop machines crashed from malformed inputs. For example, one expected an HTTP PUT request of 'GETPRINTERINFO', while slightly modifying this request by adding garbage characters to the beginning (e.g., 'GRINTERINFINFOGETPRINTERINFO') caused the machine's firmware to crash. This is similar to well-known injection attacks against webservers [52]. The crash causes the current printing task to stop, and upon power-cycling, the printing task must be restarted at the beginning. We additionally tested the surveyed networked 3D printers for susceptibility to compression bombs [39], but did not identify any vulnerabilities.

Observation 6: ***11 of 13 networked 3D printers were vulnerable to application layer DoS attacks (e.g., an attacker transmitting 1,000+ status requests simultaneously, renders the 3D printer unable to receive new print files).***

We identified three types of application layer DoS attacks: (1) stress test where a high volume of requests are sent to the networked 3D printer, (2) low-rate DoS attacks (e.g., `Slowloris`), and (3) partial file transfer where a file transfer is stopped before completion without closing the underlying connection. Similar to network layer DoS attacks, this has not been explored by prior 3D printer work.

Stress Tests: While the transport layer usually limits the maximum number of connections, three networked 3D printers (two desktop, one industrial) supported fewer simultaneous status requests (e.g., 4,000 TCP connections to 576 status requests). Further, one of these waited for the client (i.e., attacker) to terminate the connections even when data was no longer being sent, due to the printer disabling TCP timeouts.

[10] A G-code command to move the print nozzle down (e.g., `G1 Z-10.00`).



(a) Available prior to DoS (b) Unable to connect during attack

Figure 6: All networked 3D printers analyzed exhibited a DoS vulnerability, many requiring <10kbps.

Slowloris: Three industrial networked 3D printers exhibited vulnerability to a `Slowloris`-type attack [14]. These machines accept data transferred one byte per packet (allowing up to five seconds between packets), and would not process the data until all the bytes of a protocol message were received. This means a standard status request message can be used to DoS the printer for up to 45 minutes by sending only ∼290bps per connection.

Partial Data Transfer: Three networked 3D printers (two desktop, one industrial) had unique vulnerabilities when only part of a file is transferred. These machines disabled TCP timeouts when receiving a file, allowing an attacker to start but not complete multiple file transfers (some vulnerable to as few as 10). This rendered the 3D printer unavailable, which would persist as long as the attacker's TCP connections remained established without sending any data. Furthermore, one of the desktop machines required a power-cycle to recover from this attack, as the DoS continued even after the attacker closed all open TCP connections.

Table 7: Application-layer DoS vulnerabilities.

3D Printer	Minimum Bandwidth for DoS	Susceptible to `Slowloris`	Susceptible to Partial Transfer
Machine A	0.96 kbps		
Machine B	834.7 kbps		
Machine C	No DoS		
Machine D	679.6 kbps		✓
Machine E	368.1 kbps		
Machine F	2.2 kbps	✓	
Machine G	0.67 kbps	✓	
Machine H	2.2 kbps	✓	
Machine I	6.4 kbps		✓
Machine J	8.2 kbps		✓
Machine K	Not analyzed*		
Machine L	4.4 kbps	✓	✓
Machine M	No DoS		✓

* 3D printer configured as a shared network directory

Observation 7: ***10 of 13 networked 3D printers respond to a control PC's broadcast query for 3D printers (e.g., attacker can spoof a networked 3D printer to create a MitM situation where printing files could be modified in transit).***

3D Printer Discovery: In order to send printing instructions to a networked 3D printer, a control PC must first find the 3D printer on the network. Most networked 3D printers utilize an existing UDP-based, broadcast protocol (e.g., mDNS, LLMNR, SSDP, etc.) to enable zero-configuration networking. These protocols begin with the control PC sending out a broadcast query. At a minimum, these protocols provide the control PC with the hostname and IP address for each networked 3D printer. Some also include additional details in their reply (e.g., firmware version). In the event of multiple replies for the same networked 3D printer, the control PC only utilizes the first reply and drops subsequent ones.

3D Printer Spoofing: This becomes a security vulnerability as the control PC does not authenticate the 3D printer identified by its broadcast query before sending printing commands. An attacker attempting to impersonate a networked 3D printer only needs to reply to the control PC's broadcast query before the networked 3D printer. Subsequently, the attacker must imitate the 3D printer's network API, which can be as simple as a listening TCP socket. As this point, the control PC will send printing commands to the attacker thinking they are destined for the networked 3D printer, allowing the attacker to steal data or worse modifying design files (e.g., adding defects) before forwarding the file to the real 3D printer.

Observation 8: ***6 out of 13 networked 3D printers had unnecessary network services exposed (e.g., attacker could remotely control a networked 3D printer using an exposed telnet service à la [5]).***

Open Ports: Exposed network services were identified using the Nmap tool [38].[11] In order to identify if the network services were used, we analyzed the networked 3D printer's network traffic for use of these ports. The majority of 3D printers had unused, exposed TCP services, with some exposing up to 10 unused services (reference Table 8). Interestingly, we noted that in general higher cost networked 3D printers had more unused, open ports. This is likely due to the increased complexity of printing operations performed by the higher cost 3D printers. When comparing with commodity IoT, we noted fewer unused ports and a difference in the hosts contacted. Where 3D printers primarily communicated with hosts on the local network.

Observation 9: ***4 out of 13 networked 3D printers had network services vulnerable to known exploits, often from out of date libraries (e.g., an attacker could utilize a published attack to gain root access on the 3D printer [40]).***

Known Vulnerabilities: Multiple existing tools were used to perform vulnerability scans of the 3D printer's network services. These tools checked for susceptibility to Metasploit attack modules [36], Common Vulnerabilities and Exposures

[11] Machines D, F, G, H, I, & L did not use any of the top 1,000 TCP ports.

Table 8: Network services findings for individual devices.

Device	Network Services (Open / Used)	Hosts Contacted (Local / Remote)
Machine A	1 / 1	1 / 0
Machine B	4 / 1	1 / 0
Machine C	3 / 3	1 / 1
Machine D	1 / 1	1 / 0
Machine E	1 / 1	1 / 0
Machine F	5 / 1	1 / 0
Machine G	1 / 1	1 / 0
Machine H	1 / 1	1 / 0
Machine I	2 / 1	1 / 0
Machine J	10 / 1	1 / 0
Machine K	5 / 3	1 / 0
Machine L	1 / 1	1 / 0
Machine M	14 / 4	1 / 0
IoT A	0 / 0	0 / 7
IoT B	3 / 0	0 / 6
IoT C	2 / 1	1 / 0
IoT D	2 / 2	1 / 51
IoT E	0 / 0	0 / 2
IoT F	0 / 0	0 / 2
IoT G	2 / 0	0 / 15
IoT H	5 / 1	0 / 23
IoT I	0 / 0	0 / 11
IoT J	0 / 0	0 / 19

(CVEs) [24], and web server vulnerabilities [64]. Note, some scans were unable to be completed.[12] Four networked 3D printers were vulnerable to published exploits due to outdated libraries on one of their exposed network services (reference Table 9).

This becomes a security concern, as we observed a disconnect between software updates for the printing application and the supporting libraries. On some networked 3D printers, supporting libraries were not updated when the firmware was updated (e.g., no OS patches were applied). This left the networked 3D printer vulnerable to known/released exploits (e.g., WannaCry [42]).

4.3 Summary of key findings

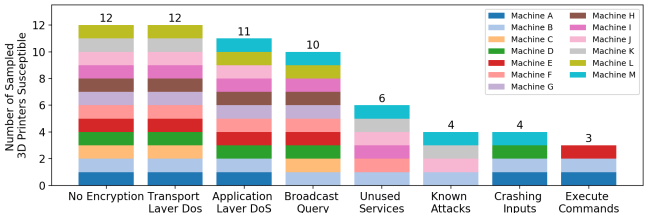


Figure 7: Summary of individual 3D printer findings, ordered by pervasiveness.

In summary, C3PO identified 33 vulnerabilities across 13 networked 3D printers evaluated (our consolidated findings

[12]The 3D printer would establish the connection but not send any replies to packets, causing the scanner to send scan packets and wait indefinitely.

Table 9: Known vulnerabilities identified per device.

Device	Metasploit attacks	CVEs	Old Libraries
Machine A	None	None	-
Machine B	None	3	FTP server
Machine C	None	1*	SSL v2
Machine D	None	None	-
Machine E	None	None	-
Machine F	Unable to run		-
Machine G	Unable to run		-
Machine H	Unable to run		-
Machine I	None	None	-
Machine J	None	2	Apache
Machine K	1	None	Remote Desktop
Machine L	Unable to run		-
Machine M	1	3	SMB
IoT A	None	None	-
IoT B	None	None	-
IoT C	1	2*	Default Credentials
IoT D	None	1*	-
IoT E	None	None	-
IoT F	None	None	-
IoT G	None	None	-
IoT H	None	None	-
IoT I	None	None	-
IoT J	None	None	-

*: CVE for supporting a weak cipher suite.

are given in Figure 7 ordered by pervasiveness of the security issue). Twelve did not use an encrypted channel (though two may send already encrypted files). Additionally, all networked 3D printers were vulnerable to DoS attacks, some resulting in the machines being unavailable until it was power cycled. Ten utilized broadcast protocols (e.g., mDNS, SSDP, LLMNR) which an attacker could spoof to create a MitM situation between the control PC and the networked 3D printer. Four had applications that were susceptible to malformed inputs, requiring a power-cycle to recover. Combinations of these vulnerabilities allowed us to perform the four attacks in Table 10 on multiple networked 3D printers.

Table 10: Attacks we demonstrated on networked 3D printers, illustrating a range of attacker goals.

		Attack			
	3D Printer	Hazard	Modify print	Crash app	DoS
Desktop	Machine A	✓		✓	✓
	Machine B		✓	✓	✓
	Machine C		✓	✓	✓
	Machine D		✓		✓
	Machine E		✓	✓	✓
Industrial	Machine F		✓		✓
	Machine G		✓		✓
	Machine H		✓		✓
	Machine I		✓		✓
	Machine J				✓
	Machine K				✓
	Machine L		✓		✓
	Machine M		✓	✓	✓

In analyzing our findings, we noted a couple of trends. As the cost of a networked 3D printer increased, there was

not a significant reduction in the number of identified vulnerabilities. A part of this is likely due to issues such as lack of encryption and susceptibility to DoS being pervasive across all networked 3D printers analyzed. We did note that the higher-cost industrial machines were more likely to be running additional services and therefore more likely to be vulnerable to published exploits (especially as the machines aged). In contrast, the desktop networked 3D printers were more likely to expose a network APIs that allowed for directly manipulating the printing actuators. Additionally, desktop machines were more likely to crash from malformed inputs.

A networked 3D printer's release year did not directly impact the number of vulnerabilities identified, with machines of different ages having a similar number of vulnerabilities. However, we noted that known best practices were least likely to be incorporated on a vendor's initial product, as these machines generally had the most vulnerabilities, regardless of its release year (e.g., a 2019 model had the most vulnerabilities). We assume this is likely due to the pressure to bring a product to market; however, with these machines likely having lifespans of 10+ years (potentially never being patched) it creates significant security risks. Next, we identify how the network deployment allow an attacker to use these vulnerabilities.

5 Network Deployment Evaluations

We evaluated 5 real-world 3D printer network deployments in order to gain an understanding of how networked 3D printers are currently deployed. This allowed us to demonstrate the benefits of C3PO as we analyzed large and complex networks.

5.1 3D Printer Deployments Evaluated

The 5 real-world 3D printer network deployments ranged from small, single 3D printer deployments (e.g., small, research-focused additive manufacturing labs) to an active makerspace with four types of networked 3D printers on multiple subnets. The five network deployments can be grouped into three deployment categories based upon their network blueprint (depicted in Figure 8):

- **Flat network.** All devices are on the same subnet.
- **Purdue Enterprise Reference Architecture [75].**[13] Networked 3D printers are on an isolated subnet, which is bridged by PCs with multiple NICs.
- **Complex.** A publicly accessible subnet, often connected to multiple subnets.

For each network deployment, the network devices identified during the network scan were placed into 4 categories: (1) networked 3D printers, (2) PCs, (3) other devices (e.g., IIoT, paper printers, building automation, etc.), and (4) network hardware. Each deployment had a large number of other devices, accounting for >41% of all the devices on each network. These devices often have weak security properties, increasing the security risks to 3D printers on the same network.

We analyzed 19 scenarios, where each scenario had a different set of assumed vulnerabilities (e.g., network hardware having code execution vulnerability [67], or other devices, such as IIoT cameras, having default credentials [5]). These scenarios were generated from a combination of prior attacks (e.g., Stuxnet using malicious USBs in order to compromise PCs connected to manufacturing networks) and discussions with operators (e.g., need for legacy systems that were added to the network). The complete list of scenarios is given in Table 11.

Each scenario was analyzed once with an attacker on the local network (e.g., an insider threat) and again with an attacker on a remote network (i.e., starting on a public network). The total number of attack paths for all attacker goals (i.e., cause a physical hazard, create defective parts, steal data, or halt operations) were normalized based upon the number of networked 3D printers and the number of devices with assumed vulnerabilities on the network, the results are shown in Figure 9. On average, C3PO identified 5 multistage attack paths to each 3D printer per insecure device on the network.

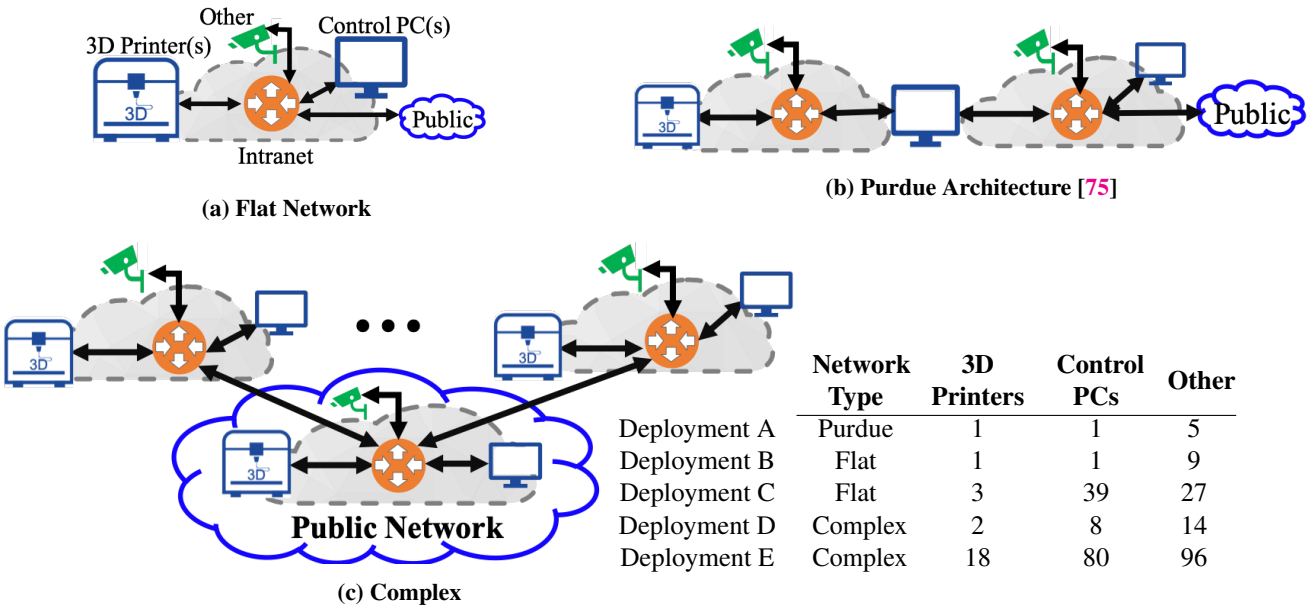
Table 11: Network security evaluation scenarios.

Device Category	Scenario	Assumed vulnerability
Baseline	0	No assumed vulnerabilities
PCs	1	Malicious USBs (e.g., [34])
	2	Malicious links (e.g., Phishing)
	3	Old OS (e.g., Windows 95)
Network hardware	4	Exploitable firmware (e.g., [67])
Other devices (e.g., IIoT)	5	Default credentials (e.g., [5])
	6	Exploitable firmware (e.g., [46])
Network hardware & PCs	7	Scenario 1 & 4
	8	Scenario 2 & 4
	9	Scenario 3 & 4
Network hardware & other devices	10	Scenario 4 & 5
	11	Scenario 4 & 6
Other devices & PCs	12	Scenario 1 & 5
	13	Scenario 1 & 6
	14	Scenario 2 & 5
	15	Scenario 2 & 6
	16	Scenario 3 & 5
	17	Scenario 3 & 6
All	18	Scenario 1, 2, 3, 4, 5 & 6

5.2 Findings

Across the network deployments analyzed we observed two trends. First, we noted a lack of network isolation. Many networked 3D printers were deployed with a large number of unnecessary and unrelated devices. Attack graphing aided our identification of how these can directly impact the security of a networked 3D printer.

[13] This is the architecture specified in ISA-95 [2].



(a) Flat Network

(b) Purdue Architecture [75]

(c) Complex

	Network Type	3D Printers	Control PCs	Other
Deployment A	Purdue	1	1	5
Deployment B	Flat	1	1	9
Deployment C	Flat	3	39	27
Deployment D	Complex	2	8	14
Deployment E	Complex	18	80	96

Figure 8: Types of real-world 3D printer network deployments, and description of those surveyed.

Observation 10: ***2 of 5 surveyed 3D printer network deployments made 3D printers easily accessible to an attacker (e.g., placing networked 3D printers on the public internet).***

Lack of Network Isolation: Most networked 3D printers were configured to be on a private network and only accessible by other devices on the same subnet. However, one network deployment placed networked 3D printers on public IP addresses. This was not required for the 3D printer's operation. Using the Censys [10] and Shodan search engines [59], 49 additional networked 3D printers from the same manufacturer were found similarly configured with publicly accessible IP addresses. This configuration allows anyone on the internet to view the networked 3D printers' camera output, as well as potentially being able to remotely stop 3D printing jobs.

Similarly, other researchers found >3,700 publicly accessible hosts running a popular web interface for 3D printers in 2018 [41]. Despite the documentation suggesting access control be enabled, many of these instances were found to not require any authentication for sending commands to the 3D printer's actuators and viewing their attached web cameras.

After discussing our findings with the manufacturing center operators, they have since modified their network deployment and removed these 3D printers from the public internet.

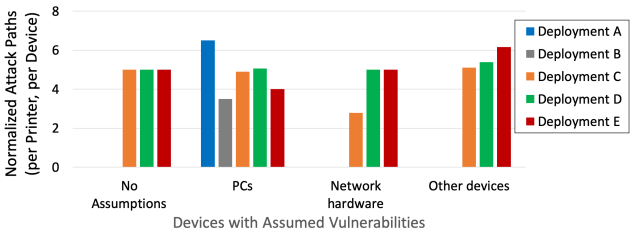
Observation 11: ***2 of 5 surveyed 3D printer network deployments had a non-network hardware device that bridged subnets. If this device had a vulnerability, it amplified the number of possible attack paths (e.g., 54% of attack paths in Deployment A required one PC be compromised).***

Devices bridging networks: In Deployment A, the 3D printer appeared to be on an isolated network. However, the control PC bridged multiple networks, some of which eventually access the internet. Thus if a remote attacker could compromise this host (e.g., using a malicious link in an e-mail [6]), it would enable her to access the 3D printer as if she were on the local network. This can be observed in our simulation data where attack paths only exist when the category with the bridging devices (generally control PCs) is assumed vulnerable. This analysis aids in identifying which devices are most critical to secure, as they are a part of the largest number of attack paths. Defenders can use this type of analysis to prioritize security efforts and resources to minimize the threats to a networked 3D printer.

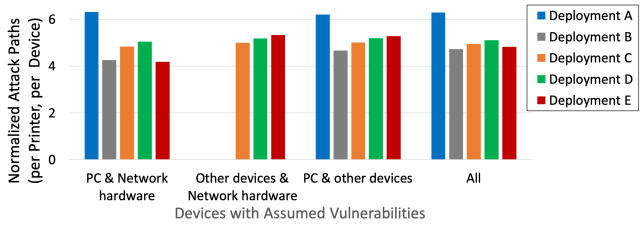
Printers attacking printers: A networked 3D printer can also be part of an attacker's multistage attack. Larger deployments often have multiple 3D printers (e.g., Deployment C). If one of these networked 3D printers is compromised by an attacker (e.g., using a published exploit) it can be used for attacking other 3D printers on the same network. This is similar to how an IIoT device could be used by an attacker to launch attacks on 3D printers. Thus, adding a new type of networked 3D printer could alter the threats posed to existing 3D printers in a manufacturing center.

5.3 Summary

C3PO demonstrated its ability to analyze real-world 3D printer deployments. All of the surveyed 3D printer network deployments were found to contain a majority of non-traditional IT devices (e.g., IIoT). C3PO was able to use theo-



(a) Normalized number of attack paths were vulnerabilities were assumed in a single device category.



(b) Normalized number of attack paths when vulnerabilities were assumed in combinations of device categories.

Figure 9: Normalized number of potential multistage attack paths for each network deployment.

retical attack scenarios to identify devices, which if compromised, result in the greatest increase in the number of possible attacks paths. We grouped the attack paths a remote attacker (e.g., on a public network) could perform based upon the vulnerabilities assumed for each category of device (depicted in Figure 9). We plot the data normalized for the number of networked 3D printers in the deployment as well as the number of devices with assumed vulnerabilities to allow for comparison between networks of different sizes (the total number of attack paths for each scenario are provided in Table 17 of Appendix B). For example, Deployment A has a maximum of 44 attack paths while Deployment E has 15,773 attack paths (both having approximately five attack paths per networked 3D printer and assumed vulnerable devices).

Across the network deployments, we noted a lack of network isolation. Deployment A provided the best network isolation for the 3D printer; where its isolation was contingent on the control PC remaining secure. However, this control PC was also connected to a network with internet access. Furthermore, it was managed independently of other IT infrastructure due to its role in supporting the networked 3D printer. Thus, the benefits of this network architecture could be negated by the management of its network deployment.

In the larger, operational deployments (C, D, and E) we noted unnecessary devices on the 3D printer's network (e.g., PCs from the business administration offices, networked conference room equipment, etc.), which could potentially become security risks to manufacturing operations. Additionally, we noted legacy devices being incorporated that were not originally intended to be networked (e.g., adding a USB-WiFi adapter to a manufacturing machine running Windows 95). These risks were further elevated when these networks were connected to publicly accessible networks. Thus having a tool such as C3PO can help inform manufacturing center operators about risks from their network deployments.

6 Recommendations and Limitations

As our study shows, today's networked 3D printers and deployments contain a number of security vulnerabilities. We focus on four high-level vulnerability categories shown in Table 12 and summarize our recommendations for each.

Table 12: Vulnerabilities in surveyed 3D printers, their security implications, and our recommendations.

Vulnerability	In	Implication	Recommendation
Vulnerable to trad. DoS attacks	13 / 13	Block access to 3D printer	Limit concurrent sessions
No encryption	12 / 13	Steal data print defects	Place 3D printer on a VPN
Bad inputs crash application	4 / 13	Stop printing operations	Drop improperly formatted inputs
Vulnerable to known exploits	4 / 13	Mod firmware, print defects	Implement patch in the network

Our goal here is to suggest *pragmatic* defenses rather than wholesale changes to the entire ecosystem or suggest draconian measures that will impact operations (e.g., avoid other IoT products or lockdown systems). Specifically, our discussions with the operators of these manufacturing centers suggest that they cannot adopt a single, fixed security solution, as they often need 3D printers from different vendors to perform different operations (e.g., printing polymers vs. metals). Thus, manufacturing centers need a flexible defense that can be tailored to the specific needs of their networked 3D printers. Additionally, as manufacturing centers increasingly incorporate new connected devices (e.g., IIoT) there is a high potential that they will be used in multistage attacks against critical assets (e.g., their 3D printers). Indeed, the increased productivity and efficiency of Industry 4.0 is predicated on incorporating these networked devices [26]. Next, we propose a possible non-intrusive defense solution that can be deployed in the network without modifying 3D printers.

6.1 Proposed Defense

We posit that the network can prevent a device's security vulnerabilities from being exploited. The network has a universal vantage point for all communications going to and from these connected devices. Defenses in the network can be deployed without modifying the connected device's operation. Additionally, network defenses do not require in-depth knowledge of vendor-specific protocols.

However, existing network defenses (e.g., firewalls) are often too coarse-grained and only deployed at the network

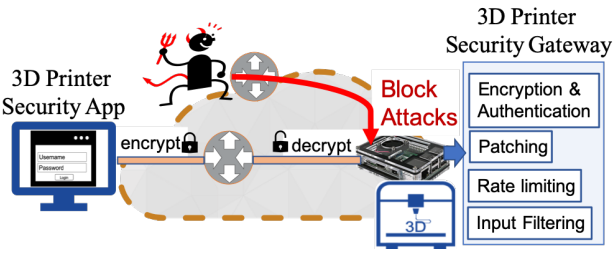


Figure 10: Conceptual overview of a security gateway system for networked 3D printer security.

perimeter. To combat these weaknesses, we propose a new network security paradigm that uses low-cost, software-defined security gateways to protect connected devices (similar to [30, 81] for wireless IoT deployments). Each connected device's traffic is routed through a security gateway, which leverages advances in software-defined networking and network function virtualization to implement agile and specific security functions for that device (à la [80, 81]).

To counter the security vulnerabilities identified by C3PO, we envision the system architecture in Figure 10, where a security gateway is used in conjunction with a security application on the control PC with the following capabilities:

- *Rate Limiting*: Traditional DoS attacks are mitigated by the security gateway limiting the number of simultaneous connections each host may have with a 3D printer.
- *Encryption and Authentication*: A VPN tunnel is set up between the control PC app and the security gateway.
- *Patching*: Known exploits are patched in the network by the security gateway running an IPS which drops traffic matching the exploit's signature.
- *Input Filtering*: Malformed inputs are dropped by the security gateway applying an input filter that only allows data payloads matching the networked 3D printer's expected protocol.

Thus, the use of security gateways could allow security to be "bolted-on" after the networked 3D printers are deployed.

6.2 Limitations

C3PO identifies network vulnerabilities in a 3D printer that an attacker could leverage; however, it is not guaranteed to identify every possible vulnerability.

- *Standalone analysis:* Fuzzing the printing application will likely not exercise the code coverage that binary analysis might achieve for a firmware image. However, some 3D printer vendors limit access to their firmware and these firmware images can be designed to run on non-x86 architectures, limiting the use of existing fuzzing tools. Additionally, as many networked 3D printers run non-standard services on non-standard ports, there is the potential for vulnerabilities to be missed by the existing vulnerability scanning tools we leverage.
- *Deployment analysis:* C3PO does not assess the security of a 3D printer's wireless networking capabilities (e.g., WiFi-specific attacks) or the client application on the control PC. Our network deployment analysis only identifies devices one hop from the networked 3D printer, potentially missing many devices an attacker could leverage or incorrectly assessing a 3D printer's network isolation. Additionally, the use of theoretical vulnerability scenarios creates the potential for false positives.

6.3 Future Work

The manufacturing domain is beginning to migrate towards manufacturing as a service, where a network of globally distributed, networked 3D printers are used for printing products at locations where 3D printers are idle or are closer to the delivery location. Companies have already deployed networks of 100+ 3D printer across multiple countries [31]. Additionally, cloud 3D printing services are connecting desktop 3D printers to a cloud service in order to allow remote management of the machines [51]. The methodology of the C3PO tool could be used for analyzing these types of network deployments.

7 Related Work

We group related work into two categories: 3D printer attacks and security assessments of networked devices.

7.1 3D Printer Attacks

Multiple works have qualitatively characterized the possible attacks on 3D printers, generating multiple attack taxonomies [49, 76, 78, 79]. Our work uses these taxonomies to specify our attacker goals and motivate the need for securing networked 3D printers. In particular, we demonstrated the vulnerability of networked 3D printers to both network and transport layer DoS attacks, which have not been shown previously. Additionally, our work provides insights about how the network deployment impacts a 3D printer's security.

Defective Parts: Beyond these taxonomies, the preponderance of research on 3D printer attacks has focused on creating defective parts. Proof of concept attacks include: injecting internal voids in the STL files on the control PC before printing instructions are sent over the network [63], modifying the 3D printer's firmware [25, 44], manipulating the in-situ feedback on the 3D printer [60], changing the part's printing orientation on the control PC [83], and replacing good parts (saved on the control PC) with ones that will fail prematurely while passing a visual inspection [6]. These works are complimentary to ours as they highlight multiple strategies an attacker could employ for creating defective parts. However, none of them leverage the network connection of the 3D printer to achieve their attack goal.

Stealing Data: ARP spoofing was used to steal data from a single vendor's networked 3D printer [22]. This work was limited to that specific vendor's protocol and only discussed stealing data over the network. Our approach is similar as it leverages a detailed analysis of the network traffic for identifying security risks. However, our work proposes a protocol-agnostic security analysis tool for identifying a broad range of security vulnerabilities, with results from analyzing 9 different vendors' networked 3D printers (covering multiple network protocols and file formats).

7.2 Security Assessments

Significant work has been done to assess the security of networked devices, looking at the manufacturing domain as well as other domains (e.g., office printers and IoT).

Manufacturing Domain: Within the manufacturing domain, numerous security assessments have been conducted. Some have focused on qualitative assessments to identify all possible security issues [21, 23, 53, 77]. Others conducted a user study evaluating if the production of defective parts would be attributed to a cyber attack [68].

Similar to our work, some of these assessments have utilized security guidelines in industry standards [74]; however, they have only looked qualitatively and not at specific devices. An experimental security analysis was performed on an industrial robot controller [56]; this work primarily focused on safety and was limited to a single device.

Orthogonal to our work, researchers have analyzed 3D printer firmware and software applications that run on the control PC [43]. These results from these analysis could be used to augment future versions of C3PO to find additional vulnerabilities.

Other Networked Devices: Researchers have also investigated the security of other networked devices. Similar to our work, a tool was developed and used for analyzing a number of office printers [45]. However, this work primarily leveraged a common language interpreted by most office printers. Networked 3D printers do not currently share a common language, requiring a different security analysis tool.

Researchers have also investigated the security of IoT, as IoT devices have gained notoriety for having security issues [16]. Most similar to our work was a survey of multiple commodity IoT devices, identifying common security issues using an amalgamation of existing network security tools [4]. However, this work focuses on individual IoT devices which are different from 3D printers (highlighted in our paper), and it did not look at the system security issues caused by different network deployments.

8 Conclusions

Our security analysis tool, C3PO, allows for systematic security evaluations of networked devices and their network deployments. We presented an example use case where we analyzed the security of 13 networked 3D printers and 5 active manufacturing network deployments. We identified 33 vulnerabilities related to lack of encryption, unpatched known vulnerabilities, crashing inputs, and multiple types of DoS. Next, we demonstrated a practical application of attack graphing for identifying potential multistage attack paths in 3D printer network deployments. Analyzing 19 simulated scenarios, we identified 3D printer on public networks, the preponderance of embedded devices in these network deployments, and the potential for 3D printers to be both targets and launch points for attacks. With the diversity and scale of networked devices in manufacturing networks, we envision that the ideal way to secure these devices is to push security into the network.

References

[1] 3Dnatives. Directory of 3d printer manufacturers. https://www.3dnatives.com/en/3d-printing-directory/categories/3d-printers-manufacturer, 2019. Accessed: 2019-05-09.

[2] P. Ackerman. *Industrial Cybersecurity*. Packt, 2017.

[3] M. A. Al Faruque, S. R. Chhetri, A. Canedo, and J. Wan. Forensics of thermal side-channel in additive manufacturing systems. *University of California, Irvine*, 2016.

[4] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose. Sok: Security evaluation of home-based iot deployments. In *2019 2019 IEEE Symposium on Security and Privacy (SP)*, Los Alamitos, CA, USA, may 2019. IEEE Computer Society.

[5] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou. Understanding the mirai botnet. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1093–1110, Vancouver, BC, 2017. USENIX Association.

[6] S. Belikovetsky, M. Yampolskiy, J. Toh, J. Gatlin, and Y. Elovici. dr0wned – cyber-physical attack with additive manufacturing. In *11th USENIX Workshop on Offensive Technology (WOOT 17)*, Vancouver, BC, 2017. USENIX Association.

[7] C3po. https://github.com/3DPrinter-Security/C3PO, 2019.

[8] T. CAMPBELL, C. WILLIAMS, O. IVANOVA, and B. GARRETT. Could 3d printing change the world? https://www.atlanticcouncil.org/images/files/publication_pdfs/403/101711_ACUS_3DPrinting.PDF, 2011. Accessed: 2019-09-10.

[9] H. Canaday. Additive manufacturing could disrupt a lot of aerospace markets. http://aviationweek.com/optimizing-engines-through-lifecycle/additive-manufacturing-could-disrupt-lot-aerospace-markets, 2018. Accessed: 2018-11-19.

[10] Censys. https://censys.io, 2019. Accessed on 2019-02-25.

[11] J. Chen, W. Diao, Q. Zhao, C. Zuo, Z. Lin, X. Wang, W. C. Lau, M. Sun, R. Yang, and K. Zhang. Iotfuzzer: Discovering memory corruptions in iot through app-based fuzzing. In *NDSS*, 2018.

[12] S. R. Chhetri, S. Faezi, and M. A. A. Faruque. Fix the leak! an information leakage aware secured cyber-physical manufacturing system. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, pages 1408–1413, March 2017.

[13] Cisco. Cisco 2018 annual cybersecurity report. https://www.cisco.com/c/dam/m/hu_hu/campaigns/security-hub/pdf/acr-2018.pdf, 2018. Accessed: 2019-09-06.

[14] Cloudflare. What is a slowloris ddos attack. https://www.cloudflare.com/learning/ddos/ddos-attack-tools/slowloris/, 2019. Accessed on 2019-02-10.

[15] L. Comi. Iot-securitychecker. https://github.com/c0mix/IoT-SecurityChecker, 2018. Accessed: 2019-05-15.

[16] G. Condra. A plea for incremental work in iot security. In *Proceedings of the 5th International Workshop on Trustworthy Embedded Devices*, TrustED '15, pages 39–39, New York, NY, USA, 2015. ACM.

[17] D. S. Correll. Travis air force base produces first certified 3d-printed aircraft parts. https://www.airforcetimes.com/news/your-air-force/2019/08/22/travis-air-force-base-produces-first-certified-3d-printed-aircraft-parts/, 2019. Accessed: 2019-09-02.

[18] CreativeTools. #3dbenchy - the jolly 3d printing torture-test by creativetools.se. https://www.thingiverse.com/thing:763622, 2015. Accessed: 2019-02-01.

[19] J. Day, R. Shepherd, P. Kearney, and R. Storer. Secure design best practices guidelines. https://www.iotsecurityfoundation.org/wp-content/uploads/2019/03/Best-Practice-Guides-Release-1.2.1.pdf, 2018. Accessed: 2019-05-02.

[20] L. DePillis. Gm is gone. now come 3d printers and robots. https://www.cnn.com/2019/03/07/economy/future-of-manufacturing-youngstown/index.html, 2019. Accessed on 2019-03-07.

[21] Z. DeSmit, A. E. Elhabashy, L. J. Wells, and J. A. Camelio. Cyber-physical vulnerability assessment in manufacturing systems. *Procedia Manufacturing*, 5:1060 – 1074, 2016. 44th North American Manufacturing Research Conference, NAMRC 44, June 27-July 1, 2016, Blacksburg, Virginia, United States.

[22] Q. Do, B. Martini, and K. R. Choo. A data exfiltration and remote exploitation attack on consumer 3d printers. *IEEE Transactions on Information Forensics and Security*, 11(10):2174–2186, Oct 2016.

[23] D. Glavach, J. LaSalle-DeSantis, and S. Zimmerman. *Applying and Assessing Cybersecurity Controls for Direct Digital Manufacturing (DDM) Systems*, pages 173–194. Springer, Cham, April 2017.

[24] G. N. GmbH. Openvas - open vulnerability assessment system. http://openvas.org, 2019? Accessed: 2019-04-03.

[25] X. Z. Hang. Security attack to 3d printing, 2013. Keynote at XCon2013.

[26] M. Hermann, T. Pentek, and B. Otto. Design principles for industrie 4.0 scenarios. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 3928–3937, Jan 2016.

[27] T. Huelsman, E. Powers, S. Peasley, and R. Robinson. Cyber risk in advanced manufacturing. https://www2.deloitte.com/us/en/pages/manufacturing/articles/cyber-risk-in-advanced-manufacturing.html, 2016. Accessed: 2019-09-06.

[28] IEC. Iec 62443: Network and system security for industrial-process measurement and control. https://www.isasecure.org/en-US/Documents/Authentication-Required-Specifications/EDSA-3-0-0/CSA-311-Functional-security-assessment-for-compone, 2018. Accessed: 2018-11-18.

[29] A. Khan and K. Turowski. A survey of current challenges in manufacturing industry and preparation for industry 4.0. In *Proceedings of the First International Scientific Conference Intelligent Information Technologies for Industry (IITI '16)*, volume 1, pages 15–26, 01 2016.

[30] R. Ko and J. Mickens. Deadbolt: Securing iot deployments. In *Applied Networking Research Workshop*, Montreal, Quebec, Canada, 2018.

[31] B. Koenig. Jabil establishes new 3d printing network. https://advancedmanufacturing.org/jabil-establishes-new-3d-printing-network/, 2019. Accessed: 2019-05-06.

[32] L. Kolodny. Elon musk emails employees about 'extensive and damaging sabotage' by employee. https://www.cnbc.com/2018/06/18/elon-musk-email-employee-conducted-extensive-and-damaging-sabotage.html, 2018. Accessed: 2019-04-12.

[33] E. Kovacs. Flaw exposes mitsubishi plcs to remote dos attacks. https://www.securityweek.com/flaw-exposes-mitsubishi-plcs-remote-dos-attacks, 2019. Accessed: 2019-08-03.

[34] D. Kushner. The real story of stuxnet. https://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet, 2013. Accessed on 2019-02-13.

[35] R. Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security Privacy*, 9(3):49–51, May 2011.

[36] S. C. LLC. Armitrage - cyber attack management for metasploit. http://www.fastandeasyhacking.com/index.html, 2019. Accessed: 2019-04-03.

[37] J. Lorincz. Cyber secure manufacturing is smart manufacturing. https://advancedmanufacturing.org/cyber-secure-smart-manufacturing/, 2018. Accessed: 2019-09-06.

[38] G. Lyon. Nmap. https://nmap.org, 2018. Accessed: 2018-11-19.

[39] C. Marie. I came to drop bombs: Auditing the compression algorithm weapons cache. https://bomb.codes, 2016. Accessed: 2019-05-14.

[40] L. Mathews. Boeing is the latest wannacry ransomware victim. https://www.forbes.com/sites/leemathews/2018/03/30/boeing-is-the-latest-wannacry-ransomware-victim/#9b1382d66344, 2018. Accessed: 2018-11-19.

[41] X. Mertens. 3d printers in the wild, what can go wrong? https://isc.sans.edu/forums/diary/3D+Printers+in+The+Wild+What+Can+Go+Wrong/24044/, 2018. Accessed: 2019-11-15.

[42] Microsoft. Microsoft security bulletin ms17-010 - critical. https://docs.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010, 2017. Accessed: 2018-11-19.

[43] S. Moore, P. Armstrong, T. McDonald, and M. Yampolskiy. Vulnerability analysis of desktop 3d printer software. In *2016 Resilience Week (RWS)*, pages 46–51, Aug 2016.

[44] S. B. Moore and W. B. Glisson. Implications of malicious 3 d printer firmware. In *Hawaii International Conference on System Sciences*, 2016.

[45] J. Müller, V. Mladenov, J. Somorovsky, and J. Schwenk. Sok: Exploiting network printers. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 213–230, May 2017.

[46] P. H. O'Neill. Russian hackers are infiltrating companies via the office printer. https://www.technologyreview.com/f/614062/russian-hackers-fancy-bear-strontium-infiltrate-iot-networks-microsoft-report/, 2019. Accessed: 2019-09-02.

[47] X. Ou, S. Govindavajhala, and A. W. Appel. Mulval: A logic-based network security analyzer. In *Proceedings of the 14th Conference on USENIX Security Symposium - Volume 14*, SSYM'05, pages 8–8, Berkeley, CA, USA, 2005. USENIX Association.

[48] OWASP. Owasp internet of things project. https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project, 2018. Accessed: 2018-11-18.

[49] Y. Pan, J. White, D. C. Schmidt, A. Elhabashy, L. Sturm, J. A. Camelio, and C. Williams. Taxonomies for reasoning about cyber-physical attacks in iot-based manufacturing systems. *IJIMAI*, 4:45–54, 2017.

[50] J. Pereyda. boofuzz: Network protocol fuzzing for humans. https://boofuzz.readthedocs.io/en/latest/, 2017. Accessed: 2019-05-08.

[51] Polar cloud. https://polar3d.com, 2019. Accessed: 2019-05-06.

[52] M. J. Pomraming. Injection flaws: Stop validating your input. https://www.blackhat.com/presentations/bh-usa-05/bh-us-05-pomraning-update.pdf, 2005.

[53] G. Pope and M. Yampolskiy. A hazard analysis technique for additive manufacturing. *CoRR*, abs/1706.00497, 2017.

[54] J. Pozzi. Airbus seeing value of 3d printed parts. https://www.mro-network.com/maintenance-repair-overhaul/airbus-seeing-value-3d-printed-parts, 2019. Accessed: 2019-05-15.

[55] Y. Pungaliya. Iotvulnerabilityscanner. https://github.com/yashpungaliya/IoTVulnerabilityScanner, 2018. Accessed: 2019-09-06.

[56] D. Quarta, M. Pogliani, M. Polino, F. Maggi, A. M. Zanchettin, and S. Zanero. An experimental security analysis of an industrial robot controller. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 268–286, May 2017.

[57] I. Research. Ibm x-force threat intelligence index 2018. https://www.ibm.com/downloads/cas/MKJOL3DG, 2018. Accessed on 2019-03-21.

[58] S. Sanfilippo. hping. http://www.hping.org, 2006. Accessed: 2019-05-10.

[59] Shodan. https://www.shodan.io, 2019. Accessed on 2019-02-13.

[60] A. Slaughter, M. Yampolskiy, M. Matthews, W. E. King, G. Guss, and Y. Elovici. How to ensure bad quality in metal additive manufacturing: In-situ infrared thermography from the security perspective. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ARES '17, pages 78:1–78:10, New York, NY, USA, 2017. ACM.

[61] C. Song, F. Lin, Z. Ba, K. Ren, C. Zhou, and W. Xu. My smartphone knows what you print: Exploring smartphone-based side-channel attacks against 3d printers. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 895–907, New York, NY, USA, 2016. ACM.

[62] J. Spadaro and L. Wyatt. Mutiny fuzzer. https://github.com/Cisco-Talos/mutiny-fuzzer, 2019. Accessed: 2019-05-03.

[63] L. D. Sturm, C. B. Williams, J. A. Camelio, J. White, and R. Parker. Cyber-physical vulnerabilities in additive manufacturing systems: A case study attack on the .stl file with human subjects. *Journal of Manufacturing Systems*, 44:154 – 164, 2017.

[64] C. Sullo and D. Lodge. Nikto2. https://cirt.net/Nikto2, 2019. Accessed: 2019-04-03.

[65] Tenable. Nessus. https://www.tenable.com/downloads/nessus, 2019. Accessed: 2019-05-03.

[66] F. M. P. Thomas R. Kramer and E. Messina. The nist rs274ngc interpreter - version 3. Technical Report NISTIR 6556, National Institute of Standards and Technology, 2000.

[67] L. Tung. Cisco critical flaw: At least 8.5 million switches open to attack, so patch now. https://www.zdnet.com/article/cisco-critical-flaw-at-least-8-5-million-switches-open-to-attack-so-patch-now/, 2018. Accessed on: 2019-02-07.

[68] H. Turner, J. White, J. A. Camelio, C. Williams, B. Amos, and R. Parker. Bad parts: Are our manufacturing systems at risk of silent cyberattacks? *IEEE Security Privacy*, 13(3):40–47, May 2015.

[69] E. Vadala and C. Graham. Downtime costs auto industry $22k/minute - survey. https://news.thomasnet.com/companystory/downtime-costs-auto-industry-22k-minute-survey-481017, 2019. Accessed on 2019-02-13.

[70] Verizon. 2016 data breach investigations report. https://enterprise.verizon.com/resources/reports/DBIR_2016_Report.pdf, 2016. Accessed: 2019-09-09.

[71] V. Visoottiviseth, P. Akarasiriwong, S. Chaiyasart, and S. Chotivatunyu. Pentos: Penetration testing tool for internet of thing devices. In *TENCON 2017 - 2017 IEEE Region 10 Conference*, pages 2279–2284, Nov 2017.

[72] J. Walker. Ent: A pseudorandom number sequence test program. http://www.fourmilab.ch/random/, 2008. Accessed: 2019-08-03.

[73] R. Wang, Y. Shoshitaishvili, C. Kruegel, and G. Vigna. Steal this movie: Automatically bypassing DRM protection in streaming media services. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*, pages 687–702, Washington, D.C., 2013. USENIX.

[74] Y. Wang, O. Anokhin, and R. Anderl. Concept and use case driven approach for mapping it security requirements on system assets and processes in industrie 4.0. *Procedia CIRP*, 63:207 – 212, 2017. Manufacturing Systems 4.0 - Proceedings of the 50th CIRP Conference on Manufacturing Systems.

[75] T. J. Williams. The purdue enterprise reference architecture. *Comput. Ind.*, 24(2-3):141–158, Sept. 1994.

[76] M. Wu and Y. B. Moon. Taxonomy of cross-domain attacks on cybermanufacturing system. *Procedia Computer Science*, 114:367 – 374, 2017. Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 - November 1, 2017, Chicago, Illinois, USA.

[77] M. Yampolskiy, T. R. Andel, J. T. McDonald, W. B. Glisson, and A. Yasinsac. Towards security of additive layer manufacturing, 2015.

[78] M. Yampolskiy, W. E. King, J. Gatlin, S. Belikovetsky, A. Brown, A. Skjellum, and Y. Elovici. Security of additive manufacturing: Attack taxonomy and survey. In *Additive Manufacturing*, volume 21, pages 431–457, May 2018.

[79] M. Yampolskiy, A. Skjellum, M. Kretzschmar, R. A. Overfelt, K. R. Sloan, and A. Yasinsac. Using 3d printers as weapons. *International Journal of Critical Infrastructure Protection*, 14:58 – 71, 2016.

[80] T. Yu, S. K. Fayaz, M. P. Collins, V. Sekar, and S. Seshan. Psi: Precise security instrumentation for enterprise networks. In *NDSS*, 2017.

[81] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, and C. Xu. Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things. In *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, HotNets-XIV, pages 5:1–5:7, New York, NY, USA, 2015. ACM.

[82] ZDResearch. Owasp-nettacker. https://github.com/zdresearch/OWASP-Nettacker, 2019. Accessed: 2019-05-15.

[83] S. Zeltmann, N. Gupta, N. Georgios Tsoutsos, M. Maniatakos, J. Rajendran, and R. Karri. Manufacturing and security challenges in 3d printing. *JOM*, 68, 05 2016.

[84] R. Zwienenberg. Acad/medre.a 10000's of autocad files leaked in suspected industrial espionage. https://www.welivesecurity.com/2012/06/21/acadmedre-10000s-of-autocad-files-leaked-in-suspected-industrial-espionage/, 2012. Accessed on 2019-02-13.

A Standards and Best Practices

Table 13: Industry standards: IEC 62443-4-2 Foundational Requirements (FR) [28].

Category	Description
IEC FR 1	Identification and Authentication Control
IEC FR 2	Use Control (e.g., Remote session termination)
IEC FR 3	System Integrity (e.g., Protection from malicious code)
IEC FR 4	Data Confidentiality (e.g., Encryption)
IEC FR 5	Restricted Data Flow (e.g., Network segmentation)
IEC FR 6	Timely Response to Event (e.g., Audit logs)
IEC FR 7	Resource Availability (e.g., DoS Protection)

Table 14: Best practices: OWASP IoT Top 10 - 2018 [48].

Category	Description
OWASP #1	Weak, Guessable, or Hardcoded Passwords
OWASP #2	Insecure Network Services (e.g., Unneeded network services)
OWASP #3	Insecure Ecosystem Interfaces (e.g., No input filtering from mobile app)
OWASP #4	Lack of Secure Update Mechanism (e.g., Unencrypted in transit)
OWASP #5	Use of Insecure or Outdated Components (e.g., Use of deprecated software libraries)
OWASP #6	Insufficient Privacy Protection
OWASP #7	Insecure Data Transfer and Storage (e.g., Lack of encryption)
OWASP #8	Lack of Device Management
OWASP #9	Insecure Default Settings
OWASP #10	Lack of Physical Hardening

Table 15: Best practices: IoT Security Foundation [19]

Category	Description
IoTSF - A	Classification of Data
IoTSF - B	Physical Security
IoTSF - C	Device Secure Boot
IoTSF - D	Secure Operating System
IoTSF - E	Application Security
IoTSF - F	Credential Management
IoTSF - G	Encryption
IoTSF - H	Network Connections
IoTSF - J	Securing Software Updates
IoTSF - K	Logging
IoTSF - L	Software Update Policy

B 3D Printer Evaluation Data

Table 16: Compilation of all findings across all 3D printers analyzed.

	Data Transfer	Network Services		3D Printer Application			Availability					
Device	**Encrypt.**	**Unused Open Ports**	**Known Vulns.**	**Bad Input**	**Compr. Bomb**	**Update**	**SYN Flood**	**TCP Conn. Flood**	**Replay Request**	**Partial Transfer**	**Stop cmd**	**Total**
Machine A	None	0	None	✓	N/A	N/A	Off-line	Only allows 1 connection			✓	7
Machine B	None	3	3	✓	N/A	N/A	Off-line	Off-line for 340 seconds >1,194 conns	Off-line indefinitely >1,169 conns	No impact	✓	8
Machine C	None	0	1	None	No impact	N/A	Off-line	Significnat latency >135 connections			No	6
Machine D	Unlikely	0	None	✓	No impact	No encryp	No impact	Off-line indefinitely >998 connections		Off-line indefinitely >520 conns	No	5
Machine E	None	0	None	None	No impact	No encryp	No impact	Off-line for 40 seconds >4,000 conns	Off-line indefinitely >576 conns	Off-line indefinitely >767 conns	✓	5
Machine F	None	4	Unable to run	None	N/A	No encryp No Auth	Off-line	Off-line after 33 connections			No	6
Machine G	None	1	Unable to run	None	N/A	N/A	Off-line	Off-line after 10 connections			No	6
Machine H	None	0	Unable to run	None	N/A	N/A	Off-line	Off-line after 33 connections			No	5
Machine I	None	1	None	None	N/A	N/A	Off-line	Off-line after 6 connections			No	6
Machine J	None	8	2	Not tested	N/A	N/A	Off-line	Only allows 6 simultaneous connections			No	7
Machine K	N/A	4	1	N/A	N/A	N/A	Off-line	Not analyzed			No	3
Machine L	None	0	Unable to run	None	N/A	N/A	No impact	Off-line after 65 connections			No	4
Machine M	Unlikely	12	3	✓	N/A	Weak encryp	No impact	No impact	Off-line for 10 seconds >10 conns	Off-line indefinitely >10 conns	No	5
Total	12	7	5	4	0	0	9	11	12	11	3	73

Table 17: Details on all findings across 3D printer deployments analyzed.

Scenario	Deployment A						Deployment B						Deployment C			Deployment D			Deployment E		
	Data Exfil		Defects		DoS		Data Exfil		Defects		DoS		Data Exfil	Defects	DoS	Data Exfil	Defects	DoS	Data Exfil	Defects	DoS
0	2	0	3	0	3	0	1	0	2	0	2	0	2	4	4	3	6	6	18	36	36
1	3		5		5		1		3		3		12	30	30	117	237	237	1,404	2,862	2,862
2	3		5		5		1		3		3		12	30	30	117	237	237	1,404	2,862	2,862
3	3		5		5		1		3		3		12	30	30	117	237	237	1,404	2,862	2,862
4	4	0	6	0	6	0	2	0	4	0	4	0	6	12	12	9	18	18	54	108 98	108 98
5	4	0	6	0	6	0	8	0	16	0	16	0	23	64	64	81	162	162	1,710	3,420	3,420
6	4	0	6	0	6	0	8	0	16	0	16	0	23	64	64	81	162	162	1,710	3,420	3,420
7	6		10		10		3		7		7		16	38	38	123	249	249	1,458	2,975 2,970	2,975 2,970
8	6		10		10		3		7		7		16	38	38	123	249	249	1,458	2,975 2,970	2,975 2,970
9	6		10		10		3		7		7		16	38	38	123	249	249	1,458	2,975 2,970	2,975 2,970
10	6	0	9	0	9	0	9	0	18	0	18	0	32	64	64	84	168	168	1,746	3,497 3,492	3,497 2,392
11	6	0	9	0	9	0	9	0	18	0	18	0	32	64	64	84	168	168	1,746	3,497 3,492	3,497 2,392
12	7		10		10		8		17		17		42	90	90	195	393	393	3,096	6,246	6,246
13	7		10		10		8		17		17		42	90	90	195	393	393	3,096	6,246	6,246
14	7		10		10		8		17		17		42	90	90	195	393	393	3,096	6,246	6,246
15	7		10		10		8		17		17		42	90	90	195	393	393	3,096	6,246	6,246
16	7		10		10		8		17		17		42	90	90	195	393	393	3,096	6,246	6,246
17	7		10		10		8		17		17		42	90	90	195	393	393	3,096	6,246	6,246
18	10		17		17		10		21		21		42	90	90	201	405	405	3,132	6,318 6,323	6,318 6,323