International Journal of Computer Integrated Manufacturing

ISSN: 0951-192X (Print) 1362-3052 (Online) Journal homepage: https://www.tandfonline.com/loi/tcim20

# Development of testbed for cyber-manufacturing security issues

Mingtao Wu, Jinwoo Song, Snehav Sharma, Jupeng Di, Benliu He, Ziming Wang, Jingkai Zhang, Long Wang Lucas Lin, Emily Ann Greaney & Young Moon

**To cite this article:** Mingtao Wu, Jinwoo Song, Snehav Sharma, Jupeng Di, Benliu He, Ziming Wang, Jingkai Zhang, Long Wang Lucas Lin, Emily Ann Greaney & Young Moon (2020) Development of testbed for cyber-manufacturing security issues, International Journal of Computer Integrated Manufacturing, 33:3, 302-320, DOI: 10.1080/0951192X.2020.1736711

**To link to this article:** https://doi.org/10.1080/0951192X.2020.1736711

Published online: 15 Mar 2020.

INTERNATIONAL JOURNAL OF COMPUTER INTEGRATED MANUFACTURING
2020, VOL. 33, NO. 3, 302–320
https://doi.org/10.1080/0951192X.2020.1736711

Taylor & Francis
Taylor & Francis Group

ARTICLE



# Development of testbed for cyber-manufacturing security issues

Mingtao Wu, Jinwoo Song, Snehav Sharma, Jupeng Di, Benliu He, Ziming Wang, Jingkai Zhang, Long Wang Lucas Lin, Emily Ann Greaney and Young Moon

Department of Mechanical and Aerospace Engineering, Syracuse University, Syracuse, NY, USA

**ABSTRACT**

Future manufacturing systems such as Industry 4.0, Cloud Manufacturing, or Cyber-Manufacturing System (CMS) rely on the seamless integration of cyber and physical domains in the manufacturing systems. However, this increased connectivity within a factory and between factories enlarges the attack surface for malicious individuals. Potential intrusions – especially cyber-physical attacks – can cause profound damages to the manufacturing system. In order to detect, prevent and overcome the cyber-physical attacks, an experimental environment to investigate cyber-physical attacks in CMS was developed. This paper presents details of a testbed, called Cyber-Manufacturing System Security Testbed (CSST). It was designed to investigate cyber-physical intrusions and validate their detection methods in the CMS environment. Cyber and physical components of the CSST are described along with how they interact and are constructed to address the security issues. Results from using the CSST at both component and system levels are analyzed and presented.

**ARTICLE HISTORY**
Received 6 June 2019
Accepted 20 February 2020

**KEYWORDS**
Cyber-Manufacturing System; security testbed; cyber-physical attacks; intrusion detection

## 1. Introduction

Cyber-Manufacturing System (CMS) is the future of the connected factories (Song and Moon 2016a). Drawing from related concepts such as Industry 4.0, Cloud Manufacturing, or Industrial Internet, the physical domain of CMS is fully integrated with its cyber domain (Song and Moon 2016b). The connectivity not only enables direct communications between various components of CMS (Yang and Takakuwa 2017) but also derives numerous benefits such as mass customization, access to resources around the globe, and sustainable manufacturing. However, the very connectivity enlarges the attack surface of the manufacturing systems. Particularly in CMS, cyber-attacks (Wu, Song, and Moon 2019) can damage physical components and compromise workers' safety.

To detect cyber-physical attacks in CMS, an intrusion detection system (IDS) framework – DACDI (Define, Audit, Correlate, Disclose, and Improve) – has been developed (Wu and Moon 2017a). To validate IDS, data collected from real or realistic manufacturing systems are required. Since a fully developed CMS does not exist yet, it is not possible to acquire such data from a real environment. Also, unlike in the cybersecurity domain, standard benchmark datasets for cyber-physical attacks in manufacturing are not available yet.

As a result, a realistic CMS testbed can enable the investigation of security issues without building a full-scale CMS nor interrupting production systems' operation. The testbed presented in this paper is composed of physical and computational components. The physical components include equipment, controllers, sensors, and actuators at the component level in a CMS shop floor, whereas the computational components include web interfaces, IDS, and simulation models at the system level of CMS. These components are used to collect data for intrusion detection analyses. The testbed can execute customer orders, job allocation, manufacturing, post-processing, conveying, and transportation. From the testbed, necessary data have been collected and analyzed for intrusion detection studies.

The remainder of the paper is organized as follows. Section two presents the background of the CMS security. Motivations for building the testbed are explained in section three. Detail settings of the testbed are presented in section four. Attack scenarios and a case study are presented in section five. Section six discusses the conclusion and future work.

**CONTACT** Young Moon ybmoon@syr.edu Department of Mechanical and Aerospace Engineering, Syracuse University, 263 Link Hall, Syracuse, NY 13244, USA

## 2. Background

As factories are evolving toward to the vision of CMS, new and not-well-understood cyber-physical attacks are on the rise. The IBM reported that the manufacturing industry had the highest number of confirmed security incidents among all industry sectors during 2016: almost 40% higher than the average across all industries (IBM 2017).

Compared to network security, cyber-manufacturing security has differences in attack vectors, network protocols, complexity, and attack consequences. While an attack in the network environment can exploit data integrity, confidentiality, and availability only, an attack in the CMS environment can also damage physical equipment and product, or compromise production safety.

### 2.1. Cyber-physical attacks

The cyber-physical attack is defined as follows: 'The attacks initiate inside or outside CMS environment as digital format and intrude via cyber, causing physical components such as machines, equipment, parts, assemblies, products have over wearing, breakage, scrap or any other change that original design does not intend to be.' (Wu, Song, and Moon 2019) In the current manufacturing industry, cyber-physical attacks are not fully exposed yet, but start showing their potential disastrous consequences.

One of the most notorious cyber-physical attack incidents is the Stuxnet worm on Iranian centrifuges in 2010 (Langner 2011). The Stuxnet targeted specific types of programmable logic controllers (PLCs), collected information on industrial systems and caused the fast-spinning centrifuges to tear themselves apart.

The first cyber-physical attacks in manufacturing were confirmed in 2014. Multiple hackers used phishing email with malicious attachment to gain access to the industrial control system in a German steel mill. The attack compromised the blast furnace control system, making it unable to shut down by their employees, and ultimately caused significant damage (Lee, Assante, and Conway 2014).

In 2017, the WannaCry ransomware (Ehrenfeld 2017) affected the car manufacturer – Dacia (owned by French Renault) in Romania – and caused the company to temporarily stop production at several sites to prevent the spread of the attack (Lab 2017). In 2018, Boeing production plant in Charleston, South Carolina, USA was also attacked by WannaCry ransomware (Gates 2018). In 2018, a robotics supplier – Level One Robotics – leaked sensitive data belonging to over 100 manufacturing companies from an online backup server (Stacy 2018).

Cyber-physical attacks have been simulated and investigated. Sturm et al. (2014) explored the cyber-physical attacks in additive manufacturing. A case study of the malicious void attack on 'STL' file was demonstrated. In real life, the attack may be carried out by malware or malicious users who upload defective design online. Sturm et al. (2017) proved the void attack reduced yield load by 14%, and the test specimen fractured at the void location. Yampolskiy et al. (2016) analyzed the potential for weaponizing additive manufacturing equipment to cause cyber-physical attacks. An example by Belikovetsky et al. (2017a) showed that reducing the fatigue life of a 3D print drone propeller can cause sabotaged parts breaking up during a flight. Turner et al. (2015) presented an experiment from the social and human perspectives with seven groups of engineering students. With the goal of testing the students' awareness of the cyber-physical attack on a G-code file, the results showed that none of the groups were aware that the G-code file or manufacturing part was changed.

To characterize cyber-physical attacks, taxonomies were developed with different perspectives. Wu developed taxonomies on cyber-physical attacks in the manufacturing system from both attack (Wu and Moon 2017b) and detection (Wu and Moon 2018) perspectives. Yampolskiy et al. (2018) proposed taxonomies for attacks on additive manufacturing. Pan et al. (2017) proposed a taxonomy on possible types of cyber-physical attacks against manufacturing processes in the Internet of Things environment.

### 2.2. Intrusion detection of cyber-physical attacks

In computer and network security, there are two types of IDS: host-based intrusion detection system (HIDS) and network-based intrusion detection system (NIDS). It typically takes several months to detect and longer to remediate (Minnick 2016) an attack using IDS. Furthermore, IDSs are not designed for cyber-physical attacks. These limitations are critical for manufacturing system since (i) the physical components

cannot be backed up and the production period is shorter than detection period and (ii) there is a long-term effect on manufacturing systems: defective products can cause a larger scale of damages among customers and across the supply chain.

Physical data have been used to detect cyber-physical attacks in manufacturing systems (Wu, Song, and Moon 2019). Wu et al. (2016) used image data to detect malicious defects during additive manufacturing process and acoustic data to detect a dimensional change in the subtractive process. Vincent et al. (2015) detected unintended part changes with structural health monitoring (SHM) data. Chhetri, Canedo, and Al Faruque (2016) used analogue emission to detect kinetic cyber-physical attacks on additive manufacturing. Belikovetsky et al. (2017a) detected a cyber-physical attack on additive manufacturing with digital audio signals. Moore et al. (2017) were able to detect sabotage attacks in additive manufacturing with power consumption analysis. Above investigations illustrate that physical data can effectively detect the malicious changes in manufacturing processes.

To address the system level intrusion detection, a framework – DACDI – was developed to systematically address cyber-physical attack detection in CMS (Wu and Moon 2017a). Key features of DACDI are utilization of both cyber and physical data for alerts, correlation of alerts, aggregation of meta-alerts, and false alarm reduction.

## 3. Problem and motivation

Three methods can evaluate the effectiveness of an intrusion detection system: (i) benchmark dataset, (ii) computer simulation, and (iii) physical testbed (Qassim et al. 2018). These methods for CMS security domain are discussed and the need for building a testbed for security research on CMS is explained in this section.

### 3.1. Benchmark dataset

The benchmark datasets are commonly created by launching attacks under a specific environment to test the ability of the IDS to detect the manifestations of the attacks (Valeur et al. 2004). For example, in computer security, DARPA/MIT Lincoln Laboratory datasets were created in 1998 and 1999 (Lippmann et al. 2000), and are widely used as a benchmark to test and evaluate new IDS methods (Mahoney and Chan 2003).

However, for CMS, the benchmark datasets to evaluate the intrusion detection systems are not available because (i) the cyber-physical attacks are new, (ii) current manufacturing systems are not designed to monitor or detect cyber-physical attacks, and (iii) visions such as Cyber-Manufacturing System, Industry 4.0 are not fully realized yet.

### 3.2. Computer simulation

Some investigations addressed cyber-physical attacks with computer simulation. Brachoa et al. used a discrete event simulation package to model a manufacturing company under cyber-physical attack to assess the performance changes under cyber-physical attacks (Bracho et al. 2018). Three metrics – orders out, work-in-process level and customer lead time – were used to evaluate the performance. The game theory was used to imitate behaviours between a cyber-attacker and a defender.

However, the computer simulation is limited in cyber-physical attack detection study compared to real-world experiment. First, computer networks and cyber-attackers are unpredictable compared to a manufacturing system. New cyber-physical attack adversaries are nearly unknown due to limited existing cases. Each case is different from the other, making it difficult to model a true attacker behavior. Moreover, essential details for intrusion detection analysis such as physical damage, defects, emission, and power consumption are difficult to model in a system level computer simulation.

Although computer simulations are useful in reducing the experimental cost and safe to experiment with attack scenarios, it is limited in providing real-world information for cyber-physical attack detection in CMS.

### 3.3. Physical testbed

A physical testbed is a platform where scientific experiments can be conducted and new theories can be validated. Compared to digital simulation, it provides a realistic cyber-physical environment that exhibits all physical laws and phenomena.

There are some examples of testbeds, but for different purposes. For example, the National Institute of

Standards and Technology (NIST) designed and is developing Smart Manufacturing Systems (SMS) testbed (NIST 2016) for smart manufacturing research and development across the product lifecycle [36,37]. Compared with the proposed testbed, the SMS from NIST does not focus on security research in the manufacturing system. Particularly, the cyber-data and physical-data types are not comprehensive enough for cyber-physical attack intrusion detection research.

The cyber-physical smart grid research community has established large-scale physical testbeds such as in (i) Idaho National Laboratory (INL), (ii) National Renewable Energy Laboratory (NREL), (iii) Illinois Institute of Technology (IIT) Microgrid, and (iv) Energy Systems Research Laboratory (ESRL). Compared with the presented testbed, these testbeds are not for manufacturing.

There are security-oriented testbeds for other types of systems: (i) Virtual Control System Environment (VCSE), (ii) Testbed for Analysing Security of SCADA Control System (TASSCS), (iii) SCADA Security Laboratory and Power and Energy Research Laboratory, (iv) DeterLab, and (v) Virtual Power System Testbed (VPST) (Cintuglu et al. 2017). Compared with the presented testbed, these testbeds are not designed to support security research in manufacturing.

Although CMS, Industry 4.0 or Smart manufacturing have not been fully established, the core-technologies such as additive manufacturing, Internet of Things (IoT), cloud computing or machine learning are already commonly used. Integrating available CMS components into a testbed can help addressing the limitations found from all three methods. It can not only provide an environment for researchers to explore various cyber-physical security issues, but also generate benchmark datasets.

## 4. CSST: Cyber-Manufacturing System Security Testbed

A Cyber-Manufacturing System Security Testbed (CSST) has been developed specifically for intrusion detection, correlation, and prevention investigations in CMS. It is a testbed to represent the manufacturing systems that are vulnerable to cyber-physical attacks. Common cyber-attack vectors and equipment in a manufacturing system are built into the testbed. The testbed can provide a platform to collect cyber and physical data for manufacturing security research.

### 4.1. Requirements

Following requirements are desirable for such a testbed:

(i) The ability to simulate the CMS physical processes with a simple minimum setup to reduce the set-up cost and attack damage cost.
(ii) The ability to simulate the CMS network environment with the most common and basic network Transmission Control Protocol/Internet Protocol (TCP/IP) setup, with the potential to expand or replace it with more advanced protocols such as MTConnect.
(iii) The ability to collect cyber data from the network and host-based intrusion detection software such as Snort and OSSEC.
(iv) The ability to collect physical data such as image, acoustic, acceleration, or power consumption data for intrusion detection analysis, with the potential to expand the testbed with additional sensors if necessary.
(v) The ability to simulate cyber-physical attacks within a manufacturing system and over a supply chain (National Research Council 2014).
(vi) The ability to integrate cyber-attack protection mechanisms in the testbed.

### 4.2. Schematics

The CSST consists of six major components: (i) discrete event computer simulation system, (ii) cyber environment for customer web service (iii) physical manufacturing process and equipment, (iv) control system, (v) network communication system, and (vi) monitoring system. The current CSST has a Supplier-Testbed and a Demander-Testbed.

As shown in Figure 1, the computer simulation can provide randomized customer and attacker arrival schedules. The researcher can play the role of a customer or an attacker based on the randomized job schedule, place an order, or penetrate the customer database. The order can be fabricated within a single testbed or between the Supplier-Testbed and the Demander-Testbed. The flow in Figure 1 illustrates a scenario where a part is first fabricated in the Supplier-Testbed, transferred to the Demander-Testbed, then assembled with another part that is fabricated by the Demander-Testbed.

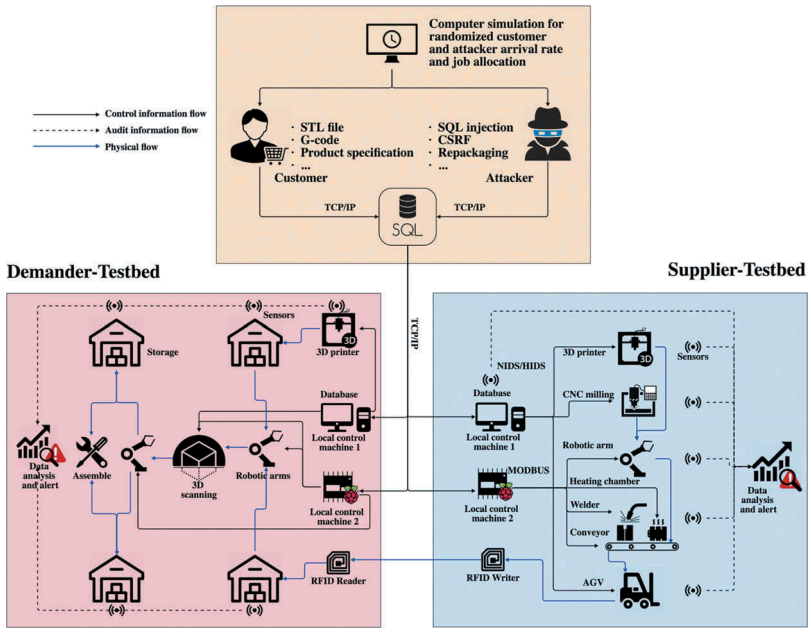



Figure 1. An attack scenario diagram in CSST.

The testbed monitoring system collects cyber-data such as network activities and host log, along with physical-data such as image, acoustic, acceleration, temperature, power consumption, and part dimensions into the database for intrusion detection analysis. Figure 2 shows the Supplier-Testbed and the Demander-Testbed that are located separately by four stories in a building.

### 4.3. Computer simulation

The computer simulations are built with a discrete event simulation and automation software – ARENA. Modules used in the simulation models are customer arrival, attacker arrival, website front end, database, order distribution, manufacturing processes, and attack and detection.

As shown in Figure 3, three suppliers are candidates for upcoming orders. The testbed plays the role of supplier #1 with the detection ability. As a result, the computer simulation can provide randomized data to the physical testbed such as customer arrival timing and attacker arrival timing. On the other hand, the testbed can generate data and feed them into the simulation model: processing time, downtime, conveying speed to the testbed, and the detection rates. In this way, the real data can be utilized for many times and numerous replications of experiments.

### 4.4. Cyber environment

The cyber environment consists of a customer front end website built using PHP 5.0 web programing language and Apache HTTP Server 2.4 along with user database based on MySQL. The customer can log in the website, upload design, and place an order.

### 4.5. Physical environment

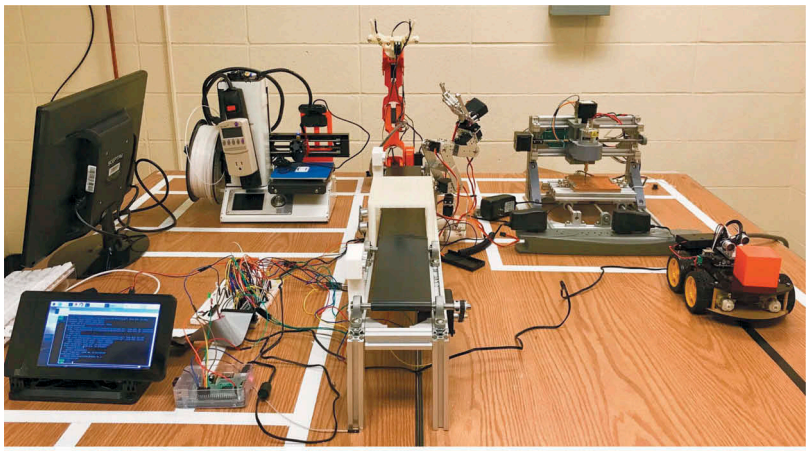
The physical portion of the testbed consists of two 3D printers, a computer numerical control (CNC) milling machine, four robotic arms, a conveyor, a heating chamber, an Automated Guided Vehicle (AGV), 3D scanner, and radio-frequency identification (RFID) reader/writer.

#### 4.5.1. 3D printer

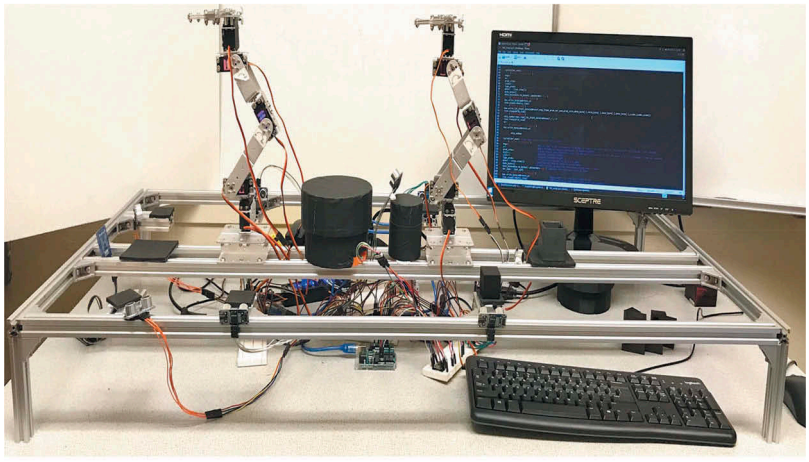
An MP Select Mini 3D printer V2 represents the additive manufacturing process. The machine is capable of producing a part under the dimensions of 120 x 120 × 120 mm with acrylonitrile butadiene styrene (ABS) or polylactide (PLA) material. The machine can print 'STL' files via a connected Windows 10 desktop machine with Cura 3D printing software.

#### 4.5.2. CNC milling machine

A three-axis CNC milling machine represents the subtractive manufacturing process. The machine can read



(a) Supplier-Testbed setup



(b) Demander-Testbed setup

**Figure 2.** CSST setup (a) Supplier-testbed setup. (b) Demander-testbed setup.

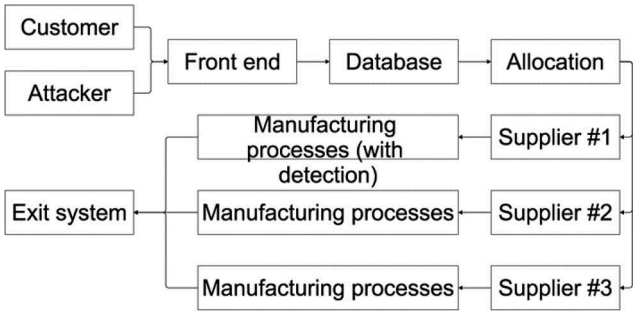



**Figure 3.** Arena computer simulation model.

'G-code' file from same desktop machine with Grbl Controller 3.0 software.

#### 4.5.3. Robotic arm for carrying/moving

An Arduino Braccio six degree of freedom robotic arm is integrated into the CSST for carrying sample objects from a storage area to a conveyor. The robotic arm executes pre-defined code in Arduino UNO R3 micro-controller. The Arduino UNO is connected to a Raspberry Pi as the control machine.

#### 4.5.4. Conveyor

Custom-built conveyor transfers sample objects from the welding station and to a heat treatment station. The conveyor is powered by a step motor controlled by the same Raspberry Pi.

#### 4.5.5. Robotic arm for welding

A custom-built robotic arm is used to simulate the welding process. The robotic arm has six degrees of freedom with the hand attached to a marking pen. The pen mimics the welding process. Similarly, the welding robotic arm executes pre-defined codes in Arduino UNO R3 micro-controller connected to Raspberry Pi as the control machine.

#### 4.5.6. Heat treatment

A heating chamber is incorporated for simulating the heat treatment process. The heating chamber consists of a tunnel that encloses the heating environment, an ultrasonic sensor that detects sample object arrivals, and three heating elements that are controlled by Raspberry Pi.

#### 4.5.7. Transporter AGV

The testbed integrates an AGV as a transportation device. It carries sample objects from a conveyor to the packaging area. The AGV is self-controlled by an Arduino UNO connected to the local control machine.

#### 4.5.8. 3D scanner and turntable

The 3D scanner and turntable check the part dimensions. The part is placed on a turntable and rotates slowly for 360 degrees. At the same time, the Kinect 360 scanner records the process and creates an STL file that can be used for comparing with the original design file.

#### 4.5.9. Robotic arm for assemble

The robotic arm for assembling is the same model type as the robotic arm for welding, with total of six servo motors that are integrated into the arm for multiple axis movements. The robotic arm can accomplish assembling tasks.

#### 4.5.10. RFID reader/writer

The RDIF read/write function utilizes Mifare RC522 sensor module and NTAG 215 NFC sticker. The near-field communication (NFC) sticker is thin and compact in size with 540 bytes of memory. Information such as part ID, customer ID can be stored and carried with the parts.

### 4.6. Control system

There are two local computers for each testbed: (i) a Windows 10 based desktop PC connected to the Ethernet and (ii) a Linux-based raspberry Pi 3 microcomputer. The Supplier-Testbed uses OpenPLC (Alves et al. 2014) software with Raspberry Pi to control the conveyor. All the connections are currently wired but with wireless capability.

#### 4.6.1. Control machine for CNC and 3D printer

The Windows 10 based desktop PC controls the 3D printer and CNC milling machine. The 3D printer control requires Cura open-source 3D printer slicing software. In Cura, 3D printing settings such as nozzle temperature, printing speed, layer height can be modified. The CNC milling machine control requires the Grbl Controller software sending 'G-code' to the machine. In the Grbl Controller, milling parameters such as feed speed and spindle speed can be changed.

#### 4.6.2. OpenPLC for conveyor

The integration of raspberry Pi 3 and OpenPLC software is an open-source alternative to the Programmable Logic Controller (PLC). The OpenPLC directly controls the step motor that powers conveyor, or the Arduino UNO micro-controllers in the robotic arms.

### 4.7. Communication

The testbed utilizes an ethernet-based communication control system. The local control machines – the Windows 10 based desktop machine or Linux-based raspberry pi – are connected to the Internet for the purpose of accessing the CMS database via TCP/IP communication protocol. Within the testbed, Modbus is used between Raspberry Pi and its connected actuators.

The reasons for adopting TCP/IP and Modbus communication protocol in the testbed are: (i) their popularity in today's manufacturing systems, (ii) the abundant resources of available network monitoring systems, and (iii) the not-yet-realized communication protocol standardization of future manufacturing applications. Regardless of which type of protocols is chosen, the proposed methodology of intrusion detection on cyber-physical attacks should function the same.

#### 4.7.1. TCP/IP

The connection between customer and website frontend, database, and local control machines is the Ethernet over TCP/IP protocol. TCP/IP refers to the Transmission Control Protocol and Internet Protocol. The TCP/IP is one of the protocols nearly all firms use currently (Boyle and Panko 2013).

#### 4.7.2. Modbus

Modbus is a free and open-source protocol that was developed by Modicon for PLCs. As the OpenPLC with raspberry Pi 3 has the capability of connecting to the Ethernet over TCP/IP, it was implemented to support the

MODBUS TCP/IP protocol (Alves et al. 2014). The Modbus TCP/IP is the Modbus protocol with a TCP interface that runs on the Ethernet (Goldenberg and Wool 2013).

#### 4.7.3. I2C

The connection between the Raspberry Pi and the robotic arm controller Arduino UNOs is via I2 C protocol. It is a multi-master protocol that virtually any number of slaves and any number of masters can be connected and communicate between each other on two signal lines (Leens 2009).

### 4.8. Auditing

The audit data is the most important part of an intrusion detection system. In the DACDI (Wu and Moon 2017a) framework, both cyber and physical data need to be collected. The physical data analysis is the unique feature of intrusion detection of cyber-physical attacks. The cyber data are capable of detecting known attacks and used as evidence to correlate with physical anomalies. The physical data are capable of detecting cyber-physical data quickly with high accuracy (Wu, Song, and Moon 2019; Song et al. 2016), and can also prevent machine malfunction and human mistakes.

#### 4.8.1. Cyber data auditing

The cyber audit data includes data from network activities and hosts. Snort – network-based intrusion detection system (NIDS) software – is used to utilize network activity log data such as login attempts, network connections, or every data packet that appears on the wire (Kemmerer and Vigna 2002). A packet can monitor network traffic in real-time on local control machines and database host of the testbed. The standard rules are used to check abnormal data in packet traffic (Khamphakdee, Benjamas, and Saiyod 2014).

OSSEC – host-based intrusion detection system (HIDS) software (Timofte 2008) – is used to monitor host activities on local control machines and database hosts. It analyzes host log, file, windows registry, and provides real-time alert responses.

#### 4.8.2. Physical data auditing

The physical data are collected from manufacturing machines and equipment in the testbed. At least two types of sensors are used on each machine and process for security and alert accuracy.

To illustrate, sample data collected from the CSST with analysis method such as feature extraction for machine learning are presented below.

**4.8.2.1. Power consumption data from power metre.** The power consumption data is recorded by a Kill-A-Watt P4400 power meter. The malicious and legitimate data are created by printing a malicious infill defect (Wu et al. 2016) that can weaken the part's structure (Sturm et al. 2017).

To analyze the power consumption data of 3D printing process as shown in Figure 4, the window time of feature extraction is set for every 100 s. During the window time period, the mean value, standard deviation, maximum, medium, minimum, skewness, kurtosis, and the number of power data points over 80, 82 and 85 kWh are calculated as features. In total, 10 features are used. In Figure 4, the unit for y-axis is kWh whereas the unit for x-axis is minute.

**4.8.2.2. Image data from the camera.** The images for 3D printing process sample part infill and welding process quality were taken by two Logitech C310 and C525 cameras. The greyscale value was used for data analysis.

The images collected from the welding mark and 3D printing infill are shown in Figure 5 (a) and (b). Each image as a dataset can be divided vertically into eight equal areas. In each area, the greyscale's mean value, standard deviation, and the number of pixels over the grayscale threshold are obtained. As shown in Figure 5 (c), the number of pixels over the grayscale threshold between malicious and legitimate images varies significantly.

**4.8.2.3. Acceleration data.** The accelerometer installed on CNC milling machine and AGV are monitoring the dynamic activity. They are MMA7361 accelerometer sensors with a sampling rate of 115,200 bauds, controlled by Arduino UNO microcontroller. Figure 6 shows the acceleration data at the first 120 s with legitimate and malicious settings while changing the feed speed.

For acceleration data analysis, the window time for feature extraction is set to 1.5 s when the sensor collects around 108 acoustic signals per second. During every 1.5 s, the acceleration mean, standard deviation,

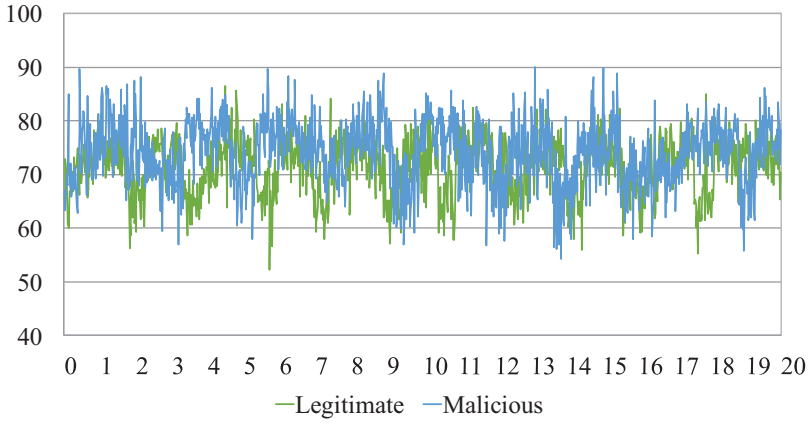



Figure 4. Power consumption data analysis.

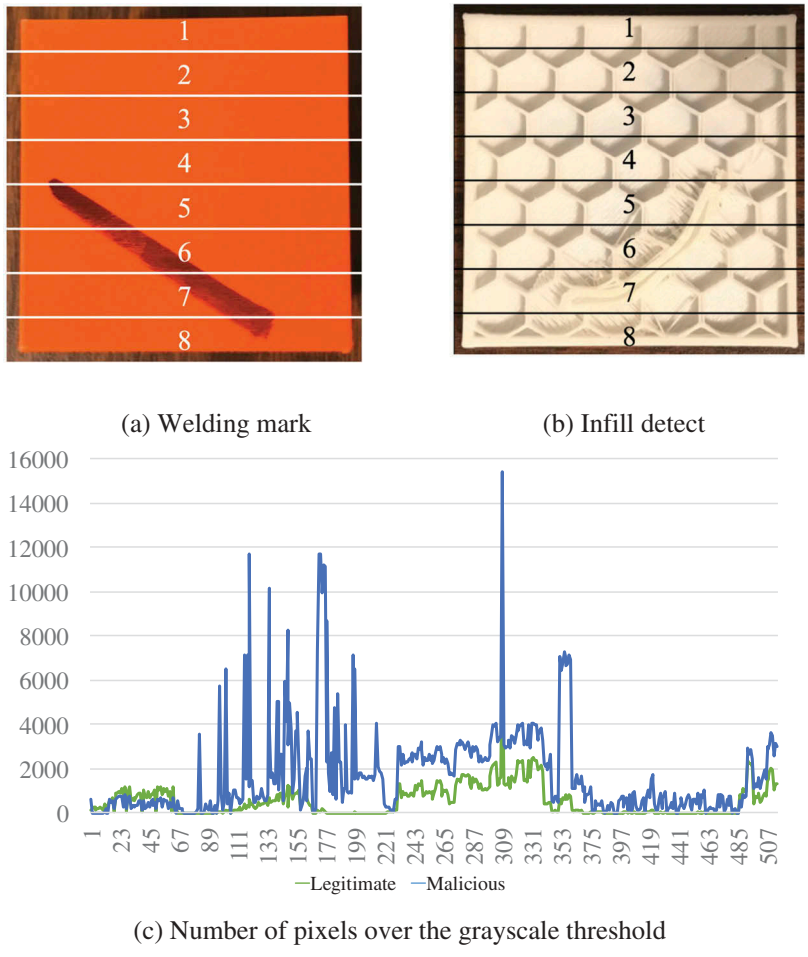



Figure 5. Image data analysis (a) Welding mark (b) Infill detect (c) Number of pixels over the greyscale threshold.

maximum, medium, minimum, number of zero crossings (after centering), peak-to-peak value, skewness, kurtosis, and root-mean-square value (RMS) are calculated as features for detection. In Figure 6, the unit for y-axis is g (G-forces) and the unit for x-axis is seconds.

*4.8.2.4. Acoustic data.* The acoustic sensor is installed on CNC milling machine and around the stepper motor of the conveyor. They are FC-04 sound sensor module with sampling rate of 9,600 bauds, controlled by Arduino UNO microcontroller.

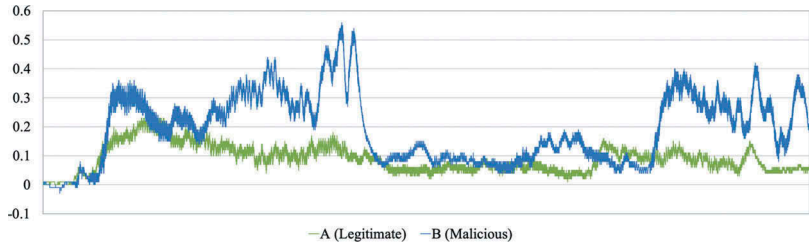



Figure 6. Acceleration data analysis.

Figure 7 shows the first 2 min of the CNC milling process with legitimate and malicious settings via manipulating the spindle speed.

To analyze the acoustic data, the window time for feature extraction is set to 1 s when the sensor collects 20 acoustic signals per second. At every second, the acoustic signal mean, standard deviation, maximum, medium, minimum, number of acoustic data amplitude over 180, 185 and 200 are calculated as features for detection. In Figure 7, the unit for y-axis is dB, the unit for x-axis is seconds.

*4.8.2.5. Current data.* The current sensor is installed in multiple places in testbed including heating elements in the heating chamber, robotic arm, conveyor stepper motor. They are Gikfun ACS712 current sensors controlled by Arduino UNO microcontroller. Figure 8 shows the current data from the conveyor.

In analyzing the current data, the window time for different machines or process can vary. Within each window time, the current mean, standard deviation, maximum, medium, minimum, and the number of acoustic data amplitude over the threshold are calculated as features for detection. In Figure 8, the unit for y-axis is amp and the unit for x-axis is seconds.

*4.8.2.6. Temperature data.* The temperature sensor is installed on the heating chamber on the conveyor in the testbed. It is a SainSmart MAX6675 temperature sensor controlled by Arduino UNO microcontroller. It collects data at the rate of 1 Hz. Figure 9 shows the temperature data from the heating treatment process.

To attack the heating treatment process, the intrude may manipulate the heating element power voltage from 5 V to 9 V. The potential physical consequence is the overload of the power system and also changes in the physical character of treated part. As shown in Figure 9, the malicious heating process has a greater increasing trend. The unit for y-axis is Celsius and the unit for x-axis is seconds.

To analyze the temperature data, the window time is set for 7 s. During each window time, the temperature means, standard deviation, maximum, medium, minimum, number of acoustic data amplitude over 25°C, 28°C and 30°C are calculated as features for detection.

*4.8.2.7. Ultrasonic data.* The ultrasonic sensor is installed the on AGV route for monitoring. They are HC-SR04 ultrasonic sensor controlled by Arduino UNO microcontroller. It collects data at the rate of 5 Hz.

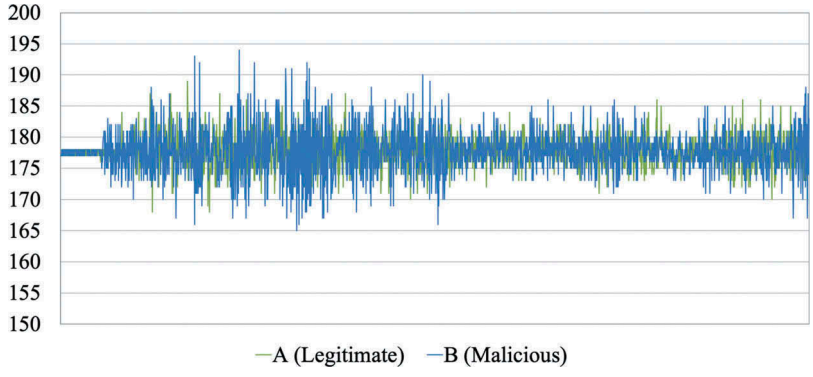



Figure 7. Acoustic data analysis.

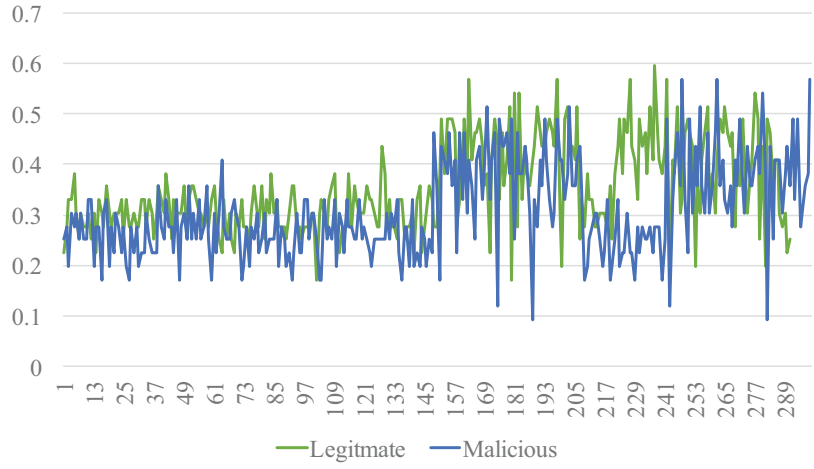



Figure 8. Current sensor analysis.

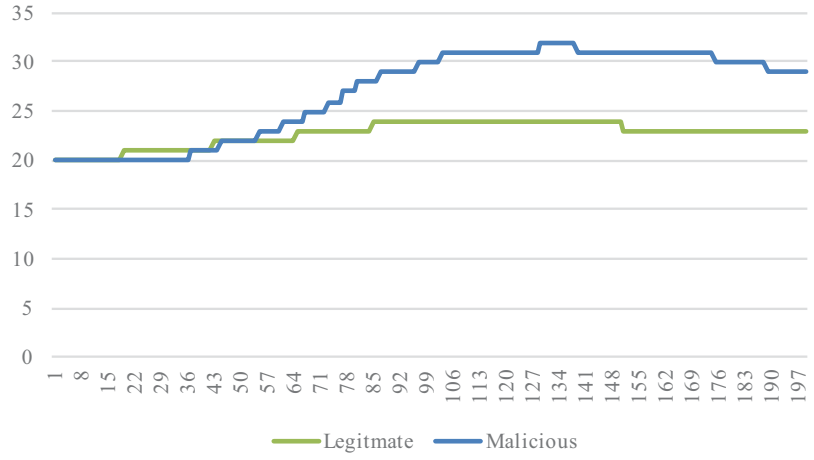



Figure 9. Temperature data analysis.

Figure 10 shows the ultrasonic data from the AGV route area.

To attack the AGV, the intruder changes the control code of the AGV, making it deviate from an assigned route. As shown in Figure 10, both sensors read maximum distance value when the AGV deviates (the unit for y-axis is mm, the unit for x-axis is seconds). The potential consequences of this attack are damages to the AGV, production environment or even humans.

To analyze the ultrasonic data, the data from two ultrasonic sensors are added and the window time is set for 1 s. In every window time, the ultrasonic mean, standard deviation, maximum, medium, minimum, and the number of acoustic data amplitude over the threshold are calculated for detection.

*4.8.2.8. Avoidance sensor data.* The avoidance sensor is installed on the robotic arm for moving sample parts. It is Gikfun ACS712 avoidance sensor controlled by Arduino UNO microcontroller. Figure 11 shows the avoidance sensor data.

The avoidance data is binary. 0 stands for none detected while 1 stands for object detected. The avoidance sensor is installed in an area where a robotic arm is not expected to intrude. It can be areas with equipment or human. When the robotic arm is under attack, its working logic can be changed, some equipment damaged, and humans hurt. For avoidance sensor data, anytime the sensor sends out the value of one, it is classified as an alert.

*4.8.2.9. 3D scanner data.* The 3D scanner is located before the assemble process in the Demander-Testbed. It is an Xbox 360 Microsoft Kinect Sensor controlled by Skanect 3D Scanning software (Figure 12). The software captures object images and export

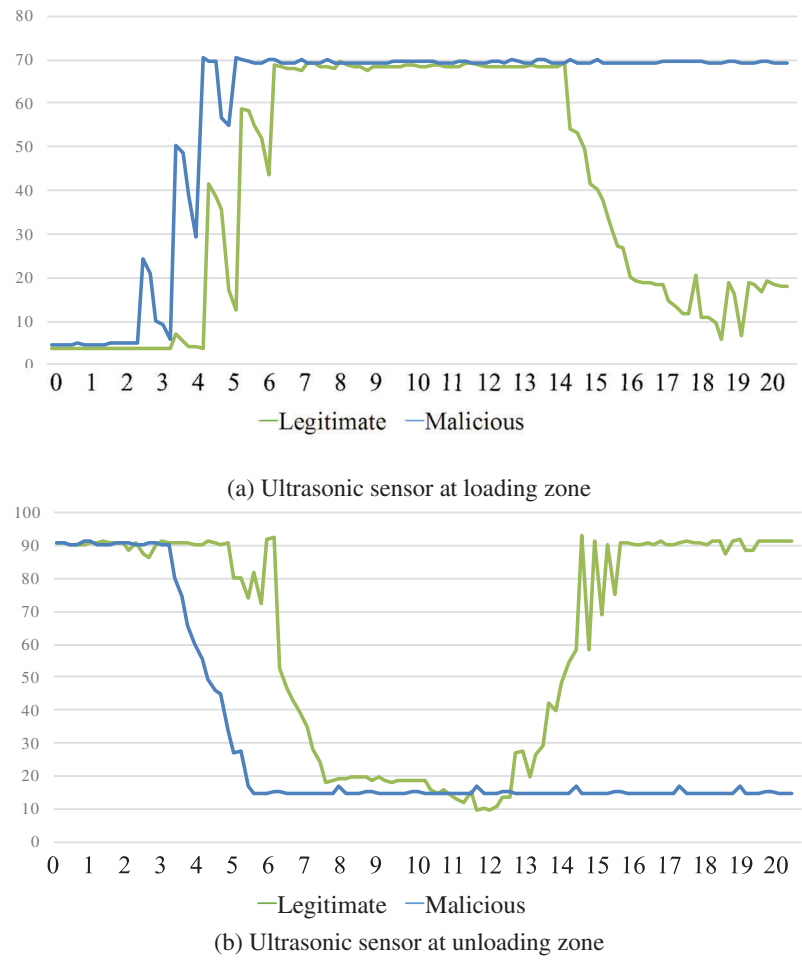



**Figure 10.** Ultrasonic sensor analysis (a) Ultrasonic sensor at loading zone (b) Ultrasonic sensor at unloading zone.

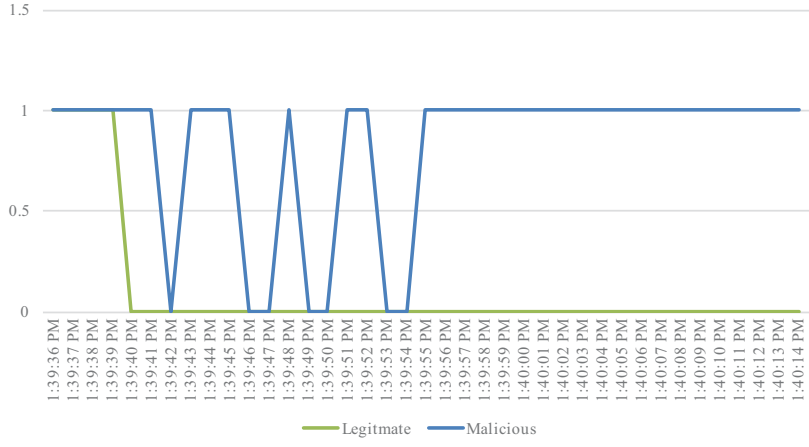



**Figure 11.** Avoidance of sensor data analysis.

to a 'STL' file. The 'STL' file is post-processed in the cloud comparing the dimension difference from the original design.

The testbed provides a platform for the manufacturing security research. The physical data have been utilized in 3D printing process void detection with image [24,43] and power consumption (Wu, Song, and Moon 2019), CNC milling process parameter and design dimension change detection with acoustic and acceleration data (Wu, Song, and Moon 2019). The

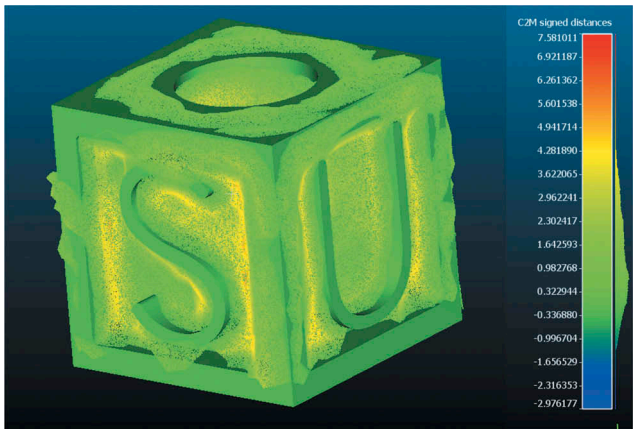



Figure 12. 3D scanning model.

cyber-physical components have been used in developing attack taxonomies [18,19] to gain an understanding of cyber-physical attacks in the manufacturing system.

Even though the testbed is designed to simulate real CMS, there are certain limitations. First, it cannot emulate failures due to continuous usage of equipment or tools since it is not a production system that is running all the time. Also, the impact from real factory settings such as temperature and humidity, is hard to emulate. However, it has the capability of scaling up in terms of various attach scenarios, payloads and consequences. Moreover, new types of cyber protocol and physical equipment can be added to the testbed more easily and without disruption than in a real production system.

## 5. Attack scenarios

As a testbed for intrusion detection and cyber-physical security study, it can undertake different cyber-physical attacks to simulate the intrusion in CMS environment. In this section, five cyber-physical attack scenarios are presented to illustrate the testbed's capability. A case study with physical data collected from the testbed bed are analyzed with the preliminary result presented. The preliminary result shows they physical data can generate a relative accurate result for intrusion detection purpose.

### 5.1. Cyber-physical attack scenarios

The cyber-physical attack scenarios can be developed based on the specifications of the testbed. Cyber-attack vectors such as code injection, shellshock, CSRF can be integrated to create cyber-physical attacks. Following are five attack scenario examples that are implemented in the CSST.

#### 5.1.1. Repackaging attack on 'STL' file

The repackaging attack originated from smartphone applications. For example, an attacker can download an online banking application (Jung et al. 2013), decompile the application, add malicious functions, and upload back to the third party application store to obtain any user's information. It is a very common type of attack on Android devices. In March 2011, it was found that more than 50 apps in Android's official market suffered from repackaging attacks. (Du 2019) The risk could be even higher in untrusted third-party markets.

In CMS, a designer can upload their finished design into an online marketplace from where customers buy some designs. The file can be an 'STL' file for 3D printing, G-code files for CNC machining, or any other types of computer-aided design (CAD) files. The customer can either upload the file to CMS database directly or revise it further and create a self-designed file. When customers select their designs and products from a third party store, the repackaging attack can be done by a malicious user. Attackers may modify a popular design from the online market; reverse-engineering the design; add some malicious defects, parameters, dimensions; and then upload the modified design to the online marketplace.

The customers can be easily misled to purchase and download the design from the online market. Once the modified design is uploaded to CMS database, it goes through a typical process: certification check, model check, order confirmation, and distribute to the specific physical provider for manufacturing. However, it may result in defective parts and malfunctioning machines.

The potential attack payload of a repackaging attack can be design alteration, such as embedding malicious infill void defect via the 'STL' file as shown in Figure 13. It is difficult for operators to observe the alterations. Moreover, Sturm (Sturm et al. 2014) proved that the structural stiffness of a 3D printing test-piece with infill void can be reduced by at least 14%. The malicious seam-shape infill defect can be embedded in the connector part, causing early breakages.

To simulate this attack in the CSST, a malicious 'STL' file can be sent to the database. 3D printer proceeds

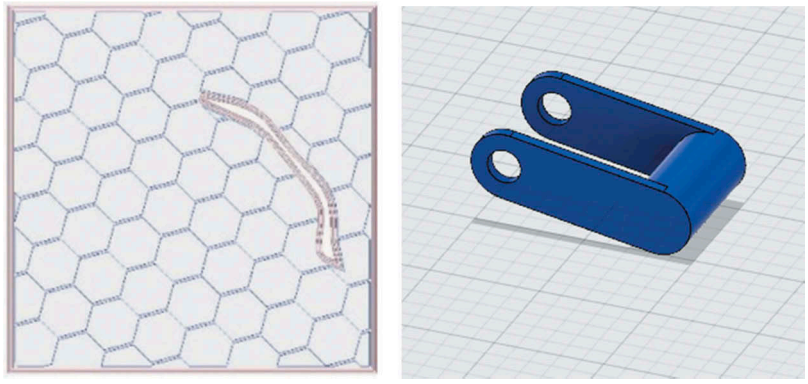


Figure 13. Repackaging on 'STL' file with malicious infill void.

to manufacture the part with a malicious void while sensors collect physical data in the process.

To detect the intrusion, defining the system's process is the first step. The audit data for 3D printing process can be the image, energy and acoustic data that are being monitored during the production process. Structural health data are being monitored in post-production stage. Using the pre-defined architecture and correlation, the physical alerts can be traced all the way back to the customer's design file. An alert is sent to the customer, and the corresponding design and designer are added to the blacklist.

#### 5.1.2. Race condition attack on order priority

A race condition attack makes a device or system to perform two or more operations at the same time (Flanagan and Freund 2001). The outcome of the race condition can obtain an elevated privilege or even superuser privilege. Under this scenario, attackers may use race condition that is elevated to superuser and change the customer orders priority and physical layer availability data.

In the CMS environment, geographically distributed manufacturing equipment is controlled by the global business center. The global business center makes job allocations depending on the customer order priority and physical provider availability. The attacker may gain superuser privilege via race condition and change the customer order priority and physical provider availability data for reducing the system efficiency. Changing the medium level priority orders to high or low can cause the victim's utilization rate and queue time to increase. To simulate this attack on the CSST, computer simulation is used to analyze the operational change after order priority change.

In detecting this intrusion, the audit data for race condition is different in the physical layer: the physical data from the system level, such as utilization rate, maximum waiting time, average waiting time, minimum waiting time, maximum processing time, average processing time, and minimum processing time. The queue time and utilization rate show an abnormal increase in the manufacturer. The investigation via correlation can trace back to the global business center. The attacker's change in customer orders priority file and physical layer availability file can be correlated as evidence for intrusion detection.

#### 5.1.3. SQL injection on 'G-code'

According to the 2016 security report from IBM, 74% of their manufacturing clients are targeted by malicious input data and code injection to attempt to control or disrupt a system, which is notably higher than the cross-industry average of 42%. Among those code injection attacks in manufacturing, SQL injection made up 45% of these attacks and ranks the most frequent cyber-attack vector among all code injection attacks (IBM-Security 2017).

With the SQL injection attack, the intruder can spoof identity, download existing data or upload malicious data to any SQL database with the injection vulnerability. In the CSST, the intruder can spoof into the MySQL 5.7 customer database without knowing the customer's password when the 'magic quote' countermeasure is turned off. For example, the CMS customer with username 'UID001' and password '1234' can log in to the system and upload designs or requirements for fabrication. However, an intruder can use the code 'UID001';–' without any password to log into the account as well. The intruder can have full

access to download, edit, upload, or remove the customers' orders.

One example of cyber-physical attacks via SQL injection is to change a CAD/CAM file or manufacturing specification. A hacker can access a user's account, download a 'G-code' file for CNC milling process, and change specifications such as spindle speed, feed speed, or even tool path. The change can be harmful to the tool life, equipment safety and design structure.

To detect such attacks, sensors such as acoustic sensor and accelerometer can be used to monitor the manufacturing process change. For example, a higher spindle speed may generate a higher amplitude of acoustic data during milling process. Also, the design-specific change should be monitored in design specific data or post-manufacturing inspection, such as dimensional check to general alerts for the cyber-physical attack, and the part can be traced back to its design, certification, and owner.

#### 5.1.4. Shellshock on 3D printer settings

Shellshock is a security bug known as the bash bug (Mary 2015) in Unix bash shell. Shell is used for executing commands and acts as a command language interpreter. Shellshock's impact is similar to the race condition: it allows an attacker to gain control over the computer with elevated privilege.

The shellshock attack can happen on any connected machine based on Unix bash shell, such as the control machine in CMS testbed. In the CMS environment, the Internet-facing service in customer layer may use Bash to process certain requests. As a result, it is under Shellshock risk to let the attacker gain the superuser privilege.

For example, the attacker can change the 3D printer's heating bed and nozzle temperature, by attacking the local control machine. It can cause problems such as low quality, gaps between infill and an outer wall, and the high defective rate at inspection.

To detect the intrusion, the energy consumption on victim 3D printers can be selected as audit data as it can show an abnormal increase. By tracing back to the connected machine, the log file shows that the bash shell has changed the temperature settings. Also, the machine may have been accessed remotely by malicious users from unknown IP addresses. The network packet shows the malicious code from the shellshock attack. The intrusion is detected, and an alert is sent to the global distribution center.

#### 5.1.5. Cross-site request forgery on 'STL' file

In a cross-site request forgery (CSRF) attack, the attacker disrupts the integrity of the user's normal login session. The attacker uses a malicious website sending network requests via the user's browser. The user's browser allows websites to send HTTP requests to any network address (Barth, Jackson, and Mitchell 2008). As a result, the attacker can force an end user to execute unwanted actions such as POST (upload) or GET (download). Similar to the SQL injection, once an attacker gains the normal session for a CMS user, the attacker can use the HTTP request to upload, download, modify the customer's information, or order information. The process of a CSRF attack is: (i) the customer logs in to CMS website with a session cookie automatically saved in the browser, and (ii) the customer visits the malicious CSRF website. The malicious website created by the attacker can send a request legally to CMS website, such as a malicious 3D printing 'STL' file.

To detect the intrusion, the audit data for the physical layer are similar as they are in 3D printing process: acoustic, image and energy data are selected as the audit data. However, the CSRF attack can be detected by the record of abnormal activities in their account as physical layer audit data. Verifications and model check before production can also help to detect explicit malicious changes. As a result, the investigation traces back either from the cyber or physical layer to correlated website hosts and suspicious activities. An alert is sent to the customer, and the malicious website is added to the blacklist.

### 5.2. Case study

In this section, a case study is presented to demonstrate the technical details including manufacturing process, attack scenario, data collection, data analysis and result from an experiment (Table 1). The result shows that physical alerts result in a higher accuracy than the cyber alerts.

In this case study, the attacker targeted at 3D printing process infill structure (Wu, Song, and Moon 2019). The attacker used SQL injection attack to illegally access database, and replace the 'STL' file with an defective file with the physical payload. The payload of the attack places an infill void inside the part to change the overall structural character. The details can be found in section 5.1.1 and 5.1.3.

The Snort is adapted as a network intrusion detection system. It is equipped with standard rules along

with SQL injection rules. However, the attacks also use influence techniques such as network scan to create false alarms and network noise.

To detect infill attacks, a camera is placed above the 3D-printed part to take pictures layer by layer. The pictures are classified with features such as mean values and standard derivation values of greyscale. Three machine learning algorithms are used in detecting malicious defect: k-Nearest Neighbours (kNN), random forest, and anomaly detection.

To evaluate the detection efficiency, accuracy and the false positive rate are used for evaluation. The accuracy is defined in equation (1), where TP stands for True-Positive, TN for True-Negative, FP for False-Positive, and FN for False-Negative. The FPR is defined in equation (2), which specifies the proportion that legitimate information is mistakenly detected as malicious (Gupta, Srivastava, and Sharma 2016).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{FPR} = \frac{FP}{TN + FP} \tag{2}$$

For cyber alerts, the SQL injection caused over 600 alarms in Snort. Some noises such as network scan caused false alarms. Those alarms can be correlated as second meta alert with a higher priority. In total, eight meta-alerts are generated including false alarms from customer activities. The network IDS can capture the evidence of intrusion, but the overall accuracy is low.

For physical alerts, each sensor achieves high accuracy and low FPR with the random forest algorithm. As shown in Table 2, the accuracy shows over 80% accuracy. The image data to detect malicious infill void attack can reach accuracy over 95%. The physical detections have been accomplished in several seconds using a computer with 1.6 GHz Intel Core i5 processor.

In this case study of preliminary experiment, the physical detection shows accurate performance as expected, while network IDS needs more refinement and correlation. More systematic cyber-physical attacks can be designed based on the taxonomy of the cross-domain attacks (Wu and Moon 2017b) and the taxonomy of cyber-physical attacks in CMS.

**Table 1.** Physical data auditing list.

| Equipment | Sensor #1 | Sensor #2 |
|---|---|---|
| 3D Printer | Power Metre | Camera |
| CNC Milling | Accelerometer | Acoustic |
| Mover Robotic Arm | Avoidance Sensor | Accelerometer |
| Welder Robotic Arm | Camera | Accelerometer |
| Conveyor | Acoustic Sensor | Current Sensor |
| Heating Chamber | Temperature Sensor | Current Sensor |
| AGV | Accelerometer | Ultrasonic sensor |

**Table 2.** Physical data intrusion detection analysis.

| Data Type | Accuracy | FPR | Equipment |
|---|---|---|---|
| Image | 0.998 | 0.002 | 3D Printer |
| Power | 0.875 | 0.125 | 3D Printer |
| Acoustic | 0.991 | 0.009 | CNC |
| Acceleration | 0.838 | 0.162 | CNC |
| Acceleration | 0.883 | 0.117 | Robotic arm |
| Current | 1.000 | 0.000 | Conveyor |
| Ultrasonic | 0.870 | 0.130 | AGV |
| Temperature | 0.870 | 0.130 | Heating |

## 6. Conclusion and future work

The CSST testbed is the first cyber-physical manufacturing testbed primarily designed for intrusion detection and prevention investigations. It provides an environment where the future connected manufacturing visions such as CMS, Industry 4.0, Smart Manufacturing or Industrial Internet are represented. The testbed integrates a vulnerable cyber environment with commonly used manufacturing equipment, along with various sensors and computer networks. In the testbed, customers can place orders to initiate the manufacture of part over a supply chain. Also, malicious users can attack cyber and physical domains. Several attack scenarios have been experimented on the testbed for detection analysis.

The results from simulating those scenarios on the testbed show that the physical data can provide accurate and quick responses to the cyber-physical alerts from both component level and system level. The intrusion detection methods and frameworks such as DACDI can be implemented, validated, and evaluated on the testbed.

The testbed contributes to the manufacturing security research. The generated datasets can be used to validate new detection methods and systems from the research community. Furthermore, new prevention countermeasures, communication protocols, or access control can be implemented on the testbed to verify their effectiveness and performance. It can also be used to experiment with new attack vectors and scenarios in

manufacturing systems to achieve a better understanding of different cyber-physical attacks.

The testbed is not a real production system in terms of scale, complexity, and equipment. Therefore, all the possible attack scenarios in fully realized CMS are not expected to happen in the testbed. However, considering the fact that there is no fully realized CMS available yet, the testbed is still the best option for cyber-manufacturing security investigations.

For future work, topics such as wireless network security can be incorporated into the testbed. Also, the correlation analysis (Cuppens and Miège. 2002) between cyber and physical alerts can be explored to create meta alert (Valeur et al. 2004) that further reduce the false alarms and find the root cause of both cyber and physical alert. Emerging technology such as the blockchain in the smart factory (Wan et al. 2019) and Cyber-Manufacturing System (Song and Moon 2019) can be adapted to protect data integrity. In addition, more elaborate computer simulation models using data obtained from the testbed can be developed to explore the correlation analysis. Expansive datasets to be generated from the testbed will enable the exploration of different machine learning techniques and enhanced feature extraction processes for higher alert accuracy rates. Evaluating response time in both physical testbed and computer simulation environment is also a possibility.

## Disclosure Statement

No potential conflict of interest was reported by the authors.

## References


Alves, T. R., M. Buratto, F. M. De Souza, and T. V. Rodrigues. 2014. "OpenPLC: An Open Source Alternative to Automation." Proceedings of the 4th IEEE Global Humanitarian Technology Conference, GHTC 2014, 585–589. doi:10.1109/GHTC.2014.6970342.

Barth, A., C. Jackson, and J. C. Mitchell. 2008. "Robust Defenses for Cross-Site Request Forgery." Proceedings of the 15th ACM Conference on Computer and Communications Security CCS 08, 75. doi:10.1145/1455770.1455782.

Belikovetsky, S., M. Yampolskiy, J. Toh, and Y. Elovici. 2017b. "Cyber-Physical Attack with Additive Manufacturing." In 11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17). http://arxiv.org/abs/1609.00133

Belikovetsky, S., Y. Solewicz, M. Yampolskiy, J. Toh, and Y. Elovici. 2017a. "Detecting Cyber-Physical Attacks in Additive Manufacturing Using Digital Audio Signing." http://arxiv.org/abs/1705.06454

Boyle, R. J., and R. R. Panko. 2013. *Corporate Computer Security*. Upper Saddle River, NJ: Prentice Hall Press.

Bracho, A., C. Saygin, H. Wan, Y. Lee, and A. Zarreh. 2018. "A Simulation-Based Platform for Assessing the Impact of Cyber-Threats on Smart Manufacturing Systems." *Procedia Manufacturing* 26: 1116–1127. doi:10.1016/j.promfg.2018.07.148.

Chhetri, S. R., A. Canedo, and M. A. Al Faruque. 2016. "KCAD: Kinetic Cyber-Attack Detection Method for Cyber-Physical Additive Manufacturing Systems." Proceedings of the 35th International Conference on Computer-Aided Design - ICCAD '16, 1–8. doi:10.1145/2966986.2967050.

Cintuglu, M. H., O. A. Mohammed, K. Akkaya, and A. S. Uluagac. 2017. "A Survey on Smart Grid Cyber-Physical System Testbeds." *IEEE Communications Surveys and Tutorials* 19 (1): 446–464. doi:10.1109/COMST.2016.2627399.

Cuppens, F., and A. Miège. 2002 January. "Alert Correlation in a Cooperative Intrusion Detection Framework." Proceedings - IEEE Symposium on Security and Privacy, 202–215. doi:10.1109/SECPRI.2002.1004372.

Du, W. 2019. *Computer Security: A Hands-on Approach*. Wenliang Du Publisher.

Ehrenfeld, J. M. 2017. "WannaCry, Cybersecurity and Health Information Technology: A Time to Act." *Journal of Medical Systems* 41 (7): 10916. doi:10.1007/s10916-017-0752-1.

Flanagan, C., and S. N. Freund. 2001. "Detecting Race Conditions in Large Programs." In Proceedings of the 2001 ACM SIGPLAN-SIGSOFT, 90–96. doi:10.1145/379605.379687.

Gates, D. 2018. "Boeing Hit by WannaCry Virus, but Says Attack Caused Little Damage." *The Seattle Times*. https://www.seattletimes.com/business/boeing-aerospace/boeing-hit-by-wannacry-virus-fears-it-could-cripple-some-jet-production/

Goldenberg, N., and A. Wool. 2013. "Accurate Modeling of Modbus/TCP for Intrusion Detection in SCADA Systems." *International Journal of Critical Infrastructure Protection* 6 (2): 63–75. doi:10.1016/j.ijcip.2013.05.001.

Gupta, N., K. Srivastava, and A. Sharma. 2016. "Reducing False Positive in Intrusion Detection System: A Survey."*International Journal of Computer Science and Information Technologies* 7 (3): 1600–1603. https://pdfs.semanticscholar.org/4a2c/0b466170af931e75ef5a0f03805807cebf2e.pdf

IBM-Security. 2017. "Security Trends in the Manufacturing Industry." *IBM*. https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=SEL03123USEN&

Jung, J. H., J. Y. Kim, H. C. Lee, and J. H. Yi. 2013. "Repackaging Attack on Android Banking Applications and Its Countermeasures." *Wireless Personal Communications* 73 (4): 1421–1437. doi:10.1007/s11277-013-1258-x.

Kemmerer, R. A., and G. Vigna. 2002. "Intrusion Detection: A Brief History and Overview." *Computer* 35 (4): supl27–30. doi:10.1109/MC.2002.1012428.

Khamphakdee, N., N. Benjamas, and S. Saiyod. 2014 May. "Improving Intrusion Detection System Based on Snort Rules for Network Probe Attack Detection." 2014 2nd International

Conference on Information and Communication Technology, ICoICT 2014, 69–74. doi:10.1109/ICoICT.2014.6914042.

Lab, K. 2017. "The State of Industrial Cybersecurity 2017." *Business Advantage Group Limited*.

Langner, R. 2011. "Stuxnet: Dissecting a Cyberwarfare Weapon." *IEEE Security and Privacy* 9 (3): 49–51. doi:10.1109/MSP.2011.67.

Lee, R. M., M. J. Assante, and T. Conway. 2014. "German Steel Mill Cyber Attack." *Industrial Control Systems*, 1–15. http://ics3.sans.org/media/ICS-CPPE-case-Study-2-German-Steelworks_Facility.pdf

Leens, F. 2009. "An Introduction to I2C and SPI Protocols." *IEEE Instrumentation and Measurement Magazine* 12 (1): 8–13. doi:10.1109/MIM.2009.4762946.

Lippmann, R. P., D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A, Zissman. 2000. "Evaluating Intrusion Detection Systems: The 1998 DARPA off-Line Intrusion Detection Evaluation." Proceedings - DARPA Information Survivability Conference and Exposition, DISCEX 2000, 2014 August 2, 12–26. doi:10.1109/DISCEX.2000.821506.

Mahoney, M. V., and P. K. Chan. 2003. "An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection." In Proceedings of the Sixth International Symposium on Recent Advances in Intrusion Detection, 220–237. 2820 (LI). doi:10.1007/b13476.

Mary, A. C. 2015. "Shellshock Attack on Linux Systems – Bash." *International Research Journal of Engineering and Technology* 2: 1322–1325.

Minnick, J. 2016. "The Biggest Cybersecurity Problems Facing Manufacturing In 2016." http://www.mbtmag.com/article/2016/01/biggest-cybersecurity-problems-facing-manufacturing-2016

Moore, S. B., J. Gatlin, S. Belikovetsky, M. Yampolskiy, W. E. King, and Y. Elovici. 2017. "Power Consumption-Based Detection of Sabotage Attacks in Additive Manufacturing," 1–19. http://arxiv.org/abs/1709.01822

National Research Council. 2014. *At the Nexus of Cybersecurity and Public Policy: Some Basic Concepts and Issues*. Washington, D.C: National Academies Press.

NIST. 2016. "Smart Manufacturing Systems (SMS) Test Bed." *National Institute of Standards and Technology*. https://www.nist.gov/laboratories/tools-instruments/smart-manufacturing-systems-sms-test-bed

Pan, Y., J. White, D. Schmidt, A. Elhabashy, L. Sturm, J. Camelio, and C. Williams. 2017. "Taxonomies for Reasoning about Cyber-Physical Attacks in IoT-Based Manufacturing Systems." *International Journal of Interactive Multimedia and Artificial Intelligence* 4 (3): 45. doi:10.9781/ijimai.2017.437.

Qassim, Q. S., N. Jamil, R. Jidin, M. E. Rusli, M. N. Ahmad Zawawi, M. Z. Jamaludin, M. R. Z'aba, and W. A. W. Kamarulzaman. 2018. "A Review: Towards Practical Attack Taxonomy for Industrial Control Systems." *International Journal of Engineering and Technology(UAE)* 7 (2.14 Speci): 145–152. doi:10.14419/ijet.v7i2.14.12815.

Song, J., and Y. B. Moon. 2019. "A Secure Cyber-Manufacturing System Augmented by the Blockchain." In Proceedings of the ASME 2019 International Mechanical Engineering Congress and Exposition, 1–8, Salt Lake City, UT.

Song, Z., and Y. Moon. 2016a. "Assessing Sustainability Benefits of Cybermanufacturing Systems." *International Journal of Advanced Manufacturing Technology* 1–18. doi:10.1007/s00170-016-9428-0.

Song, Z., and Y. B. Moon. 2016b. "Performance Analysis of CyberManufacturing Systems: A Simulation Study." *IFIP Advances in Information and Communication Technology* 492 (May): 592–605. doi:10.1007/978-3-319-54660-5_53.

Stacy, C. 2018. "'Big Red Flag': Automakers' Trade Secrets Exposed in Data Leak - the New York Times." *The New York Times*. https://www.nytimes.com/2018/07/20/business/suppliers-data-leak-automakers.html

Sturm, L. D., C. B. Williams, J. A. Camelio, J. White, and R. Parker. 2014. "Cyber-Physical Vulnerabilities in Additive Manufacturing Systems." In International Solid Freeform Fabrication Symposium, 951–963. http://sffsymposium.engr.utexas.edu/sites/default/files/2014-075-Sturm.pdf

Timofte, J. 2008. "Intrusion Detection Using Open Source Tools." *Informatica Economica Journal XII 2* 2 (2): 75–79. http://www.revistaie.ase.ro/content/46/Timofte.pdf

Turner, H., J. White, J. A. Camelio, C. Williams, B. Amos, and R. Parker. 2015. "Bad Parts: Are Our Manufacturing Systems at Risk of Silent Cyberattacks?." *IEEE Security and Privacy* 13 (3): 40–47. doi:10.1109/MSP.2015.60.

Valeur, F., G. Vigna, C. Kruegel, and R. A. Kemmerer. 2004. "A Comprehensive Approach to Intrusion Detection Alert Correlation." *IEEE Transactions on Dependable and Secure Computing* 1 (3): 146–168. doi:10.1109/TDSC.2004.21.

Vincent, H., L. Wells, P. Tarazaga, and J. Camelio. 2015. "Trojan Detection and Side-Channel Analyses for Cyber-Security in Cyber-Physical Manufacturing Systems." *Procedia Manufacturing* 1: 77–85. doi:10.1016/j.promfg.2015.09.065.

Wan, J., J. Li, M. Imran, and D. Li. 2019. "A Blockchain-Based Solution for Enhancing Security and Privacy in Smart Factory." *IEEE Transactions on Industrial Informatics* 15 (6): 3652–3660. doi:10.1109/TII.2019.2894573.

Wu, M., H. Zhou, L. L. Lin, B. Silva, Z. Song, J. Cheung, and Y. B. Moon. 2016. "Detecting Attacks in Cyber Manufacturing Systems: Additive Manufacturing Example." In the 3rd International Conference on Mechanical, Materials and Manufacturing, 1–4, October 26–29, Savannah, GA.

Wu, M., V. V. Phoha, Y. B. Moon, and A. K. Belman. 2016. "Detecting Malicious Defects in 3D Printing Process Using Machine Learning and Image Classification." In Proceedings of the ASME 2016 International Mechanical Engineering Congress and Exposition, 4–9. Phoenix, Arizona. doi:10.3389/fncom.2016.00004.

Wu, M., and Y. B. Moon. 2017a. "DACDI (Define, Audit, Correlate, Disclose, and Improve) Framework to Address Cyber-Manufacturing Attacks and Intrusions." *Manufacturing Letters*. doi:10.1016/j.mfglet.2017.12.009.

Wu, M., and Y. B. Moon. 2017b. "Taxonomy of Cross-Domain Attacks on CyberManufacturing System." *Procedia Computer Science* 114: 367–374. doi:10.1016/j.procs.2017.09.050.

Wu, M., and Y. B. Moon. 2018. "Taxonomy for Secure CyberManufacturing System." In Proceedings of the ASME

2018 International Mechanical Engineering Congress and Exposition, 1–9. Pittsburgh, PA.

Wu, M., Z. Song, and Y. B. Moon. 2019. "Detecting Cyber-Physical Attacks in CyberManufacturing Systems with Machine Learning Methods." *Journal of Intelligent Manufacturing* 30 (3): 1111–1123. doi:10.1007/s10845-017-1315-5.

Yampolskiy, M., A. Skjellum, M. Kretzschmar, R. A. Overfelt, K. R. Sloan, and A. Yasinsac. 2016. "Using 3D Printers as Weapons." *International Journal of Critical Infrastructure Protection* 14: 58–71. doi:10.1016/j.ijcip.2015.12.004.

Yampolskiy, M., W. E. King, J. Gatlin, S. Belikovetsky, A. Brown, A. Skjellum, and Y. Elovici. 2018. "Security of Additive Manufacturing: Attack Taxonomy and Survey." *Additive Manufacturing* 21 (March): 431–457. doi:10.1016/j.addma.2018.03.015.

Yang, W., and S. Takakuwa. 2017 December 3–6. "Simulation-Based Dynamic Shop Floor Scheduling for a Flexible Manufacturing System in the Industry 4.0 Environment." 2017 Winter Simulation Conference (WSC), Vol. 6, 3908–3916, Las Vegas, NV.