August 2019

Intrusion Detection for Cyber-Physical Attacks in Cyber-Manufacturing System

Mingtao Wu
Syracuse University

ABSTRACT

In the vision of Cyber-Manufacturing System (CMS) , the physical components such as products, machines, and tools are connected, identifiable and can communicate via the industrial network and the Internet. This integration of connectivity enables manufacturing systems access to computational resources, such as cloud computing, digital twin, and blockchain. The connected manufacturing systems are expected to be more efficient, sustainable and cost-effective.

However, the extensive connectivity also increases the vulnerability of physical components. The attack surface of a connected manufacturing environment is greatly enlarged. Machines, products and tools could be targeted by cyber-physical attacks via the network. Among many emerging security concerns, this research focuses on the intrusion detection of cyber-physical attacks.

The Intrusion Detection System (IDS) is used to monitor cyber-attacks in the computer security domain. For cyber-physical attacks, however, there is limited work. Currently, the IDS cannot effectively address cyber-physical attacks in manufacturing system: (i) the IDS takes time to reveal true alarms, sometimes over months; (ii) manufacturing production life-cycle is shorter than the detection period, which can cause physical consequences such as defective products and equipment damage; (iii) the increasing complexity of network will also make the detection period even longer. This gap leaves the cyber-physical attacks in manufacturing to cause issues like over-wearing, breakage, defects or any other changes that the original design didn't intend.

A review on the history of cyber-physical attacks, and available detection methods are presented. The detection methods are reviewed in terms of intrusion detection algorithms, and alert correlation methods. The attacks are further broken down into a taxonomy covering four

dimensions with over thirty attack scenarios to comprehensively study and simulate cyber-physical attacks.

A new intrusion detection and correlation method was proposed to address the cyber-physical attacks in CMS. The detection method incorporates IDS software in cyber domain and machine learning analysis in physical domain. The correlation relies on a new similarity-based cyber-physical alert correlation method. Four experimental case studies were used to validate the proposed method. Each case study focused on different aspects of correlation method performance. The experiments were conducted on a security-oriented manufacturing testbed established for this research at Syracuse University.

The results showed the proposed intrusion detection and alert correlation method can effectively disclose unknown attack, known attack and attack interference that causes false alarms. In case study one, the alarm reduction rate reached 99.1%, with improvement of detection accuracy from 49.6% to 100%. The case studies also proved the proposed method can mitigate false alarms, detect attacks on multiple machines, and attacks from the supply chain.

This work contributes to the security domain in cyber-physical manufacturing systems, with the focus on intrusion detection. The dataset collected during the experiments has been shared with the research community. The alert correlation methodology also contributes to cyber-physical systems, such as smart grid and connected vehicles, which requires enhanced security protection in today's connected world.

INTRUSION DETECTION FOR CYBER-PHYSICAL ATTACKS IN CYBER-MANUFACTURING SYSTEM

by

Mingtao Wu

B.S., Beijing Forestry University, 2013
M.S., Syracuse University, 2015

Dissertation
Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Mechanical and Aerospace Engineering.

Syracuse University
August 2019

ACKNOWLEDGEMENT

It is my pleasure to contribute my dissertation to the research community of mechanical and industrial engineering. The community inspires me with insights and knowledge that allows me to explore. It also always reminds me to conduct my research with ethics, integrity, and professionalism.

I want to express my sincere appreciation to Dr. Young B. Moon, who advised me through master to doctoral program at Syracuse University. He inspired me in both research and education. I appreciate him spending abundant amount of time in advising despite his busy schedule as department chair. I express my sincere gratitude to Dr. Joon S. Park, who provided me insightful suggestions since the very beginning of my research. I also want to appreciate him serving as the Committee Chair in my proposal exam and dissertation defense.

I am very grateful to Dr. John F. Dannenhoffer, III, who helped me through my research proposal to dissertation with honest and authentic suggestions. I must appreciate Dr. Jorge L. Romeu's encouraging talk with me through my doctoral research. He used stories of his doctoral experience to navigate me through tough times. I want to thank Dr. Wenliang (Kelvin) Du for his SEED lab-based course and research advising, which provided me the key enabling techniques in my research experiments. I want to thank Dr. Xiyuan Liu provided me opportunities and guidance in engineering education. Finally, I want to appreciate above professors agreeing to serve as my academic committee members.

Dr. Vir V. Phoha and his doctoral student Amith K. Belman offered great help on machine learning and image classification techniques in my research. I also appreciate their patience and support.

My research cannot be accomplished without help from undergraduate and graduate research students. They are Lucas Lin, Jackie Cheung, Heguang Zhou, Bruno Silva, Noé Aurelle, Yapan Liu, Bingyan Ding, Jupeng Di, Benliu He, Ziming Wang, Jingkai Zhang, Emily A. Greaney, Claire Baron, Côme Butin, Merouane Alliouche, Snehav Sharma, Borui He, and Hao Yu. I also want to appreciate my doctoral colleagues Zhengyi Song and Jinwoo Song, who assisted me in research.

I want to appreciate my family and friends who supported me spiritually. I want to thank Syracuse University supported my research financially via four years of Teaching Assistantship, and one year of Summer Ph.D. Dissertation Fellowship. I want to appreciate facility manager Bruce M. Carlson gave me a used but comfortable armrest chair, which supported my back physically through the last year of my Ph.D.

Table of Contents

Table List

Figure List

List of Abbreviations

ABS	Acrylonitrile Butadiene Styrene
AGV	Automated Guided Vehicle
ANN	Artificial Neural Network
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CMS	Cyber-Manufacturing System
CNC	Computer Numerical Control
CSRF	Cross-Site Request Forgery
CSST	Cyber-Manufacturing System Security Testbed
DACDI	Define, Audit, Correlate, Disclose, and Improve
DIAL	Defect Injection Attack Localization
DIP	Destination Internet Protocol
DNS	Domain Name System
ERP	Enterprise Resource Planning
FMEA	Failure Mode and Effects Analysis
FNR	False-Negative Rate
GE	General Electric
HIDS	Host-Based Intrusion Detection System
HTTP	Hyper Text Transfer Protocol
IBM	International Business Machines
IDMEF	Intrusion Detection Message Exchange Format
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IP	Internet Protocol
IT	Information Technology
KCAD	Kinetic Cyber-Attack Detection
kNN	k-Nearest Neighbors
MAPI	Manufacturers Alliance for Productivity and Innovation
MQTT	Message Queuing Telemetry Transport
NFC	Near-Field Communication
NIDS	Network-Based Intrusion Detection System
NIST	National Institute of Standards and Technology
OS	Operating System
OSSEC	Open Source HIDS SECurity
PIDA	Physical Intrusion Detection Alert
PLA	Polylactide
PLC	Programmable Logic Controller

RDIF	Radio-Frequency Identification
RMS	Root Mean Square Value
SCADA	Supervisory Control and Data Acquisition
SQL	Structured Query Language
SEED	SEcurity EDucation project
SQLi	Structured Query Language Injection attack
SSH	Secure Shell
STL	Stereolithography
SVM	Support Vector Machine
TCP/IP	Transmission Control Protocol/Internet Protocol
TPR	True Positive Rate
TSMC	Taiwan Semiconductor Manufacturing Company
UI	User Interface
UID	User Identification
UIP	User Internet Protocol address
US	United States
USB	Universal Serial Bus
VPN	Virtual Private Network

Chapter 1

Introduction

Cyber-physical attacks started emerging in manufacturing systems at the beginning of this research in 2015.

In this chapter, the security trend in manufacturing systems is discussed, and the terminologies: Cyber-Manufacturing System (CMS), cyber-physical attack, and intrusion detection system are defined. Next, the problem of why current intrusion detection mechanisms fail to address cyber-physical attacks in the context of Cyber-Manufacturing System is discussed, along with attack surface analysis and detection duration analysis. The hypothesis and objective of the research is stated and discussed. Finally, Chapter 1 is concluded with a dissertation overview.

1.1 Security in Manufacturing: Overview

In Cyber-Manufacturing System (CMS), physical machinery and equipment are fully and seamlessly integrated with computational resources such as machine learning, cloud computing, sensors, via computer networks, and the Internet (Z. Song and Moon 2016a). This visionary system promises dramatic improvements in productivity, quality, cost, flexibility, and sustainability (Z. Song and Moon 2016c). Over the years, the manufacturing industry is developing Cyber-Manufacturing System into different extents, such as "Industry 4.0", "Cloud Manufacturing", "Industrial Internet", and "Smart Manufacturing".

However, the openness to the Internet increases the risks of cyber-related attacks. Recently, the cyber-attacks targeting manufacturing system are active. In 2015, report revealed that the manufacturing sector is the second most attacked industry; in the following year, the manufacturing sector received the most confirmed attacks (IBM-Security 2016). Furthermore, among various attack incidents reported in manufacturing, cyber-physical attacks are emerging. Those cyber-physical attacks intrude into manufacturing systems in digital format, carrying a payload that can cause manufacturing equipment or products to develop over-wearing, breakages, scraps or any other unintended changes (Wu, Song, and Moon 2019). One example of cyber-physical attack—Stuxnet worm (Langner 2011)—illustrates that such an attack can be active for months or years before being detected.

1.2 Definitions

To better understand the scope, target and aim of this research, the terminologies are defined: Cyber-Manufacturing System, cyber-physical attack, and intrusion detection system in this section.

1.2.1 Cyber-Manufacturing System

Cyber-Manufacturing System (CMS) is defined as an advanced manufacturing system where physical components are fully integrated and seamlessly networked with computational processes (Z. Song 2018). In CMS, manufacturing resources and capabilities are digitized and encapsulated into production services, and then shared with all users and stakeholders in the network.

Similar manufacturing visions and concepts around the world have been emerging since the early 2010s. They are developed to different extents and under different names, such as "Industry 4.0" by the German government, "Cloud Manufacturing" (L. Zhang et al. 2014) in China, "Industrial Internet" by GE in the US, and "Smart Manufacturing" by NIST in the US. Each concept emphasizes different aspects of the manufacturing system.

1.2.2 Cyber-Physical Attacks

The cyber-physical attack is defined as an attack initiated inside or outside CMS environment as a digital format that intrudes via cyber, causing physical components such as machines, equipment, parts, assemblies, and products to have problems such as over-wearing, breakage, defects or any other change that their original design didn't intend (Wu, Song, and Moon 2019). Since 2010, cyber-physical attacks have emerged in the sectors of critical infrastructure and developed in manufacturing system.

The most infamous cyber-physical attack is Stuxnet in 2010. The Stuxnet worm targeted the centrifuges in an Iranian nuclear facility. The worm compromised the programmable logic controllers (PLC) with unknown flaws—undiscovered computer software vulnerability—to make the centrifuges spin faster than normal speed and tear themselves apart. The victim Windows operating system and Siemens controller are commonly used in manufacturing systems (Langner 2011).

The first ever confirmed cyber-physical attack in manufacturing happened in Germany, 2014. Multiple attackers gained access to the industrial control system in a German steel mill using emails with a malicious attachment. The attack disrupted the blast furnace control system and could not be shut down by employees, ultimately causing significant damage (R. M. Lee, Assante, and Conway 2014).

Those cyber-physical attacks in a manufacturing system are unique from cyber-attacks in manufacturing: a cyber-attack aims at digital domain consequences, while a cyber-physical attack aims for causing physical consequence via a cyber-attack. A detailed cyber-physical attack analysis is presented in Section 2.2.

1.2.3 Intrusion Detection System

An intrusion detection system (IDS) is designed to alarm malicious activities and security violations. For example, a security camera monitors that if anyone is fiddling with the front door of a house, and gives alarms can be viewed as a home intrusion detection system. In general, an IDS comprises two core functions: auditing data regarding suspects and analyzing the data (Mitchell and Chen 2014). There are various intrusion detection products available today, such as Snort (Roesch 1999a), and OSSEC (Karthikeyan and Indra 2010).

The limitation of the intrusion detection system are high false alarm rates, large quantity of alarms and slow response time. In some cases, an intrusion takes more than two months to detect and even longer to remediate. As the complexity of network grows in CMS, the case can take much longer (Jon Minnick 2016), which is longer than a production cycle. This puts the safety and security of manufacturers and customers at risk.

1.3 The Problems

For Cyber-Manufacturing System, the current intrusion detection methods cannot detect cyber-physical attacks in a timely manner. More specifically, (i) the chances of being under attack increase is enlarged due to the Internet connection through product development and manufacturing life-cycle; (ii) currently, it takes time for an IDS to reveal true alarms, sometime over months; (iii) CMS production life-cycle is shorter than a detection period, which increases the chances of physical consequence in production and the consumer market; and (iv) the increasing complexity of networks will take an even longer detection period.

1.3.1 Attack Surface

The attack surface of a connected manufacturing system is the combination of points where the attacker can intrude into the system and leave a cyber or physical consequence. Different from the attack surface for software, hardware, or operating system, the attack surface of a CMS should implicate both cyber and physical domain assets in the manufacturing system. It comprises system actions externally visible to the CMS users together with cyber and physical resources accessed or modified by each action (Manadhata and Wing 2010).

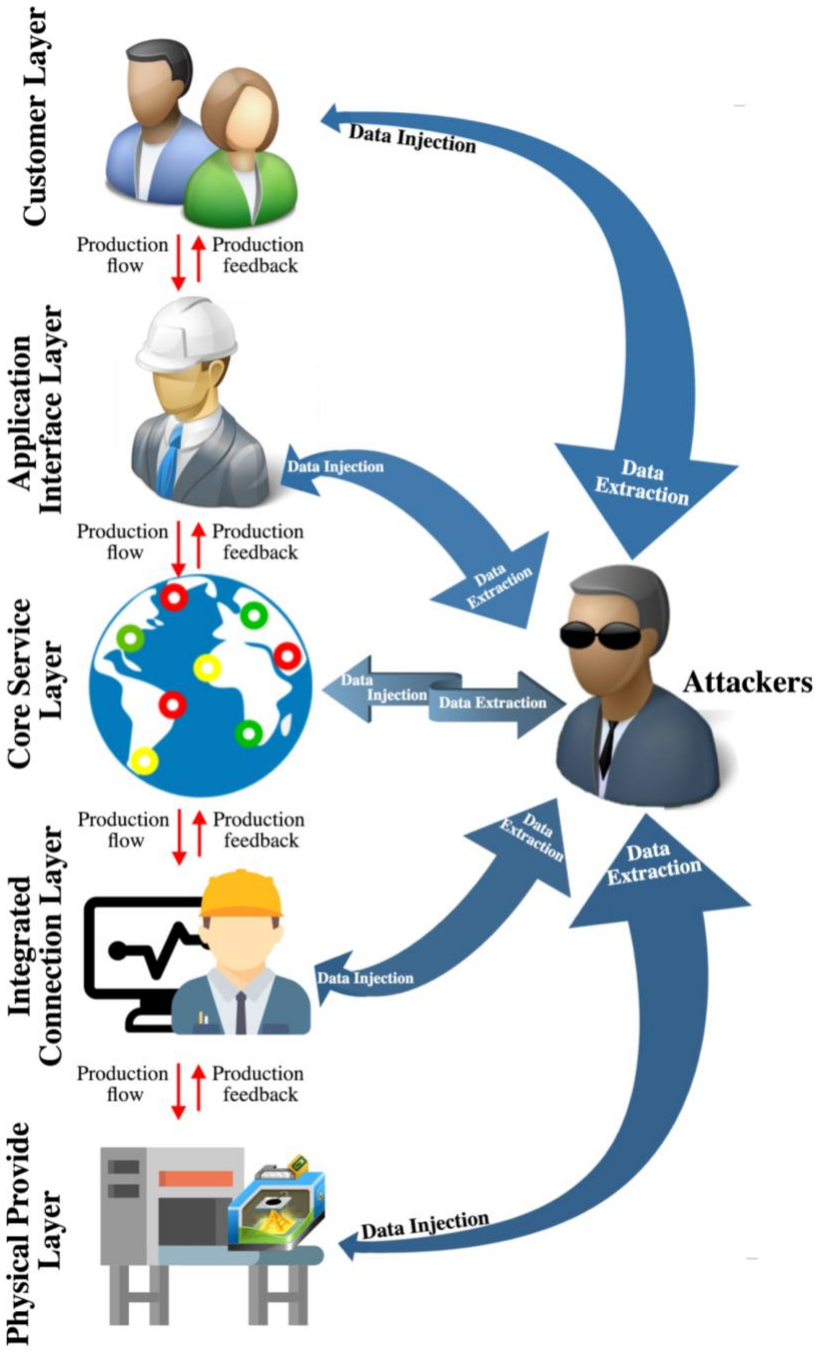


Figure 1 Attack surface analysis for Cyber-Manufacturing System.

The CMS's system constitution can be represented by a five-layer hierarchical architecture: application/customer layer, application interface layer, core service layer, integrated connection layer and physical provider layer, as shown in **Figure 1**. In each layer, the attack surface is

analyzed by enumerating (i) the system actions that are provided by each layer's service and (ii) cyber or physical domain data that could be injected or extracted for malicious purposes by attackers. Five categories of data can be compromised in manufacturing systems (Hutchins et al. 2015a): high-level manufacturing data, low-level manufacturing data, financial data, physical data, and user data are examined in the architecture. Additional cyber payload data for cyber-attacks are also incorporated.

As shown in **Table 1**, targeted data and attack methods are listed (Wu and Moon 2017b). The attack methods are generalized. For example, a privilege compromise attack can be achieved by attack vectors such as Shellshock (Mary 2015) or Buffer Overflow (Moore et al. 2016) depending on different computer environments.

Table 1 Data extraction and injection in a manufacturing security breach.

	Data Extraction	Method	Data Injection	Method
Customer Layer	• Low-level manufacturing data: machine program or model. • Financial data: user financial information. • User data: user personal information. • High-level manufacturing data: design specification.	• Privilege compromise. • User compromise.	• Low-level manufacturing data: machine program or model. • High-level manufacturing data: Design specification. • Cyber payload data: network traffic flooding, executable code.	• File Compromise. • Privilege compromise. • User compromise. • Denial of Service.
Application Interface Layer	• User data: user personal information. • High-level manufacturing data: design specification. • Low-level manufacturing data: machine program or model.	• Privilege compromise. • Access control compromise.	• High-level manufacturing data: design specification. • Low-level manufacturing data: machine program or model.	• Spoofing. • Access control compromise. • Privilege compromise. • Malware installation.
Core Service Layer	• Financial data: user and service provider financial info. • User data: user personal information. • High-level manufacturing data: operational schedule, inventory, productivity (ERP).	• Privilege compromise. • Access control compromise.	• High-level manufacturing data: operational schedule, inventory, productivity (ERP). • Financial Data: malicious financial info.	• Spoofing. • Access control compromise. • Privilege compromise. • Malware installation.

Integrated Connection Layer	• High-level manufacturing data: operational data, inventory data. • Low-level manufacturing data: machine program or model. • Physical data: tooling, quality, control, monitoring data.	• Privilege compromise. • Access control compromise.	• High-level manufacturing data: operational data, inventory data. • Low-level manufacturing data: machine program or model. • Physical data: tooling, quality, control, monitoring data.	• Spoofing. • Access control compromise. • Privilege compromise. • Malware installation.
Physical Provide Layer	• Physical data: tooling, process, monitoring data. • Low-level manufacturing data: machine program or model. • High-level manufacturing data: production plan.	• Privilege compromise. • Access control compromise.	• High-level manufacturing data: production plan. • Low-level manufacturing data: machine program or model. •Physical data: tooling, process, monitoring data.	• Supplier Compromise. • Spoofing. • Access control compromise. • Privilege compromise. • Malware installation.

In conclusion, the attacks surface of a five-layer CMS architecture is greatly enlarged, with various attack methods to manipulate the cyber and physical domain of CMS. Following section 1.3.1.1 to section 1.3.1.5 explains **Table 1** in detail.

1.3.1.1 Customer Layer

The customer layer receives manufacturing service requests from consumers. In this layer, the system action is receiving customers' uploaded design requirements, models, or purchased designs. This layer contains cyber-physical data resources including customer's personal and financial information, as well as high-level and low-level manufacturing data. As a result, the attack surface compromises the system action of file uploading and cyber and physical data. For example, an attacker can intrude via uploading a malicious design model or injecting malicious payload via cyber and physical data. Moreover, the attacker could steal data from the customer layer, such the intellectual property (customer's design specification, design drawing or model), personal information or financial information.

The attacker can intrude via the file uploading system action. The attacker can compromise user accounts by SQL injection or cross-site scripting to access individual customers' data. Moreover, the attacker can inject malicious commands, malicious CAD/CAM code into the cyber and physical data in the customer layer, such as editing the design or product specification in the database. Moreover, the system action in this layer is externally visible and heavily relies on network service. As a result, the attacker can send a large amount of network traffic to cause a denial-of-service attack.

1.3.1.2 Application Interface Layer

The application interface layer transfers a production request into a sequence of implementable production procedures. In this layer, the system action includes services such as computer-aided design or manufacturing (CAD/CAM) provided by CMS. This layer contains cyber and physical data: processed high-level and low-level manufacturing data that could be manipulated to cause a physical consequence, or be stolen.

The attacker could target both the high-level and low-level data, as well as users' personal information. Different from the customer layer, the attack method of compromising a user account will not grant access to the application interface layer. However, compromise to the access control via a CMS insider, such as an employee or supplier, can also allow access to read and download the critical data.

Alongside reading the data, the attacker can also maliciously edit the critical manufacturing data to cause a cyber-physical consequence. Attack methods such as malware, spoofing, access control compromise, and privilege compromise can allow a hacker to edit manufacturing critical data in the application interface layer.

1.3.1.3 Core Service Layer

The core service layer provides system action that allocates manufacturing job requests to the production service provider globally. This layer contains high-level data such as operational schedule, inventory information, productivity, etc. Moreover, this layer also contains financial data from both the customer and manufacturing service provider.

Using similar attack methods in the second layer, the attacker can extract those data to cause financial fraud or intellectual property theft. Moreover, malicious data injection in the database or job allocation algorithms can cause high-level operational chaos in CMS production flow.

1.3.1.4 Integrated Connection Layer

The integrated connection layer provides system actions that control, analyze and predict the manufacturing conditions in the physical provider layer via techniques such as real-time simulation, machine-learning, and digital-twin. The manufacturing data at this level contains both high and lower levels: operation, inventory, machine programs or model. Moreover, there are physical data, such as tooling data, quality data, and monitoring data from the manufacturing process.

The attacker can extract the data from this layer for intellectual property in the physical domain, such as control algorithm, factory layout, etc. The attacker can also inject malicious data that can cause mistakes in decision-making, such as editing the inventory data or machine availability data to mislead job allocation. The attacker can also inject low-level manufacturing data in the machine program or model to influence the physical provider layer further (Wu, Song, and Moon 2019).

1.3.1.5 Physical Provider Layer

The physical provider layer provides system action in manufacturing the customer's order via geographically distributed service providers. Physical assets such as equipment, machines, and sensors are integrated into this layer. The cyber and physical data resources, such as high-level and low-level manufacturing data, directly operate and monitor the machines and production plant.

The attacker can extract the data from this layer for the details of the manufacturing process. Design and machine programs, as well as the manufacturing processes data, such as acoustic emission (C. Song et al. 2016), can be exploited for intellectual property theft. The injection of malicious data in this layer, such as editing the machine program or control algorithm can cause a defective product, machine breakage or even safety incidents. The malicious data may infiltrate the local network or even connected manufacturing equipment.

Alongside the attack methods, such as spoofing, access control compromise, privilege compromise, and malware installation, the compromise of the supplier can also cause cyber-physical consequences in this layer. The supplier of a service provider may possess intellectual properties, production plans, or even access control credentials. One service provider's security breach can influence multiple related manufacturers.

1.3.2 Detection Duration

To discuss the cyber-attack detection duration, attacks are categorized into two categories: unknown attack and known attack.

The unknown attacks exploit vulnerabilities that have not been disclosed publicly. The signature of unknown attack can hardly be defined ahead of time for IDS. As a result, the unknown can hardly trigger any alerts. Unknown attacks commonly last between 19 days and 30 months

without being detected, with a median of 8 months and an average of approximately 10 months (Bilge and Dumitras 2012). It is typically longer than a production cycle, which can cause the defective products on market.

The known attacks exploit discovered vulnerabilities, such as code injection, buffer overflow, phishing attack, denial of service attacks (DoS), etc. Even though the patterns of known attacks can be defined as rules for intrusion detection software, it still can take over 4 months to discover a sophisticated attack among a large quantity of false alarms.

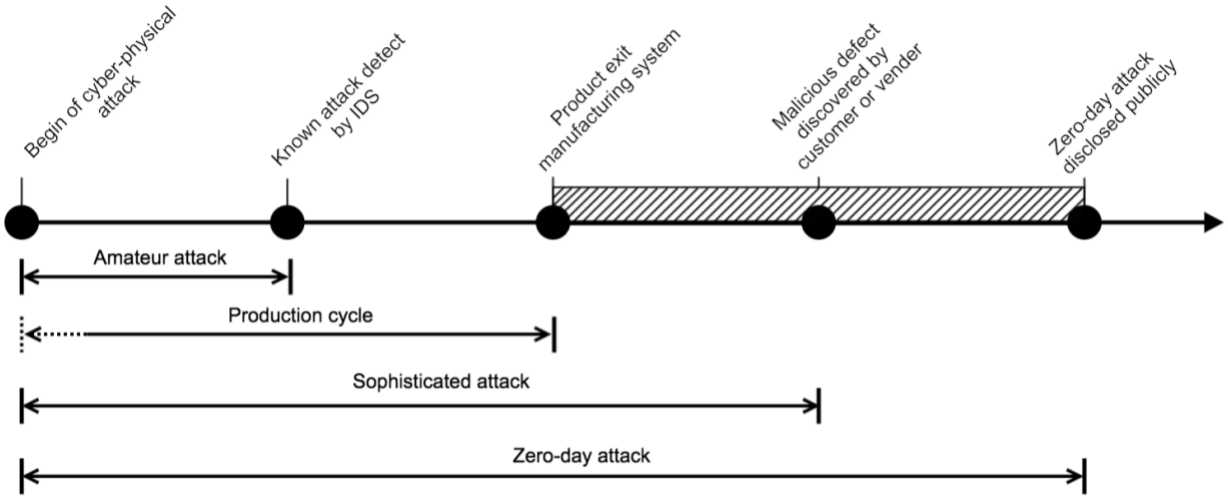


Figure 2 Attack timeline

As shown in **Figure 2**, the sophisticated attack and unknown attack can cause time-delay effect in manufacturing system: malicious defective products could reach consumer market before the IDS reveals the attack action. The defective products could be purchased, caused safety risk, and require market recalls.

1.4 The Objective and Hypothesis

The objective of this research is **to prevent physical damage to equipment and defective parts from cyber-physical attacks in CMS while reducing false alarms.** To achieve the objective, new methodologies are developed and applied, in addition to conventional network and host based intrusion detection system. The new methodology takes advantage of the high accuracy of physical data machine learning, and high efficiency alert correlation method in intrusion detection.

The hypothesis of this research is that **manufacturing process physical data analysis and cyber-physical correlation analysis in Cyber-Manufacturing Systems can prevent physical damage to equipment and defective parts from cyber-physical attacks while reducing false alarms.**

The cyber-physical intrusion in CMS is a new and unique problem. It enters from the cyber network, but influences and damages physical equipment, machines, or even products. Currently, there are not any detection methods for the cyber-physical intrusion. It is a new type of attack that is not well understood and cannot effectively be detected according to prior research.

The Cyber-Manufacturing System is a unique environment as compared to a computer network environment. The physical components are integrated with computational resources via the Internet. In such a system, both physical data (such as acoustic emission and energy consumption) and cyber data (such as network activity and computer host activity) can be extracted from CMS for the study of cyber-physical intrusion detection. The physical data can help detect intrusion quickly, and the correlation between cyber and physical data can help reduce false alarm rate.

The conventional network and host based intrusion detection will be integrated for correlation and root cause analysis. Naturally, the limitations will be inherited from the current network based IDS, such as large number of alarms and the high false alarm rate. The alarm number reduction and false alarm reduction are critical for accomplishing this work.

The measurement of the objective is the accuracy of the detection, the alert reduction rate and response time. The accuracy is defined as the total of the False-Negative Rate (FNR) and the True Positive Rate (TPR). Response time measures the time interval between when the attacker begins the intrusion and the time intrusion detection identifies the adversary. The alert reduction rate shows that how many alerts can be correlated to reduce the detection time.

1.5 Dissertation Overview

In this section, the contribution and organization of this dissertation is summrized.

1.5.1 Contribution

This dissertation contributes to the manufacturing security domain with the following points: (i) applied machine learning in manufacturing process for defect detection, (ii) defined similarity-based cyber-physical alert correlation method, (iii) defined physical alert correlation format, (iv) established the first CMS security testbed, (v) collected data on the testbed with cyber-physical attack experiments for research community.

The application of machine learning in manufacturing processes provides detection for specific problems: 3D printing infill voids attack, and CNC milling process feed and spindle speed attack. The 3D printing infill void attack was defined by Sturm (Sturm et al. 2014) without any detection and prevention methods. The CNC milling process attack was defined by Vincent (Vincent et al. 2015) without any detection method during the manufacturing process. This

dissertation shows methodologies in data selection, collection, feature extraction and classification to detect those attacks during manufacturing processes. They are examples to show how to use physical data to detect malicious changes in manufacturing processes.

The similarity-based cyber-physical alert correlation method has been developed for this work. There was no alert correlation method defined previously between the cyber and physical domains. The possible reasons are: (i) there was no physical security alert to be correlated in the past; (ii) cyber-physical attacks were not well noted, only emerging in recent years; (iii) physical alerts were not standardized for study and analysis; (iv) physical detection does not have high false alarm rate and large number of alerts that requires alert correlation. This correlation method is defined for root cause analysis and false alarm reduction in both cyber and physical domains. Moreover, a Physical Intrusion Detection Alert (PIDA) format is defined for information exchange during alert correlation process.

The Cyber-Manufacturing System Security Testbed (CSST) is established for conducting scientific experiments and validating theories for this dissertation. It is the first security-oriented CMS testbed for intrusion detection research. It can provide an environment for researchers to explore and create new cyber-physical attack scenarios, and validate detection/prevention methods. The data collected from testbed have been shared with researchers, and could potentially be utilized as benchmark datasets for intrusion detection study.

1.5.2 Dissertation Organization

The remainder of this dissertation is organized as follows. Chapter 2 reviews related works on CMS, cyber-physical attacks, intrusion detection and alert correlation. Chapter 3 analyzes cyber-physical attacks in depth, with examples, attack methods, and risk analysis in manufacturing.

Chapter 4 discusses alert in both cyber and physical domains. Chapter 5 describes our methodology of similarity-based cyber-physical alert correlation methods. Chapter 6 introduces the experiment environment, attack design, and four case studies that validate cyber-physical attack detection and alert correlation. Chapter 7 provides an implementation framework for application of this work on candidate systems. Chapter 8 summarizes the dissertation and outlines the limitations and future work.

Chapter 2

Literature Review

In this chapter, publicly known cyber-physical attacks in critical infrastructure domain and manufacturing sector are reviewed; the security status in Cyber-Manufacturing System and related work in two topics: intrusion detection and alert correlation. The domains are limited to computer security, industrial and manufacturing engineering, and cyber-physical system. The intrusion detection and alert correlation are typical computer security domain research topics. However, because of the nature of cyber-physical detection, methodologies that apply to manufacturing quality control, process monitoring, side channel detection can also apply to detecting cyber-physical attacks.

2.1 Cyber-Manufacturing System

Cyber-Manufacturing System is a vision for future manufacturing systems. It integrates physical components with network and computational components seamlessly. An architecture of CMS consists of five layers: the Application/User Layer, the Application Interface Layer, the Global Core Service Layer, the Integrated Connection Layer, and the Physical Provider Layer (Z. Song and Moon 2016b). The first layer—Application/User Layer—includes users and consumers. The second layer—Application Interface Layer—includes support techniques as a buffer of inventory and information processing. The third layer—Core Service Layer—is the global information hub of machine resources, personnel, geographical locations, logistics, user information, etc. The fourth layer—Integrated Connection Layer—is a local analysis and self-control network center. The fifth layer—Physical Provider Layer—is the physical layer, which includes all the manufacturing resources in factory floor.

While similar visions—cloud manufacturing, Industry 4.0, IoT manufacturing, and smart manufacturing—differ in detail, one common character is the physical layer is connected, causing physical components being targets of cyber-physical attack. Furthermore, some of the existing equipment to be integrated are rarely updated (Pan et al. 2017b)—making CMS extremely vulnerable to cyber-attacks. At the same time, emerging unknown (Bilge and Dumitras 2012) cyber-physical exploits such as “STL” file altering attacks (Sturm et al. 2017a) endanger the CMS cyber and physical domain stealthily.

2.2 Cyber-Physical Attacks

Cyber-physical attacks initiate as digital format and intrude via cyber network, causing physical components such as machines, equipment, parts, assemblies, products to develop over-

wearing, breakage, scrap or any other changes that original design does not intend to do (Wu, Song, and Moon 2019). Furthermore, cyber-physical attacks can generate additional long-term effects. For example, a weakened 3D printing structure caused by malicious attacks can compromise customers' safety, with a series of events such as recalls and replacements.

Even though CMS has not fully been realized, cyber-physical attacks have been happening in connected manufacturing systems. In this section, existing published cyber-physical attacks, and security incidents in manufacturing system are reviewed. The information source of cyber-physical attacks can be categorized into four types: security investigation reports, research papers, industry reports and news.

2.2.1 Confirmed Cyber-Physical Attacks

In this section, two publicly confirmed cyber-physical attacks: Stuxnet (Langner 2011) and German steel mill attack (R. M. Lee, Assante, and Conway 2014) are presented. These two incidents are well documented and published. Cyber-physical attacks in the perspective of attack vector, attack impact, attack target and attack consequence (Wu and Moon 2017b) are analyzed.

2.2.1.1 Iran Stuxnet Attack

The Iran Stuxnet attack is documented by multiple sources (Langner 2011; Kelley 2013; Karnouskos 2011; Yadegari and Mueller 2012; Lindsay 2013).

In 2010, secret Iranian centrifuges were targeted by Stuxnet—a malicious computer worm. The worm compromised the programmable logic controllers (PLC) with unknown flaws—undiscovered computer software vulnerability—to make the centrifuges spin faster (Karnouskos

2011). Even though it happened in a critical infrastructure, the victim hardware such as Windows operating system and Siemens controller are commonly used in manufacturing systems.

Even though Stuxnet is viewed as a turning point in the history of cybersecurity, and the controller hijacking technique on the 315 and 417 controllers are complicatedly designed, the attack vectors are not uncommon: computer worm distributed by Universal Serial Bus (USB) sticks, payload code uses code injection and man-in-the-middle attack (Langner 2011). From Stuxnet example, and following cases, the attack vector and direct attack impact of a cyber-physical attack are similar to existing cyber-attack exploits.

Stuxnet targeted at a particular type of programmable logic controllers (PLCs), collecting information on industrial systems and causing the fast-spinning centrifuges to tear themselves apart. On the infected machines, the centrifuges unintentionally sped up or slowed down and finally were destroyed. Different form a convention cyber-attack on industrial system, the attack target and consequence are in physical domain: destroy of equipment.

2.2.1.2 German Steel Mill Attack

In 2014, multiple hackers using phishing email with malicious attachment gained 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 (R. M. Lee, Assante, and Conway 2014). It is one of the first confirmed cyber-physical attacks in the manufacturing system.

2.2.2 Cyber-Physical Attacks in Research

Several research address attacks on altering CAD/CAM file in manufacturing systems. Sturm (Sturm et al. 2014) examines the cyber-physical vulnerabilities in additive manufacturing

system—a key enabling technology for CMS. Malicious infill void placement by altering an "STL" file is demonstrated in the work. Several tensile test specimens with and without voids were tested. The experiment shown that the specimen fractured at the void location, with average reduction in yield load of 14%, and strain at failure reduction from 10.4% to 5.8%. Sturm (Sturm et al. 2017a) later conducted a case study to evaluate the ability of human subjecting to detect and diagnose a cyber-physical attack on the STL file of a test specimen. Recommendations—improved software checks, hashing or secure signing, improved process monitoring, and operator training—are presented.

Two experiments of cyber-physical attack were conducted among engineering students to test the response and awareness from human upon cyber-physical attacks. Wells (Wells et al. 2014) conducted a cyber-physical attack experiment with sophomore-level engineering students; virus infected the computer to alter the tool path file for 3-axis milling machine. The students are hardly aware of the change and cannot diagnose the problem as a cyber-physical attack. Turner (Turner et al. 2015) conducted a similar experiment: this time, the virus infected computer terminal rewrites the students' G-code for 3D printing to alter the part's geometry. The results show none of the groups were aware that the computer system was under attack.

Zeltmann (Zeltmann et al. 2016b) investigated two cyber-physical attack methods on 3D printing: embed defects and alter printing orientation. This research presents a different defect location compare to Sturm 2014; moreover, the alter printing orientation is explored in this research. The result shows that the ultrasonic detection method can hardly detect both attack methods, while the attack methods cause reduction in strength and failure strain.

Belikovetsky (Belikovetsky, Yampolskiy, et al. 2017) demonstrated a complete chain of attack from cyber-attack aimed at compromising a manufacturing environment, ending with the

destruction of the target system. The final result shows that structural change reduced the fatigue life of a 3D printed drone propeller, causing the part broke down during the flight.

Yampolskiy (Yampolskiy et al. 2016) conducted security analysis in the ability of compromised 3D Printing equipment by cyber-physical attacks. The attack weaponize the equipment in order to cause kinetic, nuclear/biological/chemical or cyber damages. The targets analyzed including 3D object physical properties, contamination, electronic circuits, equipment lifetime, damage, explosion, and environment fire, contamination.

Pan (Pan et al. 2017b) identifies and classifies possible cyber-physical attacks against IoT-based manufacturing processes. The attack vectors include social engineering, malware, cross-site scripting and insufficient authentication. It also identifies the vulnerability in different manufacturing processes such as milling, turning, drilling, 3D printing, soldering, heat treatment, and surface coat.

2.2.3 Cyber-Physical Attacks in News

The news documents the occurrence of cyber-physical attacks incidents, but rarely provides detail of the attack method, consequence and post attack investigation results. News that documents the potential of the cyber-physical attacks in manufacturing system nowadays are presented. Two types of attack: ransomware and data breach are included.

2.2.3.1 Ransomware

Though the ransomware was not designed to target the manufacturing specifically, the consequence of downtime in production illustrates how critically cyber-physical attacks can damage manufacturing systems that are connected by the Internet.

In 2017, the WannaCry ransomware affected the car manufacturer Dacia (owned by French Renault) in Romania, and caused Renault to temporarily stop production at several sites to prevent the spread of the attack (Kaspersky Lab 2017).

In 2018, Boeing production plant in Charleston, South Carolina, US is hit by WannaCry ransomware with few machines influenced (Gates 2018). Same year, the world's largest manufacturer of semiconductors and processors Taiwan Semiconductor Manufacturing Company (TSMC) was forced to shut down for a production day, because of the WannaCry ransomware (Mohit Kumar 2018).

2.2.3.2 Data breach

The data breach in manufacturing not only can cause secret and intellectual property theft, but also control data alteration.

In 2018, a total of seven auto companies were impacted by the data leak, including auto manufacturer Chrysler, Ford, GM, Tesla, Toyota and Volkswagen. The data included 10 years of assembly-line schematics and control settings for robotics used to build the cars, along with internal ID and VPN-request forms. The permissions to the server were set to allow anyone to write, which means the data could be accessed, downloaded, and changed by anyone (Spring 2018). It potentially could cause a successful Cyber-physical attack in manufacturer shop floor. However, since the incident occurred in July 2018, there was no follow up security investigation published.

The cyber-physical attack types in manufacturing are not limited to the cases in above news. In fact, limited security incidents are published, with a trend of under-reporting security incidents in manufacturing sector (IBM-Security 2017).

2.2.4 Industry Insights on Cyber-Physical Attacks

The security industry report provides the security trend overview yearly. The institutes and companies provide those reports including International Business Machines (IBM) X-Force, Kaspersky, Verizon, Symantec, Deloitte.

2.2.4.1 IBM X-Force

The IBM X-Force publish yearly *Threat Intelligence Index* security report about all industries. The manufacturing sector always ranks in top five attacked industry in all recent five years (Bradley et al. 2015; IBM-Security 2016; Alvarez et al. 2017; IBM-Security 2018, 2019). Specially in 2017, a separate report *Security Trends in the Manufacturing Industry* is published focus on manufacturing security (IBM-Security 2017) reveals that the manufacturing industry had the most number of confirmed security incidents among all industry sectors in 2016—with almost 40 percent higher than the average across all industries. In their 2019 report, the cyber-physical attack is the future risk of manufacturing system: "future trigger events or new attack tactics may lead to damage to physical infrastructure—and potentially human lives… manufacturing sector must rethink the security of its operational zones and its preparedness to respond to potential attacks of this nature" (IBM-Security 2019).

Moreover, over the years, there is an underreporting trend (IBM-Security 2017; Alvarez et al. 2017) in manufacturing section shown from IBM X-Force yearly report.

2.2.4.2 Kaspersky

Kaspersky Lab publishes yearly report *The State of Industrial Cybersecurity* focus on industrial cybersecurity since 2017. The survey based report shows that the cyber-physical attacks—sabotage or other intentional physical damage by external actors—is one of the major

concerns for industrial control system across the world (Kaspersky Lab 2017; Schwab and Poujol 2018). In 2017, the new rumors of cyber-physical attacks, including Triton and Industroyer, increased the concern.

2.2.4.3 Deloitte

Deloitte published a study *Cyber Risk in Advanced Manufacturing* (Waslo et al. 2017) in collaboration with Manufacturers Alliance for Productivity and Innovation (MAPI) and Forbes Insights. In the study, production downtime, equipment damage or failure, loss of life, fines, litigation expenses, and loss of revenue from brand damage that can persist for months or even years are highlighted are highlighted as consequence of potential cyber-physical attacks via the integration of Internet of Things (IoT) devices in manufacturing system.

2.3 Intrusion Detection System

An intrusion detection system (IDS) is designed to alarm malicious activities and security violations in a system. It comprises two core functions: auditing data regarding suspects and analyzing the data (Mitchell and Chen 2014). It is broadly used in computer security domain to monitor networks and hosts for cyber-attack. In this work, the intrusion detection works in the (i) computer security domain, (ii) manufacturing process domain, and (iii) industrial control domain are reviewed.

Among the three topics, the (i) computer security domain will focus on its original algorithms, limitations and improve methodologies. Although the development of intrusion detection in computer security domain is continuous, it cannot fit into the background of Cyber-Manufacturing System.

The between topic (ii) and (iii), the focus will be on the (ii) manufacturing process domain. The cyber-physical manufacturing and industrial control in intrusion detection are two separate domains (Giraldo et al. 2017; A. Elhabashy 2018) despite the overlapping. Even though the two domains could be correlated in the future, the methodology of this work utilizes more on cyber-physical manufacturing process domain.

2.3.1 Computer Security Domain

In computer security, there are two types of IDS according to audit data: (i) host-based intrusion detection systems and (ii) network-based intrusion detection systems. According to the data analysis method, there are two types of IDS detection methods: (iii) knowledge-based and (iv) behavior-based. Over the years, although new data analysis techniques are applicable to the intrusion detection problem, these four categories of detection methods do not change.

A host-based intrusion detection system (HIDS) is an intrusion detection system that monitors and analyzes the internals activities of a computing system to determine if it is compromised. For example, the “etc/shadow” file in the Linux operating system keeps account passwords. A change of the “etc/shadow” file can trigger the HIDS alarm for potential malicious account information changes.

A network-based intrusion detection system (NIDS) audits network activities to determine if a node is compromised (Liao et al. 2013). For example, if any IP address contributes a large amount of traffic within a short time, the NIDS can be triggered for potential denial of service (DoS) attack.

The knowledge-based intrusion detection techniques work like a blacklist. The technique applies knowledge about known attacks and system vulnerabilities. The intrusion detection system

contains information about these vulnerabilities and triggers an alarm when attack attempts are detected.

By contrast, behavior-based (anomaly-based) intrusion detection approaches look for runtime features that are out of the ordinary (Mitchell and Chen 2014). This approach assumes that an intrusion can be detected by observing a deviation from the normal or expected behavior of the system or the users (Herve Debar 2017).

The limitation of the general intrusion detection system is its slow response time. In some cases, an average intrusion takes more than two months to detect and even longer to remediate. As the complexity of network grows in CMS, the case can take much longer (Jon Minnick 2016). The response time is longer than a product manufacturing lifecycle. This puts the safety and business of users and manufacturers at risk.

For an IDS system, there are three aspects that can be improved: (i) machine learning algorithms, (ii) feature selection and extraction methods, and (iii) training data/detection rules. However, this work does not intend to improve the intrusion detection system in the computer security domain. The IDS system for cyber-physical alert detection with current techniques are adapted and integrated along with their limitations, such as false alarms.

2.3.2 Manufacturing Process Domain

The physical data collected from manufacturing process has been extensively used to detect malicious change during the process. This method is referred as side-channel detection, or physical based detection in this domain.

Vincent (Vincent et al. 2015) proposed a method of Trojan detection and side-channel analyses for cyber-physical manufacturing systems. The structural health monitoring detection

system is used to detect changes to a manufactured part's intrinsic behavior. However, Vincent's detection occurs only after the part is manufactured. Also, it cannot tell if the manufactured part's intrinsic behavior is caused by system flaw or intrusion attacks.

Wu (M. Wu et al. 2016) presented a method of detecting embedded void via "STL" alteration attack during the 3D printing process. The method uses a camera taking top view images during 3D printing process layer by layer. The method can detect any malicious design with machine learning image classification methods. The detection method reached accuracy of 96.1% (Wu et al. 2017). Wu then proposed a detection method for G-code alteration with dimensional change in the CNC milling process (Wu, Song, and Moon 2019). The method utilized acoustic signal during CNC milling process and machine learning, reaching an average detection accuracy of 91.1%.

Chhetri (Chhetri, Canedo, and Faruque 2016) proposed a detection system for cyber-physical attacks in 3D printing process, called KCAD. The system used acoustic analog emissions for detecting potential unknown kinetic cyber-attacks. The system reaches an accuracy of 77.45% in detecting the kinetic cyber-physical attack. This work used statistically estimating function to simulate the analog emissions with corresponding G-code to detect kinetic cyber-physical attack.

Belikovetsky (Belikovetsky, Solewicz, et al. 2017a) presents work for detecting the cyber-physical attacks in a drone propeller, previously presented in (Belikovetsky, Yampolskiy, et al. 2017). Similar to KCAD (Chhetri, Canedo, and Faruque 2016), Belikovetsky used acoustic signal generated by onboard stepper motors during 3D printing process. The method evaluated the deviation between acoustic signals.

Monroy (Monroy et al. 2018) proposed a defect injection attack localization (DIAL) algorithm that uses machines' energy consumption and voltage measurements to identify

compromised machines in the system. The work used multiple 3D printers to simulate a large-scale IoT-enabled manufacturing system. The method can efficiently observe and locate the compromised machine, without providing any detection accuracy.

Overall, the above methods use physical performance and indicators to detect cyber-physical attacks. However, while they can detect physical change, they cannot find the root cause residing in the cyber domain. Moreover, most of the existing research considers the attack or detection from component level: they are investigating one type of machine or a single type of manufacturing process. However, CMS is designed at the system level. This research aims for developing an intrusion detection system from the system level point of view.

2.3.3 Industrial Control Domain

One example of intrusion detection system in industrial control domain is (Hadžiosmanović et al. 2014). It presented a semantic, network-based intrusion detection system monitoring the communication of PLCs in two real-world water plants. The method detects cyber-physical attack by monitoring the state variables of the system, including: constants, attribute data, and continuous data (Giraldo et al. 2018).

There is also a large amount of work in the control theory community, which are mostly high-level and highly mathematical. Most of the work looks at models of the system satisfying a particular equation (Giraldo et al. 2018). This work takes a different route compared to the control theory community: start with low-level manufacturing processes with data intensive methods. Compared to control theory, this work follows the computer security domain method that focuses specifically on intrusion detection problem. However, the control theory domain work provides a good method to generalize the detection method to a high-level perspective.

2.4 Alert Correlation Theory

Different from the review of intrusion detection, the alert correlation research has only been developing in the computer and network security discipline since the early 2000s. The research can be categorized into two categories: (i) alert correlation methodology and (ii) alert correlation process. The correlation methodologies (Qin 2005; Kabiri and Ghorbani 2007; K. Lee et al. 2008; Smith et al. 2008; Ahmadinejad and Jalili 2009; Roschke, Cheng, and Meinel 2011) provide techniques to correlate alerts, while the correlation processes or frameworks (Elshoush and Osman 2012; Cuppens and Miège 2002; Valeur et al. 2004; Shittu et al. 2015) supply general principles from a high level in correlation processes.

Three types of alert correlation methods have been developed: (i) similarity-based method, (ii) sequential-based method, and (iii) case-based method (Salah, Maciá-Fernández, and Díaz-Verdejo 2013).

2.4.1 Similarity-Based Method

The similarity-based method correlates different alerts by defining and using their alert similarities. The main assumption of this method is that similar alerts have the same root causes or similar effects on the system being monitored. The similarity is evaluated by comparing pre-defined features.

The temporal similarity-based method uses time as the pre-defined feature for alert correlation. It is assumed that the alerts by the same attacks are generated within a short time window. Alerts generated within a time window are correlated or aggregated. Other types of similarity-based methods use different attributes in evaluating similarities—such as IP addresses, ports, kinds of service, and users. A similarity measure is typically calculated by computing certain metrics, such as Euclidean distance function. The resulting scores, when compared with threshold

values, determine whether these alerts are to be correlated or not (Salah, Maciá-Fernández, and Díaz-Verdejo 2013).

The similarity-based method has advantages in alert correlation research. It is less complicated, so can be implemented in diverse systems. Moreover, it has proven the effectiveness in reducing the total number of alerts (Salah, Maciá-Fernández, and Díaz-Verdejo 2013) for alert correlation and aggregation processes. In general, this method cannot discover causality relationship between alerts. However, for CMS environments, the correlation can reveal the causality relationship between correlated alerts from different components of CMS because of the connection within the physical production flows.

2.4.2 Sequential-Based Method

The sequential-based method correlates different alerts by using causality relationships. The causality relationship exists between the attack pre-conditions and the attack consequences. The attack pre-conditions are the necessary requirements for a successful exploit, while the attack consequences are the influence of a specific attack payload that occurred. The results from the sequential-based method may embody many false alarms. This is especially prevalent when the logical predicates are not well configured, or the quality of the sensor alerts is not adequate.

For CMS, the cyber-attack pre-condition and physical attack consequence do not necessarily have strong logical predicates. For example, a privilege elevation attack, such as shellshock attack, can cause various types of cyber and physical consequences—production parameter changes, design alteration, or even downtime. It is almost impossible to define the causality relationship between cyber and physical alerts exhaustively.

2.4.3 Case-Based Method

The case-based method correlates different alerts by comparing specific system behaviors with the pre-defined scenarios in a knowledge-based system. The case-based method has been implemented to correlate alerts based on known attack cases. The knowledge base is being updated by inferencing mechanisms, or expert interventions with successfully correlated cases and newly brought-up cases. The case-based correlation method can efficiently correlate pre-defined attack scenarios, but heavily depends on its knowledge base. It is difficult to enumerate every attack scenario in advance and create a useful knowledge base even within a reasonable time frame.

For CMS, the attack case scenarios can help to understand the attack adversary and define the monitoring strategy. However, a case-based method is not adequate for continuously emerging cyber-physical attacks. Also, it is not efficient for the need of real-time alarm correlation. Furthermore, it is not practical to develop different knowledge bases for various CMS enterprises with different network and manufacturing environments.

The alert correlation process provides a high-level principle view on correlation processes. Over the years, several alert correlation frameworks have been developed to correlate IDS datasets. An overview of six alert correlation processes developed over the years is shown in Table 2.

Table 2 Alert correlation process review

Paper	Alert Correlation Process
(Valeur et al. 2004)	Normalization, Preprocessing, Alert Fusion, Alert Verification, Thread Reconstruction, Attack Session Reconstruction, Focus Recognition, Multi-Step Correlation, Impact Analysis, Prioritization.
(Siraj 2006)	Normalization/Formatting, Reduction Severity/Prioritization, Attack Scenario Contribution, Attack Prediction.
(Maggi and Zanero 2007)	Normalization, Prioritization, Aggregation, Correlation and Verification.

(Elshoush and Osman 2012)	Normalization, Preprocessing, Prioritization, Alert, Verification, Alert Fusion.
(Maggi and Zanero 2007)	Alert Normalization, Alert Clustering, Alert Correlation and Intention Recognition.
(Bhuyan, Bhattacharyya, and Kalita 2017)	Alert Normalization, Preprocessing, Correlation Techniques, Post-Processing, and Validation.

Unlike other cyber-alerts-only correlation methods—such as alert correlation (Cuppens and Miège 2002; Valeur et al. 2004; Qin 2005; Valeur 2006), log correlation (Abad et al. 2003), alert aggregation (H Debar and Wespi 2001), alert management (Bhuyan, Bhattacharyya, and Kalita 2017),and alert mining (Julisch and Dacier 2004)—the cyber and physical alerts from CMS possess different causal relationships. Currently available methods are not adequate for correlating cyber and physical alerts. Attributes such as time, IP address and port numbers are not shared between cyber and physical processes. Attributes from manufacturing processes that can enhance the correlation efficiency have not been investigated.

2.5 Summary

From the literature review, the cyber-physical attacks in CMS shows growing toxic potential, but lack systemic understanding. In Chapter 3, two taxonomies are presented to study cyber-physical attacks from intrusion and detection perspective. In chapter 4 and 5, attack detection and correlation methods are analyzed and developed in depth, to detect cyber-physical attacks in CMS.

Chapter 3

Cyber-Physical Attacks

In this chapter, cyber-physical attack is analyzed further in depth. Taxonomy of cyber-physical attack in cyber-physical manufacturing system is presented to give a comprehensive understanding of the attack targets, methods and consequences. 37 cyber-physical attack scenarios are presented based on six common targets in CMS: human, product, equipment, intellectual property, environment, and operation. This section provides a better understanding of cyber-physical attacks, and potential validation methods for detection and prevention research.

3.1 Cyber-Physical Attack Decomposition

To understand cyber-physical attack, the attacks are decomposed into four dimensions: cyber-attack vector, attack cyber-impact, attack physical target and attack physical consequence.

3.1.1 Cyber-Attack Vector

Cyber-attack vector in CMS mainly comes from a network and computer attacks in a digital format. The taxonomy includes shellshock, buffer overflow, race condition, cross-site request forgery, code injection, repackaging, virus, and worms.

Shellshock: It is a security bug in Unix Bash shell, first discovered on 24 September 2014. This vulnerability can exploit various systems and be launched either remotely or from a local machine. The Internet-facing services in CMS, such as service facing customers, can use Bash to process certain requests. This can allow an attacker to gain the root/super or user/administrator access and run malicious commands that result in unauthorized access to a computer system.

Buffer Overflow: This refers to a condition when a program tries to write data beyond the limit of pre-allocated fixed length buffers. It happens when a piece of code or data do not check for appropriate length of input and the value is not the size the program expects(Simmons et al. 2014). This vulnerability can be exploited by a malicious user who gains the root/super or user/administrator access and executes arbitrary commands.

Race Condition: A race condition occurs when multiple processes access and manipulate the same data concurrently. It allows an attacker to gain the root/super or user/administrator privileges while a program or process is in those privilege modes.

Cross-Site Request Forgery (CSRF): Also known as *session riding*, this is a type of attack on website where unauthorized commands are transmitted from a user that the website trusts.

It can happen on web applications facing customers in CMS. A CSRF attack involves a victim user (customer), a trusted site (CMS web), and a malicious site (attack site). When the customer holds an active session with a CMS web application while visiting a malicious site, the malicious site can inject an HTTP requests to the CMS web application user session, causing change in account information.

Code Injection: Code injection is caused by attackers' inputting code into a vulnerable computer program and change the process of execution. The places in CMS for code injection may include SQL (Structured Query Language), OS commands, etc. For example, most small and industrial strength database applications can be accessed using SQL statements for structural modification and content manipulation (Zhu, Bonnie, Anthony Joseph 2011). Malicious users can use SQL injection and manipulate other customer's information.

Repackaging: This is a type of attacks on Android OS applications. Attackers download popular applications from a store, unpack and modify the application with malicious requests of privileges, then post the application in certain third-party app stores. In CMS, the designs with CAD models can be offered online. Similar to repackaging an application, attackers can repackage a design by reverse engineering or just modifying the CAD file; then uploading back to online platforms. Such attacks can cause defective parts, products, or even machine malfunctions.

Virus: This self-replicating program can spread through some types of infected file (Hansman and Hunt 2005).

Worms: This self-replicating program can propagate without using infected files. Worms usually propagate through network services on computers or through emails.

Among all those cyber-attack vectors, 74 percent of manufacturers 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 percent. Among those code injection attacks in manufacturing, SQL injection made up 45 percent of these attacks ranks the most frequent cyber-attack vectors among all code injection attacks (IBM-Security 2017).

3.1.2 Attack Cyber-Impact

The cyber impact shows the impact on digital platforms, such as web application, program, operating system, digital file, etc. The taxonomy includes privilege compromise, user compromise, file compromise, denial of service, and malware installation.

Privilege Compromise: By using attack vectors such as buffer overflow, shellshock, race condition, the attacker can gain higher privileges such as superuser.

User Compromise: An attacker gains unauthorized use of other user account or privileges on a host, web application, or database. An attack such as CSRF can achieve this goal on web applications.

File Compromise: In CMS, CAD/CAM files play a major role. Attacker makes malicious change by using repackaging, code injection, thus can change the critical structure and physical characteristic of the design.

Denial of Service (DoS): An attacker can conduct a denial-of-service attack (DoS attack) that makes a connected machine such as a database or computation resource inaccessible to its intended clients.

Malware Installation: An attack can be launched via user-installed malware, whether user installation or drive-by installation. Installed malware can allow an adversary to gain full control of the compromised systems, potentially leading to the exposure of sensitive information or remote control of the host.

3.1.3 Attack Physical Target

The target of a cyber-physical attack is in physical domain. For a cyber-physical manufacturing system, the target could be sensor, actuator, machine, part/product or even human.

Sensor: Sensors allow monitoring to the manufacturing system and provide data for manufacturing status perdition and simulation.

Actuator: An actuator is a fundamental component of a machine that moves or controls a mechanism or system.

Machines: Machine is the key component of physical provider layer in CMS. It can also be an assembly of actuators, sensors and control unit such as programmable logic controller (PLC).

Manufactured Parts: Manufactured parts or assemblies are the finished products from a production line.

Human: Human can be a target victim in CMS as well. Operators, assembly workers working next to robots are endangered when hackers can send malicious control to actuators.

3.1.4 Attack Physical Consequence

The consequence of a cyber-physical attack is in physical domain, such as tear down a centrifuge (Langner 2011), control a blast furnace (R. M. Lee, Assante, and Conway 2014), or a defective 3D printed drone (Belikovetsky, Yampolskiy, et al. 2017). In general, six types of attack consequence are summarized.

Defective Product: Defective products or even malicious products are physical consequences. The scrap cost, recall will be drawn with defective products or part being manufactured. Following consequences can damage company image or risk human lives.

Machine Manipulation: Attacks can cause problems on machines such power over consumption, unpredicted breakage, compromised precision, slow-down, etc.

Malfunction and Breakage: The breakage or malfunction can be a consequence of machine manipulation.

Loss of System Availability: The critical availability of physical components such as 3D printers, CNC machines, logistics can be compromised.

Environmental Disaster: Environmental disasters such as leakage and explosion are critical physical consequences.

Risk of Death and Serious Injury: Human as most fragile component of CMS is at risk of their health and life when working in environment with hazardous chemical, radiation and robots.

3.2 Scenarios of Cyber-Physical Attack in CMS

Based in prior analysis, six common targets in CMS is emphasized: human, product, equipment, intellectual property, environment, and operation. 37 cyber-physical attack scenarios are designed for detection validation for this study, as well as attack prevention and mitigation study. This section is generalized in the format of an intrusion taxonomy.

3.2.1 Human

In the human category, two types of people are the major target in CMS: customer and worker. Although, those two targets are not further decomposed into sub-targets, the customers can be both end-consumers or people involved in an entire supply chain. Also, the customers' safety can be endangered by a product, while workers' safety can be compromised by the working environment.

As shown in Figure 3, the human as an affected entity is decomposed into four major categories: customer safety risk from product quality or security compromise, and worker safety risk from environment safety or production safety compromise.

Product *quality* risk on *customer* safety. The product quality can be compromised via a cyber-physical attack. Furthermore, the physical consequence can cause human safety risks, such as defective products, weakened structures (Sturm et al. 2014) and reduced product lifetime.

Product *security* risk on *customer* safety. The product security that is compromised via software or hardware can cause human safety risks. The vulnerability through a backdoor may allow attackers to access the product remotely via the Internet and result in safety compromises. For example, a vulnerable infotainment system can allow a hacker to control a Jeep Cherokee's ignition switch, brakes and steering system (WIRED 2015), leading to several accidents. The product can become dangerous even without the remote control: attackers may alter the product software or hardware during production, causing the product to malfunction in the future.

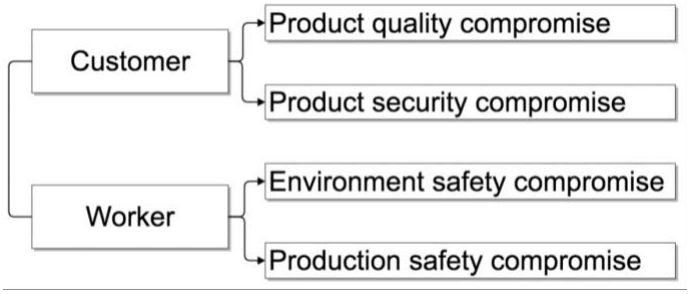


Figure 3 Human category decomposition

***Environment* risk on *worker* safety.** Environmental risk in work space can endanger workers' safety. An attacker may manipulate manufacturing process or emission treatment to increase pollution in a production environment.

***Production* risk on *worker* safety.** The production accident based on malfunctioning manufacturing processes can occur. For example, the UI modification attack (Quarta et al. 2017) on an industrial robotic arm may lead operators on a critical safety hazard.

3.2.2 Product

The product can be compromised in CMS with consequences in unacceptable quality. Three major targets are design processes, manufacturing processes, and quality inspection processes.

As shown in Figure 4, compromises in the product quality are classified into eight categories: product design structure, dimension, design feature, production raw material, equipment, specification, software, inspection equipment and specification.

***Structure* property compromise on *design* integrity.** The attacker can manipulate the design process or the design document to change the structure of a part. The physical performance of a part—such as stiffness, natural frequency—can be affected according to the structural changes and additional quality issues.

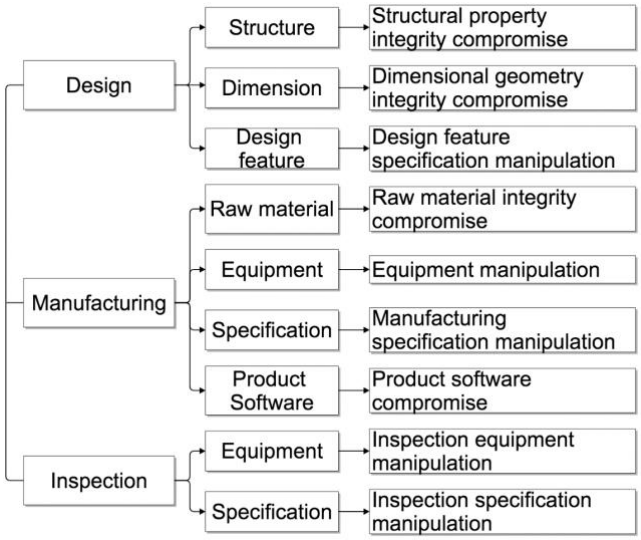


Figure 4 Product category decomposition

***Dimension* compromise in *design* integrity.** The dimensional change can be embedded in CAD/CAM file. Attackers may scale a part incorrectly in one or more dimensions or make alterations in the file. It can cause the part unfit in the assembly design (Pan et al. 2017b).

***Design feature* specification manipulation in *design*.** The design feature such as drilling hole, fillet can be removed or added maliciously by modifying design file. Such attacks can increase manufacturing cost and cause assembly problems.

***Raw material* manipulation in *manufacturing* process.** The raw material or part from upper stream supplier can be affected by the attack. The material changes in the manufacturing processes—such as change of colors, strength, surface roughness—can cause the finished part with a different physical property to the original design. For example, change the 3D printing filament from ABS to PLA plastic. The source part manipulation—such as hardware Trojan on circuit board—can result in malicious defective parts.

***Equipment* manipulation in *manufacturing* process.** Connected equipment can be manipulated by attackers during manufacturing processing. One of the consequences of equipment manipulation is product alteration—resulting in malicious products, defective products, etc.

***Specification* manipulation in *manufacturing* process.** The specification in manufacturing processes can be the target to make product quality alterations. Examples are changes in heat treatment temperature, changes in feed speed of milling, and changes in 3D printer heating nozzle temperature. As a result, the product may be generated in poor quality.

***Product software* compromise in *manufacturing* process.** The software or operating system is a common part of a product—such as automobiles, computers, smartphones, etc. The software can be compromised from a backdoor for further malicious activities by attackers.

Inspection *equipment* manipulation. The equipment for product inspection process can be manipulated. An attacker may make inspection process to accept manufactured products as conforming, despite their unacceptable quality (A. E. Elhabashy et al. 2018).

Inspection *specification* manipulation. Inspection process specification can be manipulated by altering control limits, data, etc. Similarly, parts or products of poor quality may be ignored during the process. Moreover, manipulation with stricter specification can classify otherwise acceptable parts as defective; and cause more downtimes in the investigation for quality improvement.

3.2.3 Equipment

The equipment itself is a target that can bring damage to different physical components in manufacturing systems. The sensors, machines, and controllers are further discussed in equipment manipulation dimension.

As shown in Figure 5, the equipment manipulation can be decomposed into seven categories: sensor manipulation with data integrity and sensor availability, actuator manipulation with its maintenance, specification and availability, controller manipulation with control logic and availability.

Sensor *data* integrity manipulation. The data integrity is important especially for the controlled manufacturing processes, such as heat treatment, injection molding, etc. The loss of integrity in data causes malfunctions in the control of manufacturing processes and can induce production accidents.

Sensor *availability* manipulation. The denial of service (DoS) attack on sensors can make the manufacturing process or even whole system lose availability.

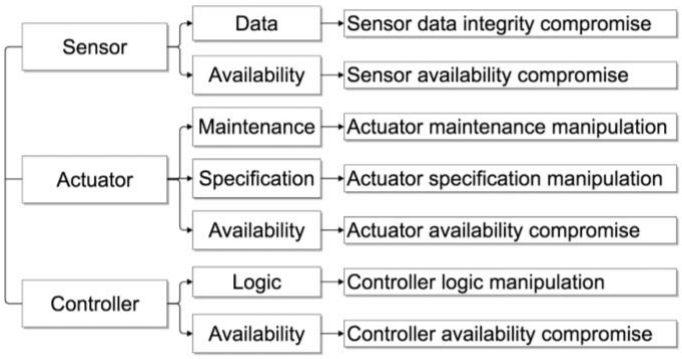


Figure 5 Equipment category decomposition

Actuator *maintenance* manipulation. The schedule or process of actuator maintenance can be attacked, resulting in malicious machine wearing or damage by attackers.

Actuator *specification* manipulation. The malicious change in actuator specification can directly cause physical consequence, such as motor damage, drilling bit damage, etc.

Actuator *availability* manipulation. Loss of actuator availability can result from the denial of service attack (DoS) on a connected actuator.

Controller *logic* manipulation. Controllers such as programmable logic controller (PLC) is commonly used in manufacturing systems. They control assembly lines, robotic arms, etc. But the controller logic can also be manipulated. For example, changes the spindle speed on a CNC milling machine can increase the excessive wearing on the end mill bill and also motor itself. Intensive manipulation may cause incidents such as in the Stuxnet worm incident (Langner 2011).

Controller *availability* manipulation. Similarly, with a denial of service attack (DoS), CMS operator may lose control of the controller and corresponding actuators. For example, the blast furnace cannot be shut down by its control system in the steel mill incident from Germany in 2014 (R. M. Lee, Assante, and Conway 2014).

3.2.4 Intellectual property

Intellectual property theft is a common problem in the current manufacturing system and so will be in CMS. The direct consequence of intellectual property theft is the loss of trade secrets. However, the long-term influence can be physical—counterfeit goods, modified designs, etc. Moreover, new cyber-physical attack methods such as side-channel attack (C. Song et al. 2016) add new methods for intellectual property thefts.

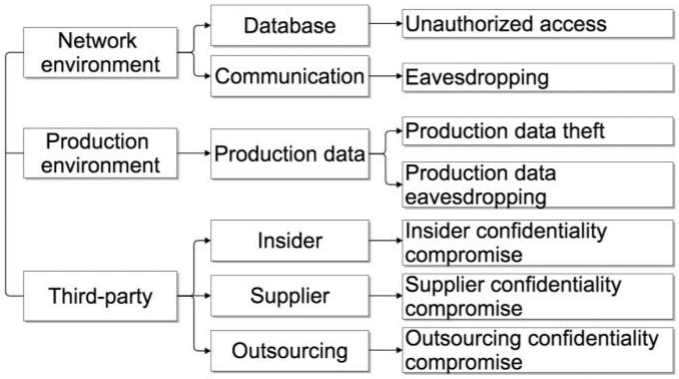


Figure 6 Intellectual property category decomposition

As shown in Figure 6, intellectual property thefts are decomposed into seven categories: intellectual property theft by unauthorized access or compromise network communication; production data theft or eavesdropping; leakage from insider, supplier or outsourcing manufacturer.

Database *unauthorized access* in *network* environment. Attackers can use methods such as code injection, shellshock, or social engineering to make unauthorized access to database or computers that contain intellectual properties and trade secrets.

Communication *eavesdropping* in *network* environment. In the age of CMS, network communication among customers and service providers are ubiquitous. A weak link in communication can create a channel for intellectual property thefts.

Production data *theft*. The data from production can be used to reverse developing and engineering. For example, the acoustic emission data (C. Song et al. 2016) can be used to reconstruct the object being manufactured. Obtaining production data is an indirect way of intellectual property theft.

Production data *eavesdropping*. Similarly, the production data can also be picked up and eavesdropped by compromised devices for reverse engineering. For example, a smartphone (C. Song et al. 2016) in a connected environment can monitor 3D printing processes.

***Insider* confidentiality compromise.** Ill-intended insiders may be able to steal intellectual properties. The employee can sell those data or start up a competing company. The insider threat is a significant factor in intellectual property theft, accounting for 15% of breaches (Verizon 2017).

***Supplier* confidentiality compromise.** Suppliers may be a weak link in the supply chain that leaks intellectual property. A supplier from a country with weak intellectual property law or little intellectual property protection culture, is more exposed to intellectual property thefts.

***Outsourcing* manufacturer confidentiality compromise.** Similarly, outsourcing companies can be a weak link in the supply chain—vulnerable to intellectual property thefts.

3.2.5 Environment damage

Two targets for environmental damage are manufacturing processes and products.

As shown in Figure 7, the environmental damage can be decomposed into six categories: attack on manufacturing process via environmental spill, energy over-consumption or emission manipulation; product chemical property change, energy over-consumption or emission sabotage.

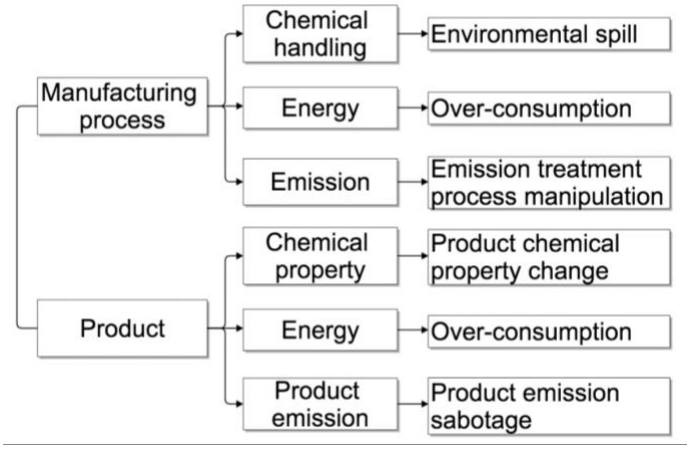


Figure 7 Environment category decomposition

***Environmental spill* in the *manufacturing* process.** Cyber-physical attacks can cause environmental spills in manufacturing systems. In some of the manufacturing processes, such attacks can cause oil and chemical spills, radiological and biological discharges, and accidents causing releases of pollutants.

Energy *over-consumption* in the *manufacturing* process. Attacks that manipulating power consumption in CMS can influence environment indirectly—for example, increase in the process temperature, decrease in storages' environment climate control temperature, etc.

***Emission* manipulation in the *manufacturing* process.** Manipulating the emission treatment process can cause environmental damages in the manufacturing processes. For example, in 3D printing process, the emission rates were observed to depend strongly on extruder temperature (Mendes et al. 2017). As a result, the emission may increase simply by attacking extruder temperature.

Product *chemical property* change. Changes in chemical property of a product, such as acid and alkaline, can influence the product's damage to environment during its lifecycle.

Product *energy* over-consumption. Similarly, the product power consumption during customers' usage can be manipulated by changing product specifications, software, controllers, etc.

Product *emission* sabotage. Product emission during customers' usage can also manipulated by altering software or controller to damage the environment. For example, manipulation of the automobile emission via a software can increase the emission without being noticed during pollution inspection (Contag et al. 2017).

3.2.6 Operation

The operational change and delay have significant consequences for manufacturing systems. For example, unplanned downtime can cost as much as $20,000 potential profit loss per minute (Quarta et al. 2017). Manufacturing processes and supply chains are two major targets for operational schedule delay target.

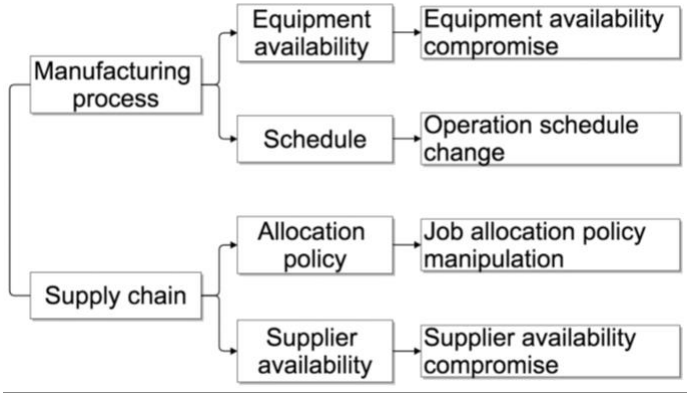


Figure 8 Operation category decomposition

As shown in Figure 8, attacks on operations can be decomposed into four categories: attack on manufacturing process with equipment availability and production schedule; attack on supply chain with job allocation policy and supplier availability.

***Equipment availability* compromise.** Such a downtime in an equipment can cause operational changes within the manufacturer. By carrying out denial of service attacks on machines, assembly lines, the attacker can delay the operational schedule.

Operation *schedule* change. Attackers can change the scheduling in manufacturing processes. Slowing down or speeding up the process both can cause chaotic operations. For example, slightly slowing down the feed speed of CNC milling machine, printing speed of 3D printer, or even conveyor speed, may significantly decrease the utilization of the machine in the long run, and substantially delay the operational schedules.

Allocation *policy* manipulation. The job allocation policy is predefined based on factors such as cost, geographical distance, sustainability, etc. A compromised policy can make incorrect decisions and delay manufacturing schedules.

Supplier *availability* manipulation. Operations rely heavily on suppliers. Attacks on a supplier's service availability or real-time data availability can influence the operations. If the data have been manipulated by attackers, the job allocation system may make incorrect decisions and delay manufacturing schedule.

Overall, the attack scenarios is presented with six potential affected entities. They are further decomposed into 15 targets and 32 sub-targets. Finally, 37 potential attack methods are identified.

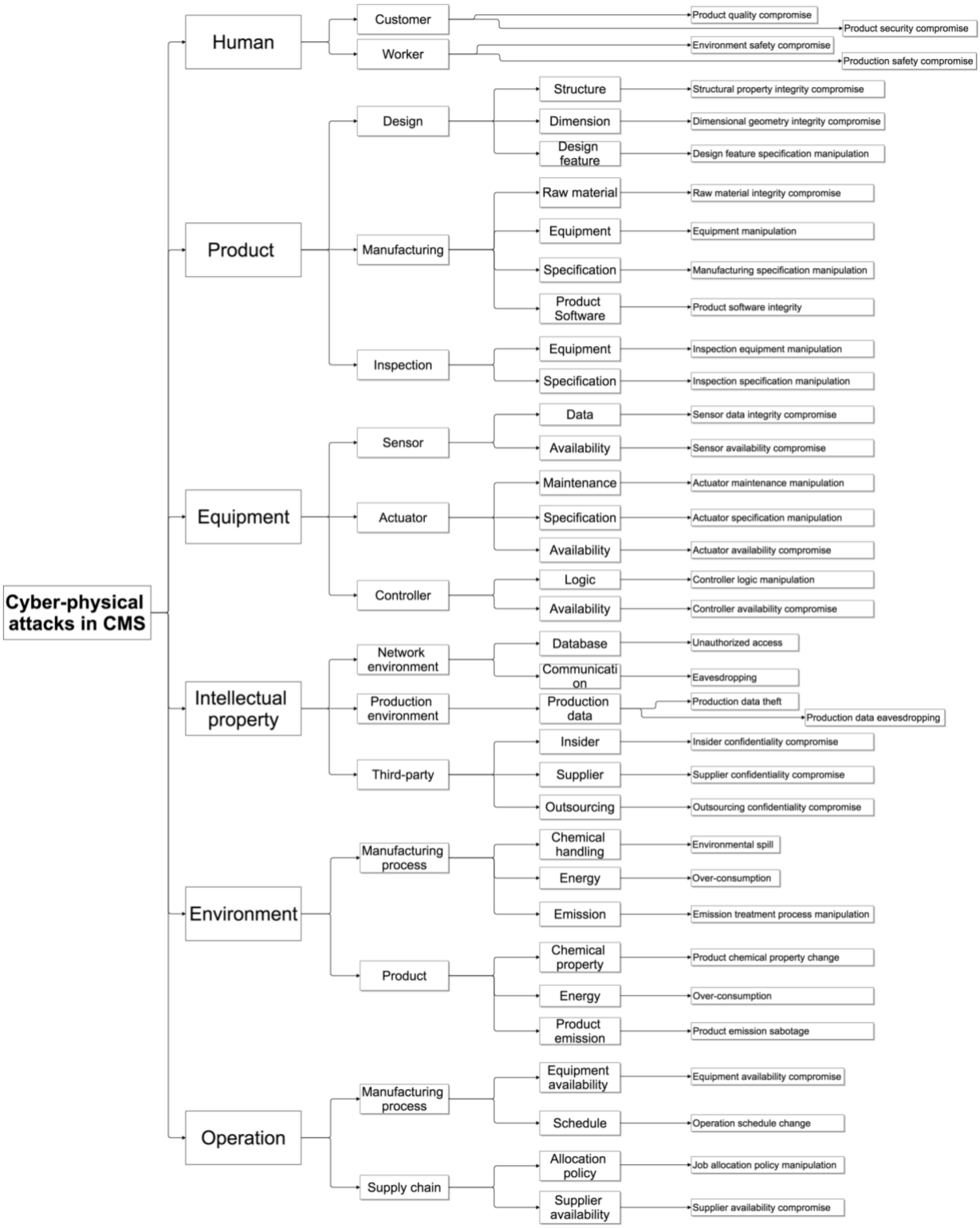


Figure 9 Cyber-physical attacks scenarios in CMS

Chapter 4

Alerts in Cyber and Physical Domain

In this chapter, the cyber and physical intrusion detection systems and their alerts are introduced. The cyber domain utilizes network and host-based intrusion detection software. The physical domain alert are generated by machine learning data analytics in the manufacturing process. The performance of each systems is analyzed.

4.1 Intrusion Detection Alerts in Cyber Domain

Cyber intrusion detection alerts can be generated by packages like Snort (Roesch 1999b), OSSEC (Karthikeyan and Indra 2010) when suspicious activities are detected. Intrusion detection alert in cyber domain includes both network-based and host-based intrusion detection system (NIDS & HIDS).

In this section, an intrusion detection alert format to review the information that can be utilized for alert correlation is introduced; how to generate cyber IDS alerts by using NIDS software snort and HIDS software OSSEC is explained; example of alerts generated in our experiment environment are shown.

4.1.1 Standard Format

The standard format for an intrusion detection alert is primarily for alert normalization: translate features of each sensor alert into a generic format for feature extraction and alert correlation. For cyber domain, there is well-established Intrusion Detection Message Exchange Format (IDMEF).

The Intrusion Detection Message Exchange Format (IDMEF) was proposed by Internet Engineering Task Force (IETF 2018). The purpose of the IDMEF is to define data formats and exchange procedures for sharing information of interest to intrusion detection and response systems and to the management systems that may need to interact with them (H. Debar, Curry, and Feinstein 2007).

An IDMEF alert message is composed of nine different components (Bhuyan, Bhattacharyya, and Kalita 2017):

Create Time: The time when the alert was generated.

Detect Time: The time when the event(s) leading up to the alert was (were) detected.

Analyzer Time: Current time on the analyzer.

Analyzer: Identification information for the analyzer that generated the alert.

Source: The source that triggered the alert.

Target: The main target of the alert.

Classification: Information that describes the alert.

Assessment: Impact, action and response against the generated alerts with evaluation.

Additional data: Additional information that does not fit into the data model.

By preprocessing, the attributes can be extracted for alert correlation. As shown below, is an example of IDMEF alert from ping-of-death attack:

4.1.2 Snort

In a computer network, network activity log data can be information, such as login attempts, network connections, or every data packet that appeared on the wire (Kemmerer and Vigna 2002). It can be monitored by Network based Intrusion Detection System (NIDS). For example, Software Snort is a packet sniffer that can monitor network traffic in real time. It checks each packet closely to detect a dangerous payload or suspicious anomalies.

Snort is an open source, lightweight, cross-platform software, originally developed by Martin Roesch in C language in 1998. It uses predefined rules for checking abnormal data in packet traffic (Khamphakdee, Benjamas, and Saiyod 2014). Snort is available for most operating systems and most major platforms, including Windows, Linux, MacOS, BSD and Solaris. It can generate alerts according to network activities in real time.

The software packages and detection rules can be downloaded from the Snort homepage <www.snort.org>. The rule can also be defined by computer security professionals. For example, **Figure 10** shows a basic snort rule. This rule will generate an alert when traffic from any port of

IP address 10.0.1.5 send to any destination IP address and destination port number is 53. It also will show message “DNS request” (Khamphakdee, Benjamas, and Saiyod 2014).

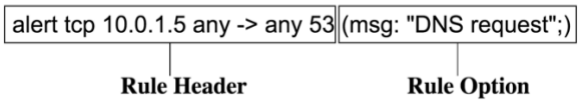


Figure 10 Snort Rule example

In **Table 3** is a full list of alerts defined by Snort 2.9.9.0. The alert list consists of the short name of alert, description and priority. Currently, there are four levels of priority, 1 stands for high, 2 stands for medium, 3 stands for low, and 4 stands for very low.

Table 3 Snort alert list

#	Name	Description	Priority
1	attempted-user	Attempted User Privilege Gain	1
2	unsuccessful-user	Unsuccessful User Privilege Gain	1
3	successful-user	Successful User Privilege Gain	1
4	attempted-admin	Attempted Administrator Privilege Gain	1
5	successful-admin	Successful Administrator Privilege Gain	1
6	shellcode-detect	Executable code was detected	1
7	Trojan-activity	A Network Trojan was detected	1
8	web-application-attack	Web Application Attack	1
9	inappropriate-content	Inappropriate Content was Detected	1
10	policy-violation	Potential Corporate Privacy Violation	1
11	file-format	Known malicious file or file-based exploit	1
12	malware-cnc	Known malware command and control traffic	1
13	client-side-exploit	Known client side exploit attempt	1
14	bad-unknown	Potentially Bad Traffic	2
15	attempted-recon	Attempted Information Leak	2
16	successful-recon-limited	Information Leak	2
17	successful-recon-largescale	Large Scale Information Leak	2
18	attempted-dos	Attempted Denial of Service	2
19	successful-dos	Denial of Service	2

20	rpc-portmap-decode	Decode of an RPC Query	2
21	suspicious-filename-detect	A suspicious filename was detected	2
22	suspicious-login	An attempted login using a suspicious username was detected	2
23	system-call-detect	A system call was detected	2
24	unusual-client-port-connection	A client was using an unusual port	2
25	denial-of-service	Detection of a Denial of Service Attack	2
26	non-standard-protocol	Detection of a non-standard protocol or event	2
27	web-application-activity	access to a potentially vulnerable web application	2
28	misc-attack	Misc Attack	2
29	default-login-attempt	Attempt to login by a default username and password	2
30	sdf	Sensitive Data	2
31	not-suspicious	Not Suspicious Traffic	3
32	unknown	Unknown Traffic	3
33	string-detect	A suspicious string was detected	3
34	network-scan	Detection of a Network Scan	3
35	protocol-command-decode	Generic Protocol Command Decode	3
36	misc-activity	Misc activity	3
37	icmp-event	Generic ICMP event	3
38	tcp-connection	A TCP connection was detected	4

Studying the alert instead of network activity data is a process of feature extraction and dimensional reduction for network activity data.

07/10-17:01:30.096292 [**] [1:399:6] ICMP Destination Unreachable Host Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 10.0.2.2 -> 10.0.2.15

Figure 11 Snort Alert Example

Shown in **Figure 11** is an example of snort alert. The key information of an alert is: alert priority level, alert time, and description.

4.1.3 OSSEC

A network host is a computer or other device connected to a computer network. A network host may offer information resources, services, and applications to users, or other nodes, on the network. It can be monitored by a host based intrusion detection system (HIDS). For example, Software OSSEC can do log analysis, file integrity checking, Windows registry monitoring, centralized policy enforcement, rootkit detection, real-time alerting and active response (Timofte 2008). In section 2.1.2, the implementation of OSSEC is introduced, along with introduction of other similar HIDS.

OSSEC is an open source, multi-platform, scalable host-based intrusion detection system (HIDS). It can run on most operating systems, such as Windows, Linux, MacOS, OpenBSD, FreeBSD and Solaris (Timofte 2008). It analyzes host log, file, windows registry and gives real-time alert response.

The OSSEC can be installed as a stand-alone tool to monitor one host, or can be deployed in a multi-host scenario. In CMS, one installation acts as the IDS monitoring server and the others as agents in different layers of CMS.

Similar to Snort, OSSEC gives alerts with a number representing its priority. Different from Snort, the OSSEC uses ascending order instead of descending. Moreover, OSSEC has 15 different levels of severity, as shown in **Table 4**: alert level from OSSEC 2.8.1 rules classification.

Table 4 OSSEC alert examples

Alert Level	Action	Description
0	Ignored	No action was taken. Used to avoid false positives. These rules are scanned before all the others. They include events with no security relevance.
1	None	-
2	System low priority notification	System notification or status messages. They have no security relevance.
3	Successful/Authorized events	They include successful login attempts, firewall allow events, etc.
4	System low priority error	Errors related to bad configurations or unused devices/applications. They have no security relevance and are usually caused by default installations or software testing.
5	User generated error	They include missed passwords, denied actions, etc. By itself they have no security relevance.
6	Low relevance attack	They indicate a worm or a virus that have no effect to the system (like code red for apache servers, etc). They also include frequently IDS events and frequently errors.
7	"Bad word" matching	They include words like "bad", "error", etc. These events are most of the time unclassified and may have some security relevance.
8	First time seen	Include first time seen events. First time an IDS event is fired or the first time an user logged in. If you just started using OSSEC HIDS these messages will probably be frequently. After a while they should go away, It also includes security relevant actions (like the starting of a sniffer or something like that).
9	Error from invalid source	Include attempts to login as an unknown user or from an invalid source. May have security relevance (specially if repeated). They also include errors regarding the "admin" (root) account.
10	Multiple user generated errors	They include multiple bad passwords, multiple failed logins, etc. They may indicate an attack or may just be that a user just forgot his credentials.
11	Integrity checking warning	They include messages regarding the modification of binaries or the presence of rootkits (by rootcheck). If you just modified your system configuration you should be fine regarding the "syscheck" messages. They may indicate a successful attack. Also included IDS events that will be ignored (high number of repetitions).

12	High importancy event	They include error or warning messages from the system, kernel, etc. They may indicate an attack against a specific application.
13	Unusual error (high importance)	Most of the times It matches a common attack pattern.
14	High importance security event.	Most of the times done with correlation and it indicates an attack.
15	Severe attack	No chances of false positives. Immediate attention is necessary.

As shown in **Figure 12**, an OSSEC alert shows that an important system file size changed, which is an integrity alert. Potentially it could be changed by an intruder for getting user or superuser privilege.

```
** Alert 1499713141.34392: mail  - ossec,syscheck,
2017 Jul 10 11:59:01 ubuntu->syscheck
Rule: 550 (level 7) -> 'Integrity checksum changed.'
Integrity checksum changed for: '/etc/php5/apache2/php.ini'
Size changed from '68428' to '68429'
Old md5sum was: 'a0ed8c3fc8bcf0d41efaeb5bc53eb98e'
New md5sum is : '4ed8aa5fcd256def07178fae0a5f8b00'
Old sha1sum was: 'ca9fb4ae0334a6735370ca7f56947665c6a8d8a8'
New sha1sum is : 'b27c5bb340415fc967d3ae5440be84e8e869cd3c'
```

Figure 12 OSSEC alert example

Similar to a network data process, the OSSEC transfers the host data into alerts data. By monitoring the OSSEC alert's time, level and description, adminstrator can make decisions in intrusion detection.

4.1.4 Alert Generation

To observe the alert generation during an attack, an experiment is presented to simulate the cyber-physical attack. The experiment is designed based on the CMS testbed (Wu et al. 2018) with both cyber and network environments.

The experiment cyber environment comprises a data host equipped with Ubuntu 14.04 operating system (with magic quote function turned off to be vulnerable to SQL injection attack),

a user data based on MySQL 5.7 and Apache HTTP Server 2.4, and a website application for customers front-end.

The Snort is equipped with standard rule along with additional SQL injection rules as follows.

Snort SQL Injection Local Rules

```
alert tcp any any -> any 80 (msg: “Error Based SQL Injection”; content: “%27” ; sid:100000011; )
alert tcp any any -> any 80 (msg: “Error Based SQL Injection”; content: “22” ; sid:100000012; )
```

The three factors in the experiment are: normal customer activity, SQL injection attack, and false alarm noise by NMAP software (Orebaugh and Pinkard 2011) network scan:

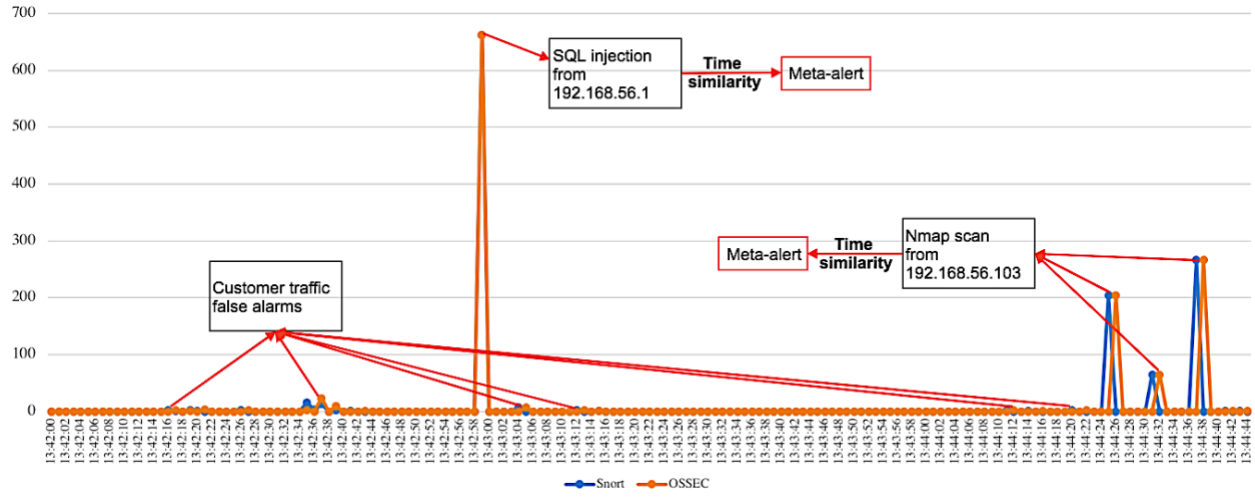
Normal customer activity. Students simulated customers used computer visiting customer front-end website, and created events such as login, uploading orders, deleting orders, editing orders and logging out.

SQL injection attack. Students simulated hacker used commands such as “*UID_xxxx';* -- ” or “*' or 1=1;* --” directly accessing into customer account or administrator account without knowing the password. Such an act could trigger alerts from Snort software.

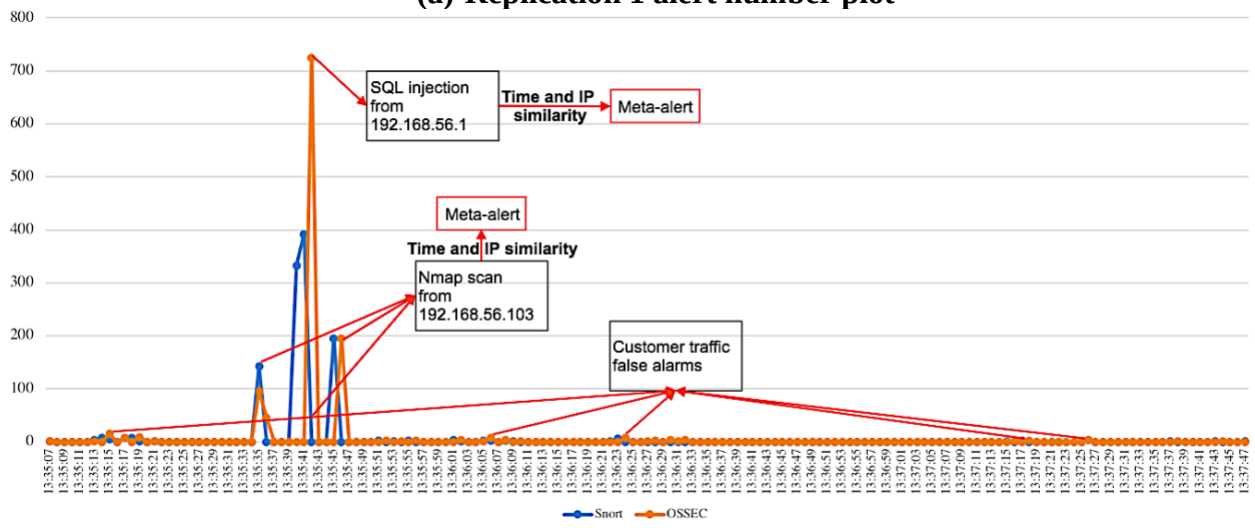
NMAP network scan. Students simulated hackers used NMAP intense scan on customer website and database host to create false alarms.

In replication one as shown in **Figure 13** (a), the customers randomly visit the front-end website during the whole process and caused minor false alarms. The SQL injection in the middle of the experiment and caused over 600 alarms. Those alarms can be correlated via time similarity directly. The Nmap scan at the end of the experiment and caused three alarm-peaks at around 200 counts, with a total number of 600 alarms. Those alarms can be correlated as second meta alert with high priority.

In replication two as shown in **Figure 13** (b), the customers are regularly visiting and causes similar minor false alarms. In this replication, the SQL injection and Nmap scan happen at the same time. To correlate and create meaningful meta-alerts, IP address and time can correlate alerts caused by the same attack. The two replications use randomized attack pattern to prove the effectiveness of similarity-based correlation methods under different circumstances.



(a) Replication 1 alert number plot



(b) Replication 2 alert number plot

Figure 13 Snort and OSSEC experiment alert number plot

4.2 Physical Alerts

For physical domain, a Physical Intrusion Detection Alert (PIDA) format is proposed for the purpose of information exchange for alert correlation in a cyber-physical manufacturing system.

4.2.1 Physical Intrusion Detection Alert (IPDA)

The physical alert is new and not standardized yet for cyber-physical intrusion detection system. They are generated by analyzing audit data from pre-production, in-production and post-production stages. In the production process, the physical alert will be continuously generated until the abnormal production pattern is paused, finished or return to normal. In a 15-minutes malicious CNC milling process, two to three hundred alerts could be generated with real-time data analysis.

As a result, a physical intrusion detection alert (PIDA) format is proposed as shown in **Table 5** to provide vital information for alert correlation. Different from IDMEF, or other IDS format, PIDA embodies information from the physical domain. The key information provided by the physical alert format including:

Create Time: The time when the physical event caused the alert is generated.
Analyzer Time: The time when the alert is generated.
Sensor ID: The physical sensor/inspection station collected alert data.
Analyzer ID: The name of analyzer generated alert.
User ID: The user identification that triggers the alert.
Order ID: The product identification that triggers the alert.
Equipment ID: The identification of the equipment where the alert happens.
Supplier ID: The identification of the CMS service provider.
Manufacturing Process: The general manufacturing process the equipment belongs to.
Additional Information: The information can be added by the operator or administrator.

Table 5 PIDA alert

<PIDA-Message_873642>	#Alert message title and ID
<Create_Time_2018-06-13 11:16:10.817137>	#Create Time
<Analyze_Time_2018-06-13 12:01:01>	#Analyzer Time
<Ultrasonic_Sensor_1_1>	#Sensor ID
<KNN_classifier_k_1_feature_12>	#Analyzer ID
<UID_976378452>	#User ID
<Order_20180708_CNC16_T1>	#Order ID
<CNC_Milling_1>	#Equipment ID
<SupID_72654213>	#Supplier ID
<Metal_Subtractive_Mill>	#Manufacturing Process
<Cause_tardy_job_and_equipment_damage>	#Additional Information

4.2.2 Machine Learning Based Physical Intrusion Detection

Machine learning has been intensively applied both in physical security data and manufacturing system, but not in manufacturing security so far. Physical security data needed for machine learning can come from voice recognition, fingerprint authentication, gait authentication, keystroke and other biometrics (Jain, Ross, and Prabhakar 2004). Machine learning implementations in manufacturing includes real-time vision system for surface defect detection (Jia et al. 2004), weld defect defection (Shen, Gao, and Li 2010), surface defect detection (X. W. Zhang et al. 2011), preventative maintenance, supply chains optimization, etc.

The integration of cyber security and physical data machine learning is an approach to detect cyber-physical attacks. It can effectively enhance the accuracy and shorten the respond time. The cyber security approaches have been intensively researched in the past and can be implemented with IT security professionals. At the same time, the machine learning approach utilizing physical data can filter the false alerts from cybersecurity aided by domain experts from manufacturing.

4.2.2.1 Supervised Learning: Classification

Classification is a supervised machine learning method with the purpose of categorizing data sets. In machine learning, classification is implemented with various algorithms, also known as classifier, such as Support Vector Machine (SVM), C4.5 decision tree, artificial neural network (ANN), k-Nearest Neighbors, etc. Data sets for classification are pre-processed and analyzed to features. The process to define feature is a key process to enhance accuracy in machine learning results, called feature extraction which requires domain knowledge with data mining experience.

In this research, image and acoustic classifications have been used to detect malicious attacks in CMS processes. Random forest, k-nearest neighbors (kNN) machine learning algorithms have been implemented. *k-Nearest Neighbors (kNN)* classifier is used to perform discriminant analysis when reliable parametric estimates of probability densities are unknown or difficult to determine (Peterson 2009). *A random forest multi-way classifier* consists of a number of trees, with each tree grown using some form of randomization. The leaf nodes of each tree are labeled by estimates of the posterior distribution over the image classes. Each internal node contains a test that best splits the space of data to be classified (Bosch, Zisserman, and Munoz 2007). In this research, three decision trees are used and each of them has five leaf nodes to classify (Wu et al. 2017). Compared to C4.5 decision tree algorithm, the random forest classifier achieves higher accuracy with relatively shorter time to execute.

4.2.2.2 Unsupervised Learning: Anomaly Detection

Anomaly detection can identify abnormal behavior on a host or network (Kim, Park, and Lee 2013), image (Chandola, Banerjee, and Kumar 2009b), supervisory control and data acquisition (SCADA) (Garcia, Rolle, and Castelo 2011), or for equipment preventive maintenance

(Rabatel, Bringay, and Poncelet 2011). It refers to the problem of finding patterns in data that do not conform to expected behavior (Chandola, Banerjee, and Kumar 2009a). The principle is to recognize patterns of accepted behavior, which is learned or specified by the algorithm. Activities that fall outside the predefined or accepted model of behavior will alert administrators. The advantage of anomaly detection is that it can detect novel attacks comparing to supervised approaches. However, the disadvantage of network anomaly detection is the difficulty in defining rules for normal network behavior.

Since it is impossible to predict every possible attack that a hacker may try against CMS system, the anomaly detection method is implemented and combined with the random forest method to increase the accuracy.

4.2.2.3 Data in CMS Environment

To implement machine learning in CMS security, data/signal processing and feature selection and extraction are key steps. Data sources can be used including vision, acoustic, energy, temperature, weight, etc. Some of the data can be directly drawn from controlling system whereas others need additional monitoring systems.

CMS processes can consist of traditional and advanced manufacturing processes. They include additive manufacturing, subtractive manufacturing, molding, forming, joining, casting, coating, high-speed assembly and others. In this research, 3D printing and CNC milling processes were used as two examples.

To decide what data to extract from the manufacturing process for security purposes, the following factors should be analyzed: i) what is the process and what is the attack aim, ii) what is the symptom and consequence, and iii) what data can be collected from the machine for detection.

3D printing is a key enabling technology for CMS. It is getting extensively popular in recent years, and some new machines are developed with wireless network capability, which also increases the attack surface for a successful attack. The attack aims for 3D printing could be: change the design dimensions, change the infill with malicious void, change nozzle travel speed, or change heating temperature. The symptom could be quite implicit, such as a hidden void, surface gap or high energy consumption, and finally, leads to scrap parts. For 3D printing, vision, acoustic and energy consumption could be potential features.

Computer Numerical Control (CNC) milling process is a representative process for subtractive manufacturing process. The attack can aim for CNC milling process to alter design, spindle speed, or feed speed. The design change can create scrap parts. The increase in spindle speed can accelerate tool wear. Also, the increase in feed speed can break cutting tools. For CNC milling, acoustic, temperature and time can be potential features.

Table 6 CMS process attacks analysis and data extraction

Process	Attack aim	Symptom	Consequence	Detection Data
3D printing	Design Infill Nozzle travels speed Heating temperature	Hidden void Surface gap High energy consumption	Scrap parts Overheating	Vision Energy consumption Acoustic
CNC milling	Design Spindle speed Feed speed	Change in vibration Change in chip shape Cutting bit temperature Tool breakage	Scrap parts Overwear Tool breakage Overheating	Acoustic Temperature Time

4.2.2.4 Feature Extraction

For machine learning in manufacturing, feature extraction is a critical process. It starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. A feature is a good data representation of a symptom, phenomenon or measurement. For example, high value of acoustic emission during drilling process can mean wrong spindle speed or wrong part material. The feature extraction process requires domain knowledge and data processing experience.

4.2.3 Additive Manufacturing Process: a 3D Printing Example

3D printing, or additive manufacturing, is a key technology for advanced manufacturing systems (Wu et al. 2016). However, 3D printing systems have unique vulnerabilities presented by the ability to affect internal layers without affecting the exterior layers (Sturm et al. 2014). By changing design or dimensions in the ".STL" file, malicious defective parts could be manufactured without any prior alert.

4.2.3.1 Attack Mode

Man-in-the-middle attacks can easily accomplish the process of replacing an original ".STL" file with a malicious design ".STL" file. As shown in **Figure 14**, during the user's uploading original ".STL" file to manufacturing server to put an order, an attacker can alter the communication between user and server, and replace with malicious ".STL" file.

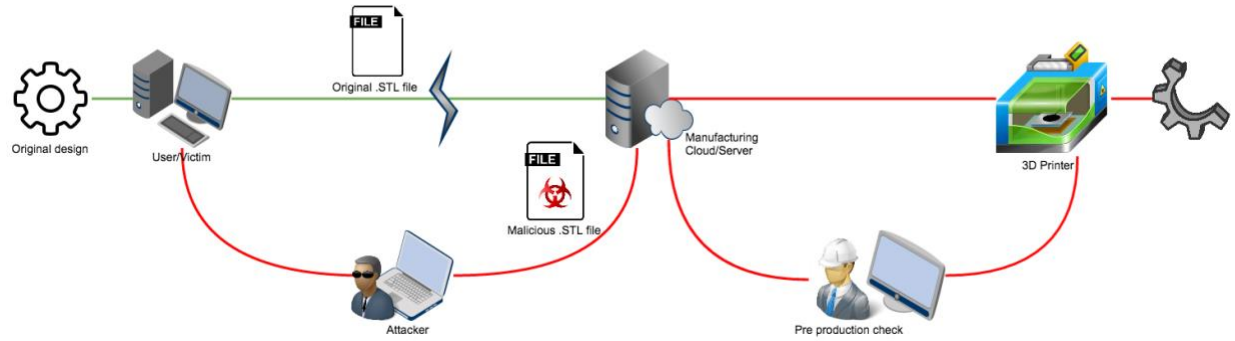


Figure 14 Man-in-the-middle attack for a cyber-based 3D printing process

If a hacker designed a malicious infill void defect that cannot be observed from the surface of the final product, the part will be manufactured without noticing any abnormalities. During the pre-production check process, operators cannot detect the difference between the original design and malicious design because the malicious design can be implicit. The malicious file will then be sent to 3D printers and the finished defective parts will be sent to the customers. As shown by Sturm (Sturm et al. 2014), the void in a 3D printing part will result in reduction of yield, with other corresponding physical characteristic changes such as weight, stiffness and natural frequency.

Five different infill defect patterns were designed as shown in **Figure 15**: Seam, Irregular Polygon, Circle, Rectangle, and Triangle to simulate attacks. The examples illustrated in **Figure 15** are parts with 10% honeycomb infill.

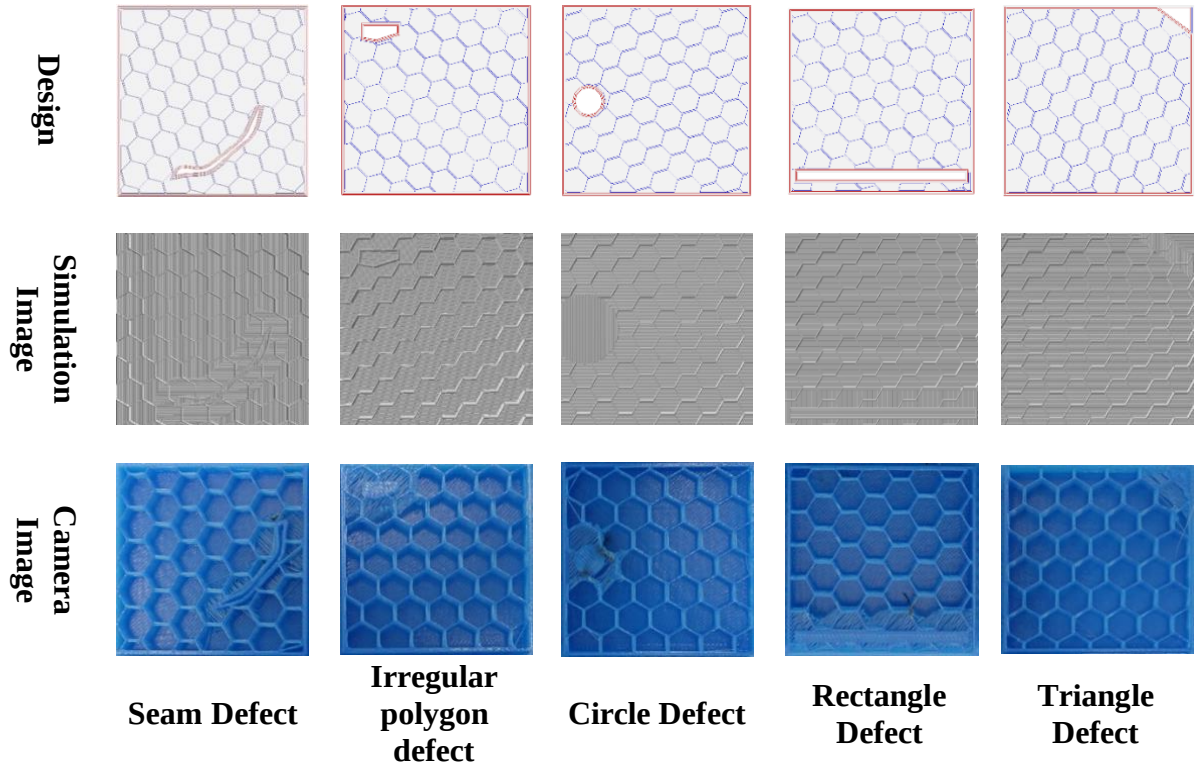


Figure 15 Malicious defect designs, simulation images and camera images

4.2.3.2 Data Collection: Image Simulation and Experiment

Images were captured from the 3D printing software MakerBot Desktop 3.9.1 preview function. The size of images is 512 x 512 pixels. The selection of image size was done in considering feature extraction process.

In total, 3887 simulation images were generated for simulation. 532 images of non-defect parts were captured, labeled as group A. The non-defect group A images were captured every 2 to 4 layers during the printing process, with infill density varied from 8%-12% to increase the diversity of the training images. 3355 images of defective parts were captured and labeled as group B. The defective group B images were captured every 2 to 4 layers during the printing process, with combinations of 5 different defects. The infill density is 10% for group B.

Another method used in images collection is to capture real images during printing process with mini cameras attached on 3D printer structures. To test and verify the image classification method in real environment, a camera-based vision detection system has been designed and installed on MakerBot Replicator™2. MakerBot Replicator™2 has the building envelope of 11.2 x 6.0 x 6.1" and can print at 100 µm per layer. Installation of the camera on a MakerBot Replicator 2 is shown in **Figure 16**. In this work, two ways to install cameras on MakerBot Replicator 2 were presented. One is mounting the camera right next to the extruder and move along with it, called 'moving camera.' The other is mounting the camera on the frame of the 3D printer, called 'static camera.' The ‘static camera’ can capture clear image and reach higher accuracy. The ‘moving camera’ should have same accuracy and can adapt to more conditions, without the blurring caused by motion.

The camera is an Arducam Mini Module Camera Shield with OV2640 2 Megapixels Lens, compatible with Arduino UNO Mega2560 Board. The camera unit dimensions are 3 x 2 x 1 inches, connected to the Arduino UNO via extended jumper wires. With programming in Arducam software, it can produce images any size scaling down from SXGA to 40×30 in jpeg format. As a result, the feature extraction process for previous 512x512 size images needed to be altered.

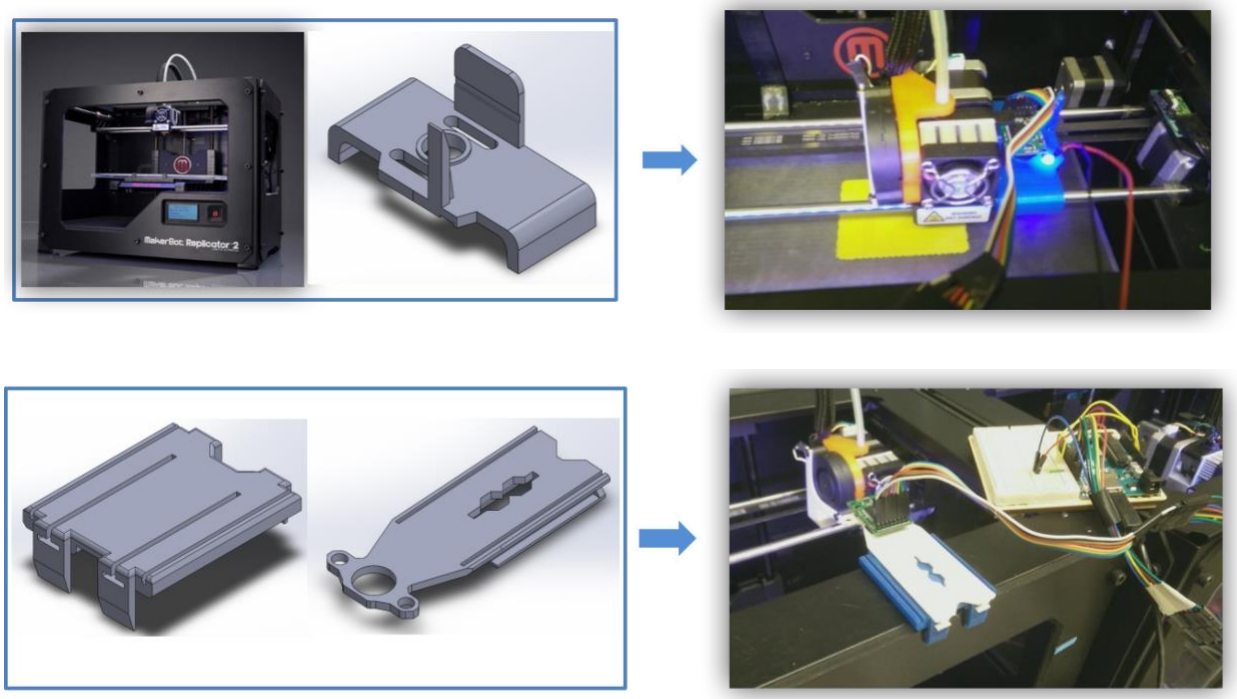


Figure 16 MakerBot Replicator 2 printer with moving camera and static camera

4.2.3.3 Feature Extraction

Feature extraction process is implemented via R 3.3.1 and RStudio Desktop 0.99.903.

By plotting simulated image row No. 250 (marked in red in **Figure 17**) grayscale value, repetitive peaks can be observed in normal area on the left, one medium peak followed by one high peak, in pairs. In defective area on the right, the greyscale plot shows constant volatility. To specify peaks, the threshold of grayscale is set at 120.

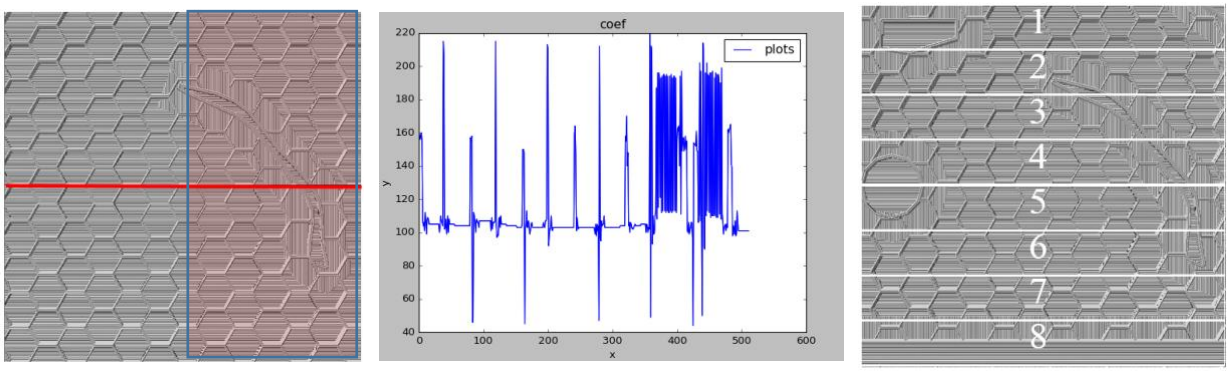


Figure 17 Grayscale Plot Row No. 250, section separation

For feature extraction, each image is equally divided into eight sections as shown in **Figure 17** Each section contains 64 rows, 32768 pixels. The following features are extracted for defect classification.

- Mean of grayscale in each section.
- Standard derivation of grayscale in each section.
- Number of pixels grayscale larger than 120.

As a result, every image has 24 features, from eight sections, each section provides three features (Wu et al. 2016).

Three machine learning algorithms are used in detecting malicious defect: k-Nearest Neighbors (kNN), random forest and anomaly detection.

4.2.3.4 Real-time Detection

A preliminary system was designed as **Figure 18**. to send real-time alert to administrator indicating malicious defect. The vision system on 3D printer is connected to Internet via Raspberry Pi B+. The camera is updated with Raspberry Pi OV 5647 Camera to be compatible with Raspberry Pi B+, and also improve the image quality.

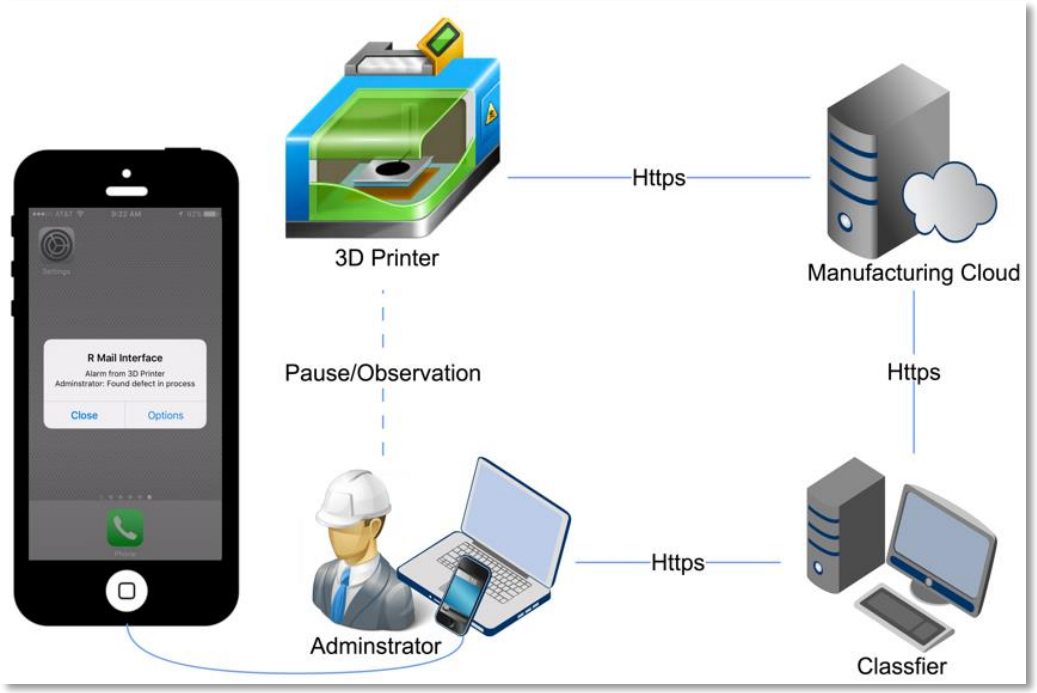


Figure 18 Preliminary wireless real-time alert system for 3D printing process

Raspberry Pi B+ is used as the mini-computer system to connect to the network and operate the Raspberry Pi OV5647 Camera to capture images of the printed object at a set time interval. Once the images are captured and saved to the Pi, BitTorrent Sync is used to synchronize the images from the device to the cloud service. The computer with classifier testing real-time collected images. If detected any malicious defects, the program will send an alert to the user via text message and email. As shown in **Figure 18**, the email says, "Alert from 3D printer, Administrator: Found defect in process".

After testing, the whole process can be accomplished within one minute, including the time for syncing and downloading images (largely depend on server and Internet speed) and feature extraction and classify time (within few seconds).

4.2.3.5 Result Analysis

The goal of this experiment is using machine learning and physical data from cameras to detect malicious defects. The accuracy of machine learning results is one of the measurements for effectiveness of the system. The machine learning accuracy is defined by the equation (1). Where *TruePositive* means images in class A that are predicted as class A, and *TrueNegative* stands for images in class B predicted as class B.

$$\text{Accuracy} = \frac{\text{TruePositive+TrueNegative}}{\text{Total}} \quad (1)$$

Moreover, the compatibility of the system is also tested by running with 5 different infill shapes of 3D printing process: Honeycomb, Diamond, Linear, Star, Catfill. Finally, the system effectively under real environment comparing to simulation is analyzed.

Table 7 3D printing process accuracy results

Accuracy		Machine Learning Method		
		Random Forest (%)	kNN (%)	Anomaly Detection (%)
Image from Simulation	Honeycomb	88.4	81.3	100.0
	Diamond	100.0	85.0	100.0
	Linear	94.6	92.5	100.0
	Star	97.8	100	99.8
	Catfill	91.5	100	100.0
Image from Moving Camera	Honeycomb	68.4	68.75	72.5
Image from Static Camera		95.5	87.5	96.1

As shown in Table 7:

1) Anomaly detection is most accurate method among three chosen methods in detecting malicious defects. Accuracy is 96.1% which is acceptable for the experimental result and can be improved by refinement in hardware and software.

2) Based on simulation images experiment, the different types of infill have a minor influence on system accuracy, but not critical.
3) Camera images have lower accuracy compared to simulated images. Among camera images, moving camera's final accuracy 72.5% is not acceptable because of the blur created by motion. Static camera images have a better accuracy of 96.1%, thus proves the system effectiveness in a real environment.

4.2.4 Subtractive Manufacturing Process: a CNC Milling Example

CNC machining is a typical subtractive processing. During the decades, CNC has been core manufacturing units in manufacturing systems. The flexibility and automation of manufacturing systems have been significantly enhanced by implementation of CNC machining. Since CNC processing could be totally manipulated by programming, it shows its vulnerability towards cyber-physical attacks.

4.2.4.1 Attack Mode

By implementing man-in-the-middle attack, attackers can replace original G-code designs with malicious G-code. Two attack scenarios have been developed as a result of malicious codes.

Scenario 1: Attack on Design

The first attacking scenario is to alternate the positioning parameters during processes and therefore change the profiling routine of tools. As a result, the geometric design will change. The change in tool path could cause assembly mistakes, structural weaken and possibly breakage. As shown in **Figure 19**, edge 2-3, 4-5, 5-6 and 6-7 offset inward the contour.

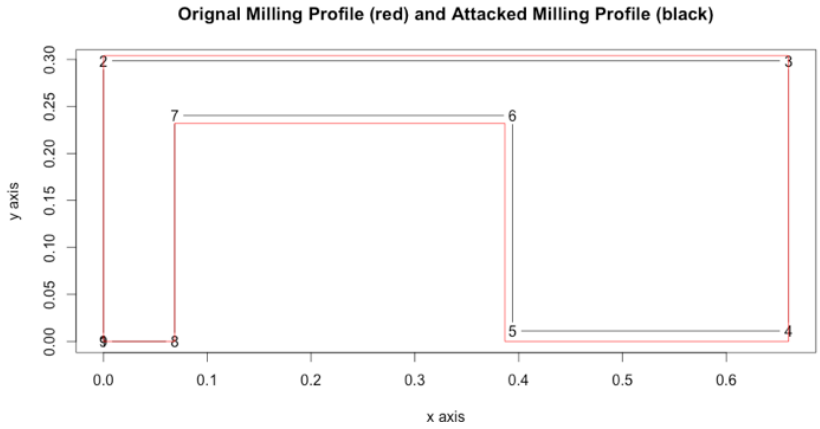


Figure 19 Comparison of Original and Attacked Milling Profiles

Scenario 2: Attack on Operation

The second attack mode proposed in this research is the change in machining operation parameters. In this section, a change in spindle speed in milling operation is captured for further research. In real case, fast rotation speed can cause over wear of tool; a tool with too slow rotation will risk in being broken by shear force in the feeding direction. In the scenario, spindle speed is maliciously altered from 1200 rpm to 2000rpm.

4.2.4.2 Data Collection: Acoustic Signal Simulation and Experiment

Acoustic signal is selected as the index to detect any malicious change in CNC milling process. Similarly, both simulation and experiment methods are adopted for testing.

Simulated signal is a time-serial amplitude numbers, created by a summation of sine-functions with fundamental frequency, harmonic frequencies and a Gaussian noise. The advantage of adopting simulated signal in this scenario is to enhance variety of signals for test and analysis with more parameters setting, and generate enough data for further analysis. The parameter used for acoustic signal generation are listed in Table 8, and the simulated signals were generated in R.

Table 8 Simulation Signal Parameters

Parameter	Value
Fundamental Frequency	40*Hz*
Harmonic Frequency	80*Hz*, 120*Hz*, 160*Hz*, 320*Hz*
Normalized Amplitude	0.3 for milling exterior boundaries; 1 for milling interior boundaries
White Noise	0.1**N*(0,1)
Acquisition Frequency	100 *Hz*

The experiments were conducted on a CNC machine Bridgeport Milling Ez-trak. The milling tool is a 2-flute, 3/16 end mill with rotation speed of 1200 rotation per minute. The material of work piece was aluminum. Moving speed of the tool was 10 inch per minute. Feed rate was 50/1000 of 1 inch for the first six milling cycles, 20/1000 of 1 inch for the last cycle.

According to (Delio, Tlusty, and Smith 1992; Duro et al. 2016), microphone provides the best balance in satisfying the many requirements of a sensor for recording acoustic signal in milling operations. Three microphones from smartphones: iPhone 5s microphone, iPhone 6s plus microphone and iPhone 6s with ear pod microphone were implemented as the acoustic sensors to recording signals.

The sample part is designed as **Figure 20**.

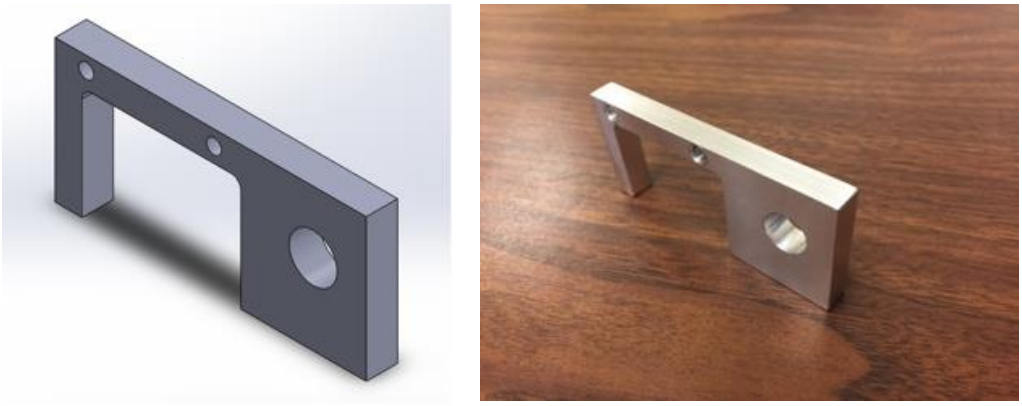
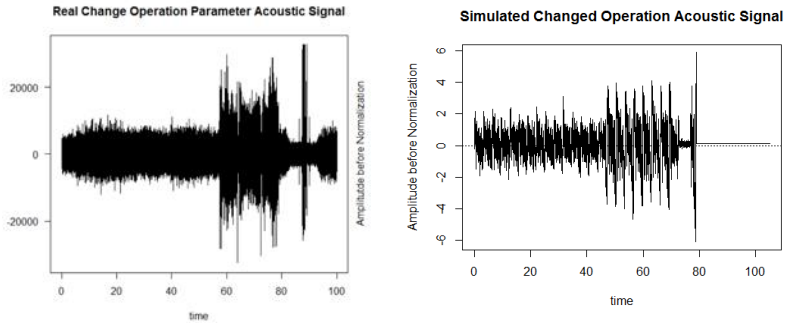


Figure 20 Sample part

4.2.4.3 Feature Extraction

The monitored signals were digitalized by MATLAB software. All the sound signal data were pre-processed by sectioning the whole period into sound periods of each individual cycle. In order to increase the number of the training data set, the real sound signal data were also sampled by 10 observations each. R was used for machine learning programming. The packages used for sound wave editing and analysis are “tuneR” and “seewave”. The packages used for machining learning detection and analysis are “randomForest”, “h2o” and “pROC”.



(a) Real Changed Operation Signal (b) Simulated Changed Operation Signal

Figure 21 Plot of Sound Wave in Attacked Scenario 2

According to the simulated and recorded signal, three key features is selected.

- Mean of amplitude in each period of time.
- Standard derivation of amplitude in each period of time.
- Number of points amplitude larger than threshold.

In experiment, period of time is set as 80 seconds, threshold for simulation is set as 1000, threshold for experiment signal is set as 2.5.

Similar to section 4.1, three machine learning algorithms are used in detecting malicious defect: kNN, random forest and anomaly detection. The real time synchronizing system can be implemented as section 4.1.4.

4.2.4.4 Result Analysis

Accuracy is the key measurement for detecting effectiveness as defined in section 4.1.5. The results of detecting malicious defects in CNC milling process via acoustic signal shown as Table 9.

Table 9 Machine learning accuracy for CNC milling process

Accuracy		**Machining Learning Method**		
		kNN (%)	**Random Forest (%)**	**Anomaly Detection (%)**
Simulated Signal	Scenario 1	50	93.1	93.8
	Scenario 2	50	100	100
Real Signal	Scenario 1	70	82.2	79.6
	Scenario 2	77.8	100	100

As shown in Table 9, anomaly detection and random forest method hold high accuracy for both scenario 1 and 2 in simulated signal, and scenario 2 in real signal. In real signal, the random forest shows highest average accuracy of 91.1%; Scenario 1 shows a slightly lower prediction accuracy comparing to scenario 2; Real signal has lower accuracy in scenario 1 than simulated signal, the reason could be the background noise from recording environment, and also the complexity of scenario 1 attack mode.

4.2.5 Alert Generation

The physical environment comprises manufacturing processes and data auditing and analyzing system. The manufacturing processes consist of a 3D printer, a CNC milling machine, two robotic arms, a conveyor, and a heating chamber, an Automated Guided Vehicle (AGV). The physical data is collected by sensors and analyzed as the source of physical alert. At least two types of sensor are used on each machine/process for the security and alert accuracy.

Table 10 Physical data auditing list

Equipment	Sensor #1	Sensor #2
3D Printer	Power Meter	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

As shown in Table 10, power meter and camera are used for data collecting for 3D printing process. For CNC milling machine, two accelerators and an acoustic sensor are used for data collection. For moving robotic arm, an avoidance sensor and a current sensor are used for data collection. For welder robotic arm, a camera and a current sensor are used for data collection. For conveyor, an acoustic sensor and a current sensor are used for data collection. For the heating chamber, a temperature and a current sensor are used for data collection. For AGV, an accelerator and an ultrasonic sensor are used for data collection.

The physical flow of victim manufacturing process starts with CNC milling, followed by conveying, heat treatment and transporting. It is manipulated via (1) changing spindle speed and the feed speed of CNC to create poor finish and (2) change heat treatment heating speed with to cause overload. The Figure 22 shows the time-series data of the CNC microphone, and accelerometer X axis, heat treatment current sensor, and temperature sensor (temperature data trend inversed).

The physical consequence in the data can be detected by machine learning and alerted quickly (Wu, Song, and Moon 2019; Wu et al. 2017). Each type of data is processed by separate analyzers and given an alert if any malicious defect discovered. Those four alerts can be correlated via time and sensor similarity and create further two physical meta-alerts.

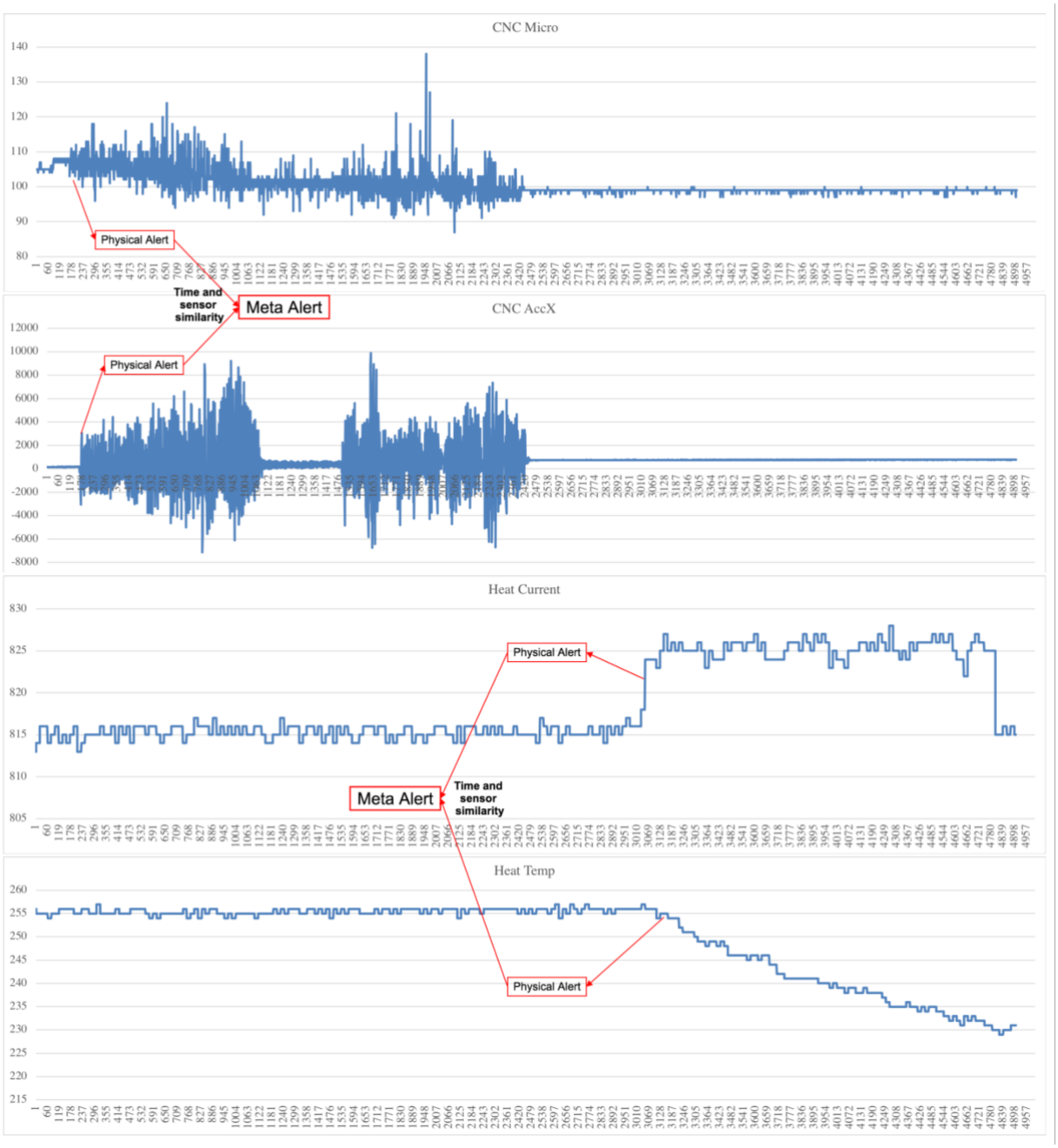


Figure 22 Physical domain alert correlation

Chapter 5

Cyber-Physical Alert Correlation Methodology

In this chapter, the cyber-physical alert correlation method discovers and establishes the mutual causal relationships between cyber and physical alerts from same or different sources. Once the relationship is established, it creates a high-level alert. This high-level cyber-physical meta-alert is a set of correlated alerts at a high level of abstraction and provide a succinct view of the intrusions (Valeur et al. 2004). It can help to trace to the origin of the attack (Bhuyan, Bhattacharyya, and Kalita 2017) and improve the accuracy of cyber-physical attack detection (Abad et al. 2003). A similarity-based correlation method for CMS cyber-physical alert correlation has been developed with a new physical alert format for reporting physical alerts.

As shown in **Figure 23**, the cyber-physical alert correlation method has three core components: (i) correlating similar cyber alerts to cyber meta-alerts, (ii) correlating similar physical alerts to physical meta-alerts, and (iii) correlating similar cyber and physical meta-alerts to cyber-physical meta-alerts. **Table 11** shows the notations used in this section.

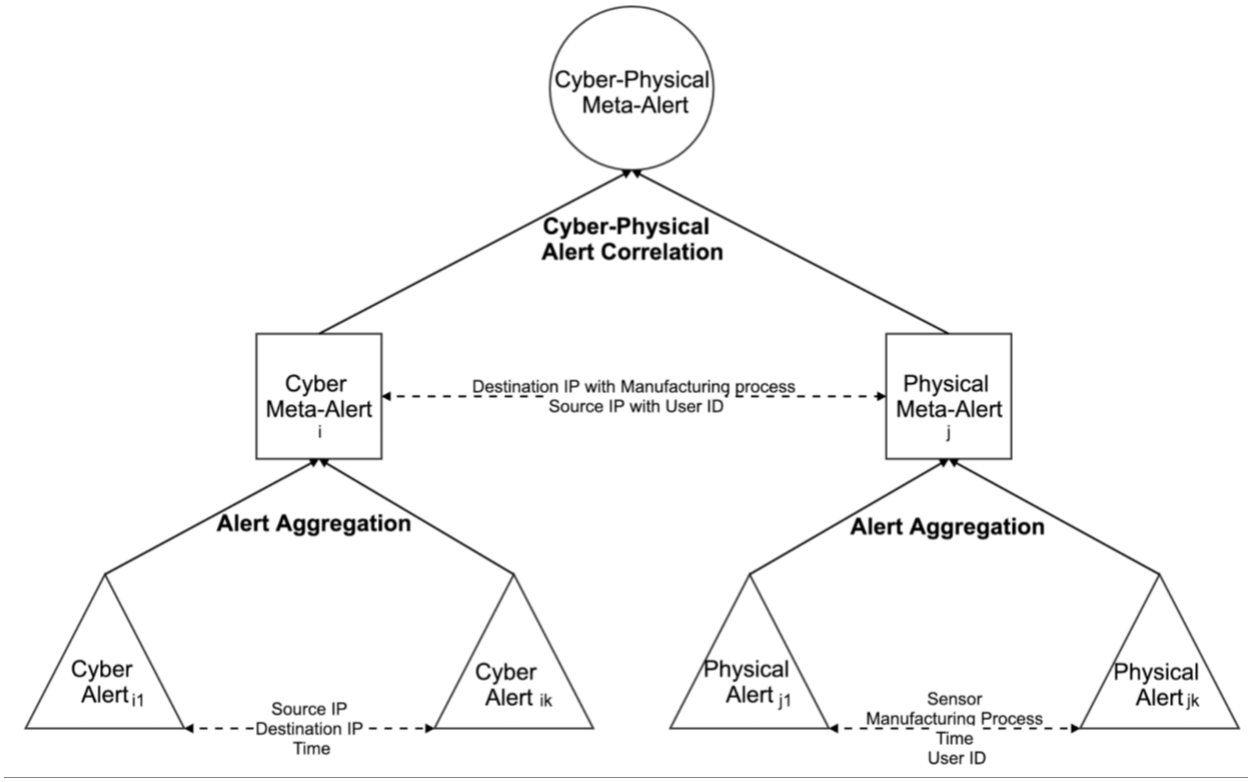


Figure 23 Cyber-physical alert correlation method

Table 11 Notations used in this section

Parameters	Definition
a	Last byte of the IP address of a four-byte IP address.
b	Second to last byte of a four-byte IP address.
c	Third to last byte of a four-byte IP address.
d	First byte of the IP address of a four-byte IP address.
IPsim_a	IP similarity score for the last byte of an IP address.
T^C_{create}	Create time of an intrusion detection alert.
T_{window}	Pre-defined correlation time window.
T^C_{term}	Termination time of a correlation period.
T^m_{create}	Create time of a meta-alert.

T^{a}_{create}	Create time of newly listed intrusion detection alert α.
T^{m}_{term}	Termination time of a meta-alert with more than 2 alerts.
$Sensor_{sim}$	Sensor similarity value.
SID	Sensor ID.
SID_{sim}	Sensor ID similarity.
SID^{k}_{max}	The maximum sensor ID of type K sensor.
EID_{sim}	Equipment ID similarity.
d_{SID}	Euclidean distance of sensor ID.
$d^{threshold}_{SID}$	d_{SID} threshold to determine alert from same type of sensor.
MP_{sim}	Manufacturing process similarity.
MPID	Manufacturing process ID.
T_{job}	Manufacturing job length time.
T_{end}	Manufacturing job end time.
UID	User identification/ user ID.
SIP	Source IP address.
DIP	Destination IP address.
K	Number of the cyber-physical meta-alert.
n_{cm}	Number of cyber meta alert.
n_{pm}	Number of physical meta alert.
n_{p}	Number of single physical alert.
n_{c}	Number of single cyber alert.
p	Total reduction rate.

To realize each function, the temporal- and attribute-based similarity analyses are defined separately. Furthermore, a physical intrusion detection alert (PIDA) format is defined for reporting and correlating physical alerts.

5.1 Cyber alert correlation

The cyber alerts derive from the intrusion detection system in CMS cyber domain. The source of alerts can be one or multiple host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS). For example, a NIDS software Snort (Roesch 1999a) and a HIDS software OSSEC (Karthikeyan and Indra 2010) are implemented as the cyber alert source in our experiments. As shown in **Table 12**, alerts are generated by Snort and OSSEC once a SQL injection attack (Clarke and Alvarez 2012) is detected by the software.

As shown in **Figure 23**, the cyber-alert correlation utilizes both attribute-based and temporal-based similarity analyses. Specifically, the attributes are source IP addresses and destination IP addresses. The temporal analysis utilizes the creation time of cyber alerts.

Table 12 Snort and OSSEC alert

Snort Alert [**] [1:100000011:0] Error Based SQL Injection [**] [Priority: 0] 07/06-09:33:00.631608 192.168.56.1:52938 -> 192.168.56.102:80 TCP TTL:64 TOS:0x2 ID:0 IpLen:20 DgmLen:715 DF ***AP*** Seq: 0x7EC4E75E Ack: 0xCD74DE37 Win: 0x1015 TcpLen: 32 TCP Options (3) => NOP NOP TS: 668492138 1130267 OSSEC Alert ** Alert 1530883982.4621: - ids, 2018 Jul 06 09:33:02 ubuntu->/var/log/snort/alert Rule: 20101 (level 6) -> 'IDS event.' [**] [1:100000011:0] Error Based SQL Injection [**]

5.1.1 Source IP Similarity

The IP address is one of the most common features used in the similarity-based correlation method in alert management research. If two alerts contain the same Source IP, then they are more likely to be under same attack source to be correlated (M Kumar, Siddique, and Noor 2009). For example, Valdes and Skinner (Valdes and Skinner 2001) compared higher bits of IP addresses for estimating its similarity. There are limitations that source IP addresses may be spoofed, and using IP alone may not provide a sufficient measure to classify the threat posed by an alert (Smith et al. 2008). But utilizing features such as time windows in Section 3.1.3 can also reduce the effect of IP reassignment (Ahmadinejad and Jalili 2009).

To effectively evaluate the IP address similarity, the IP is compared in four bytes via the IP address similarity matrix shown in **Table 13**. Assume that existing alert has the IP of “192.168.56.1”, and the new alerts with various IP addresses are calculated as follows:

Table 13 IP address similarity matrix

New alert IP	Similarity
192 . 168 . 56 . 1	1
192 . 168 . 56 . a	$1 - \mathrm{IPsim}_a$
192 . 168 . b . a	$1 - \mathrm{IPsim}_a - \mathrm{IPsim}_b$
192 . c . b . a	$1 - \mathrm{IPsim}_a - \mathrm{IPsim}_b - \mathrm{IPsim}_c$
d . c . b . a	$1 - \mathrm{IPsim}_a - \mathrm{IPsim}_b - \mathrm{IPsim}_c - \mathrm{IPsim}_d$

Where the IPsim_x the sum of score should equal to 1:

$$\sum_{x=a}^{d} IPsim_x = 1 \qquad (1)$$

The different values of IPsim_x score can be defined by specialists for different situations to meet different needs. For example, the score can be evenly distributed: $\mathrm{IPsim}_x = 0.25$ to compare both higher and lower bits of IP address. The IP address with similar higher bits could be from same subnetwork. This score generates similar results to those reported in (Ahmadinejad and Jalili 2009). Another way is comparing IP address in all four bytes to find the exact match. If the two alert source IP are identical, the similarity will be 1. Otherwise, the similarity will be 0.

5.1.2 Destination IP Similarity and Host Segmentation

The destination IP similarity can be analyzed by the same method presented in Section 5.1.1. The difference between the source IP and destination IP means that the source IP comes for the user or potential attacker, while the destination IP stands for data host in CMS.

In CMS, the data host can be segmented based on various factors: manufacturing processes, geographic locations, customer type, etc. In this work, the manufacturing process were used as an

example. As a result, the alerts from different destination IP stands for different physical meanings. The benefit of such practice includes:

1) Correlate cyber alerts to physical alerts via manufacturing process/destination IP.
2) Improve performance in each segmented network and reduce congestion.
3) Improve security when one of the segmentation is compromised while others are isolated.
4) Improve security when one type of customer can only access to limited data resource.

5.1.3 Time Similarity

The time stamp from cyber alert provides valuable information time similarity analysis. It can correlate the alerts caused by the same attacks that triggered the IDS sensor within the same short period of time.

The time stamp may contain different information in different cyber alerts. In the Snort alert, the time stamp shows the create time of the alert. In the standardized Intrusion Detection Message Exchange Format (IDMEF) (H. Debar, Curry, and Feinstein 2007), the alert will contain three timestamps: create time, detect time, analyzer time. The create time T_{create}^{C} is the feature to analyze alert similarity.

Time window, or T_{window} is a pre-defined value. The length of the correlation time window affects the potential of creating correlations. The length could vary from seconds to several hours depending on the alert characteristics, and the value will emerge from the practice of managing a specific network (Jakobson and Weissman 1995). Only alerts occurring within a time-window are to be correlated.

The method to scientifically set up an optimum correlation window is an open problem. Theoretically, a large window time can include more alerts that can provide more helpful information for security analysts on the meta-alert. However, a large window time can also include false alarms and noise that can affect the correlation efficiency (Qin 2005). One of the methods

defines the T_{window} using pre-specified attack scenario time. The assumption is that different multi-stage attack strategies usually have their own attack behavior patterns and happen in a certain time span (Jie, Li, and Li 2008). As a result, the attacks occurred within scenario time span should be correlated as meta-alert.

The alerts to be correlated are from IDS existing alerts set, called candidate alerts. Every alert has a correlation lifespan between create time T^C_{create} and termination time T^C_{term}, defined as:

$$T^C_{term} = T^C_{create} + T_{window} \quad (2)$$

When a new cyber alert α is generated during or follows the candidate alarm, the candidate alert and the new alert can be correlated based on the temporal analysis.

$$T^C_{create} < T^\alpha_{create} \leq T^C_{term} \quad (3)$$

The meta-alert is a combination of two or more alarms, and the correlation lifespan of a meta-alert is defined as:

$$T^m_{create} = \min(T^C_{create}, T^a_{create}) \quad (4)$$

$$T^m_{term} = T^a_{create} + T_{window} \quad (5)$$

For cyber alerts, any new alerts generated within the time window of the candidate meta-alert, can be correlated to a new meta-alert. The temporal alert correlation can reduce the number of alerts generated by the same attacks and convert them into high-level alerts (Salah, Maciá-Fernández, and Díaz-Verdejo 2013).

5.2 Physical alert correlation

As shown in **Figure 23**, the physical alert correlation method also utilizes attribute-based and temporal-based similarity analysis. However, different from cyber alert, the attributes are sensor similarity, manufacturing process similarity and user ID similarity.

5.2.1 Sensor Similarity

The sensor similarity aims to correlate the similar symptom caused by the same attack. A meta alert correlated by sensor similarity could be caused by: (1) alerts from similar types of sensor on different machines; (2) alerts from different types of the sensor on the same machine.

The sensor similarity can be calculated by computing certain metrics, such as Euclidean distance functions. The result will be compared to a threshold value and determine whether to be correlated (Salah, Maciá-Fernández, and Díaz-Verdejo 2013). To compare the sensor similarity: (1) the alerts from similar type of sensor can be calculated via sensor ID, (2) the alert from different types of sensor on same machine can be calculated via machine ID.

As shown in equation 6, the sensor similarity value calculates via both sensor and machine ID similarity, then use the larger value as the sensor similarity.

$$Sensor_{sim} = \max(SID_{sim}, EID_{sim}) \quad (6)$$

The sensor ID similarity SID_{sim} is calculated by comparing d_{SID} between the existing alerts set's sensor IDs SID_{set} and the new alert sensor ID SID_{new}^{i}. Comparing to the $d_{SID}^{threshold}$, any d_{SID}within threshold achieves similarity of 1, otherwise the similarity is 0.

$$d_{SID} = \sqrt[2]{(SID_{set} - SID_{new}^{i})^2} \quad (7)$$

The d_{SID} threshold need to be defined according to the structure of the sensor ID. For example, if acoustic sensor numbered from 1001 to 1010, temperature sensor numbered from 2001 to 2020, accelerometer numbered from 3001 to 3030, then the $d_{SID}^{threshold}$ should be 30. Alerts coming from sensor ID difference within threshold 30, such as 3007 and 3020, SID_{sim} should be 1; Alerts coming from sensor ID difference beyond threshold, such as 1002 and 3002, SID_{sim} should be 0.

$$d_{SID}^{threshold} = \max_{k=1..x}(SID_{max}^{k} - SID_{min}^{k}) \quad (8)$$

The equipment ID similarity EID_{sim} should be 1 if two alerts have the same EID, otherwise the similarity will be 0.

$$\begin{cases} EID_{set} = EID_{new}^{i} \ , \ EID_{sim} = 1 \\ \ otherwise \quad , \ EID_{sim} = 0 \end{cases} \quad (9)$$

5.2.2 Manufacturing Process Similarity

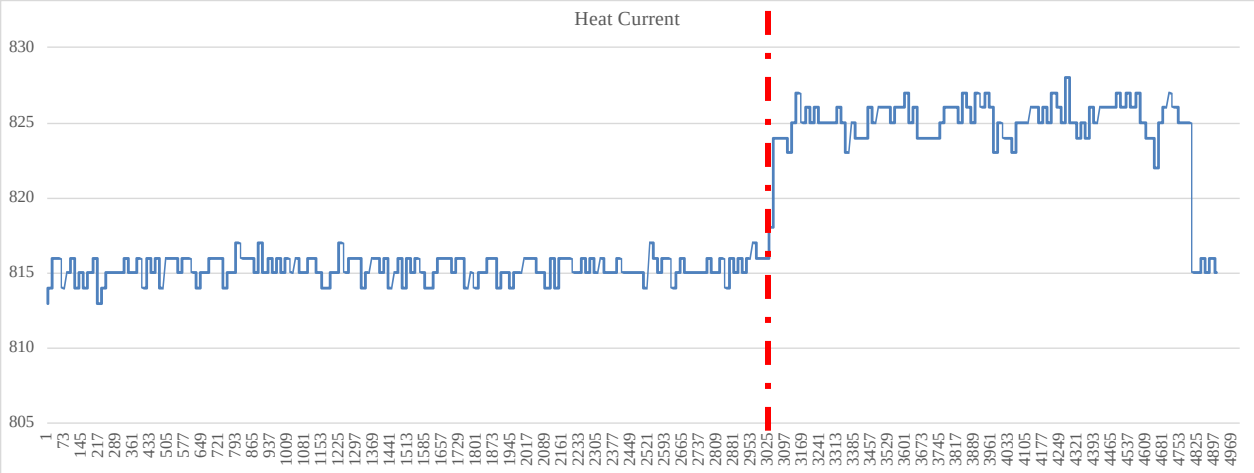
The manufacturing type similarity aims to correlate the alerts from same manufacturing processes. It indicates that a manufacturing process is compromised by a type-specific attack. For example, when the 3D printing data hose is injected with malicious infill designs, the different local supplier can have similar alert from 3D printers via camera real-time image classification.

The manufacturing process similarity MP_{sim} is 1 when two alerts have same MPID, otherwise the MP_{sim} is 0.

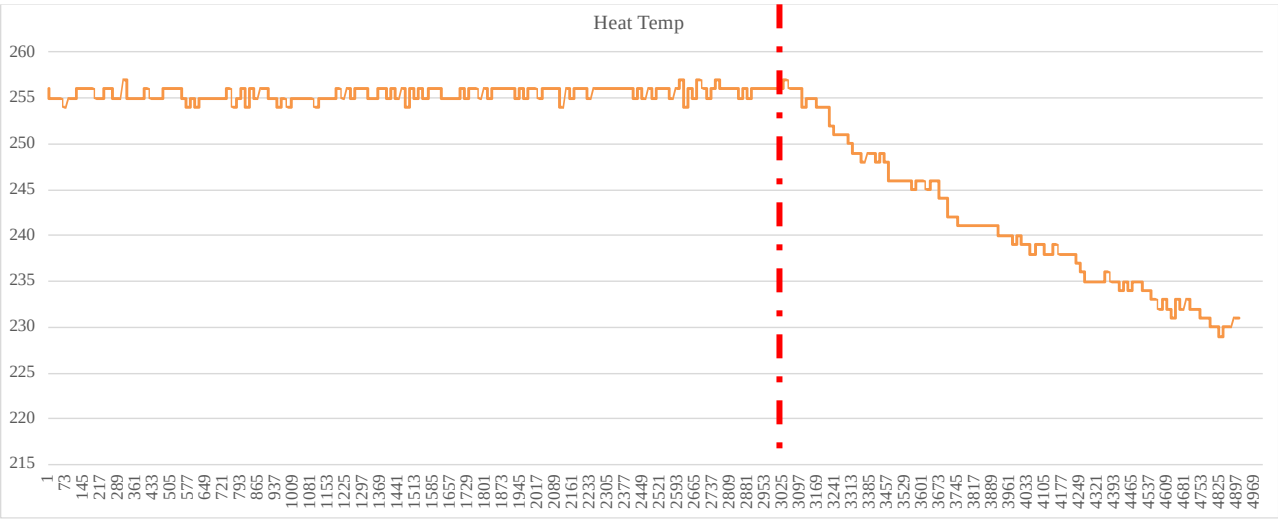
$$\begin{cases} MPID_{set} = MPID_{new}^{i} \ , \ MP_{sim} = 1 \\ \ otherwise \quad , \ MP_{sim} = 0 \end{cases} \quad (10)$$

5.2.3 Time Similarity

The time similarity aims to correlate the physical alerts from different sensors on same machine, trigger by same attack in a short time period. For example, in Figure 24, a CNC milling feed speed and spindle speed attack, the acoustic sensor and accelerometer shows alerts within 6 seconds.



(a) Heat treatment process current plot



(b) *Heat treatment temperature plot*

Figure 24 Physical alert time similarity comparison

Similar to section 3.1.3, the time similarity correlation needs to define the time window, or T_{window}. Different from the cyber alert, the physical alert can refer to the manufacturing job length time T_{job}, and the job end time T_{end}, as a result,

$$T_{term}^{c} = T_{end} \quad (11)$$

Moreover, the window time is dynamically allocated according to the job length,

$$T_{window} = T_{job} \quad (12)$$

In another word, physical alerts generated within the same manufacturing job on a physical machine should be correlated.

5.2.4 User Identification (UID)

The user identification aims to correlate the physical alerts caused by the same user. It can hardly prevent any sophisticated attacks that hiding the alerts to user accounts but can detect the alerts that caused by repackaging attack or other attacks that victim user unaware of.

The method of calculating attribute-based user ID similarity can be calculated via the Euclidean distance function, as follows:

$$\begin{cases} UID_{set} = UID_{new}^{i}, \ UID_{sim} = 1 \\ otherwise \quad , \ UID_{sim} = 0 \end{cases} \quad (10)$$

Where the threshold can be set to near zero to find an exact match of user ID.

5.3 Cyber-physical alert correlation

Different from the cyber or physical alert correlation methods, which are a combination of temporal and attribute-based analysis, the cyber-physical alert correlation only based on the attribute-based analysis. It is because the weak correlation between the cyber alert create time and physical alert create time: an intrusion can happen on cyber domain days or even weeks before and physical consequence happen. As a result, following attributes are defined for cyber-physical meta-alert correlation.

5.3.1 Destination IP with the Manufacturing Process

As defined in section 3.1.2 and 3.2.2, the CMS host network is segmented based on the manufacturing process. The destination IP of each host is designated to customers on different manufacturing service. For example, the Table 14 is a destination IP and manufacturing process similarity matrix, which shows the only correlation between IPs and manufacturing processes.

Table 14 Host IP and Manufacturing Process correlation matrix

		Manufacturing Process						
	Similarity	Additive_Manufacturing_PLA	Additive_Manufacturing_ABS	Subtractive_Manufacturing_Wood	Subtractive_Manufacturing_PLA	Subtractive_Manufacturing_ABS	Subtractive_Manufacturing_Aluminum	Heat_Treatment
Host IP Address	113.238.46.13	1	0	0	0	0	0	0
	39.3.197.234	0	1	0	0	0	0	0
	36.133.70.114	0	0	1	0	0	0	0
	93.211.37.77	0	0	0	1	0	0	0
	184.34.21.200	0	0	0	0	1	0	0
	100.109.244.1	0	0	0	0	0	1	0
	101.60.193.233	0	0	0	0	0	0	1

5.3.2 Source IP with User ID

The source IP address, or *SIP* from cyber-alerts provides the IP address of the customer or attacker who triggers the alarm in the cyber domain. The user identity, or *UID* from physical

domain alarm provides the same customer information during production. By correlating SIP and UID, the physical alarm can effectively trace back to the source of the root cause.

Different from the DIP, each user could have more than one IP addresses. It could be caused by multiple user login, log in from different locations or devices, dynamically allocated IP addresses, or even log in from a malicious user. As shown in Table 15, a user with UID_56474358546 has multiple IP addresses can be correlated to one user ID. Any alerts in the physical domain caused by this UID should be correlated to any alerts caused by those five highlighted IP addresses

Table 15 UIP and UID correlation matrix

		User ID					
	Similarity	UID_789548786	UID_678543786	UID_564743585	UID_43453654	UID_45332564	UID_12324456
User IP address	50.126.114.76	1	0	0	0	0	0
	101.39.143.223	0	1	0	0	0	0
	50.35.81.191	0	0	1	0	0	0
	165.129.82.29	0	0	1	0	0	0
	74.170.160.209	0	0	1	0	0	0
	210.98.94.55	0	0	1	0	0	0
	64.99.107.206	0	0	1	0	0	0
	208.81.201.180	0	0	0	1	0	0
	159.149.228.81	0	0	0	1	0	0
	113.200.27.26	0	0	0	0	1	0
	197.115.19.110	0	0	0	0	0	1
	188.218.195.29	0	0	0	0	0	1

Chapter 6

Experiment Design and Case Studies

In this chapter, four case studies were created, presented and analyzed for the proof of concept and validation of the cyber-physical intrusion detection and correlation methodology. Multiple cyber-attack vectors/methods, along with various physical attack payload/consequence are analyzed and integrated for the following four case studies:

- A weakened 3D printing object
- A manipulated CNC milling process
- A multiple robotic arm speed attack
- A supply chain attack

A cyber-physical intrusion detection oriented Cyber-Manufacturing System Security Testbed (CSST) is established for experiment for various reasons. One of the most important is that the cyber-physical attack is new since the emergence of Stuxnet; at the same time, CMS or other manufacturing visions, such as Industry 4.0 or Cloud Manufacturing, are not established.

6.1 Test Environment

The Cyber-Manufacturing System Security Testbed (CSST) is the environment to test and validate the intrusion detection and alert correlation case studies. It is a testbed developed for the needs of intrusion detection and prevention research, the development requirement including:

i. Simulate the CMS physical process with a simple minimum setup to reduce the set-up cost and attack damage cost.

ii. Simulate the CMS network environment with the most common and basic network setup, with potential to expand or replace with more advanced technologies.

iii. Collect cyber and physical data for intrusion detection analysis, with the potential to collect more types of data with additional sensors.

iv. Simulate cyber-physical attacks within a manufacturing system, and along a simple supply chain.

The environment is developed with the reference to CMS hierarchical five-layer architecture (Z. Song and Moon 2016c). To mitigate the risk that experiment cyber-physical attacks cloud damaging expensive equipment, a simple minimum layout is set up: use one machine to represent each type of manufacturing process; when choosing a machine, the ability and flexibility to collect data are the first priority when the fabrication size, speed, precision are the second priority.

Supply chain as a core part CMS is loosely managed in most cases, making penetrations through supply chain attack possible. As a result, a simple supply chain is also created in the test bed for research on the cyber-physical attacks through the supply chain.

6.1.1 System Architecture and Design

The CMS testbed consists of six major components: (i) discrete event computer simulation, (ii) cyber environment for customer web service (iii) physical manufacturing process and equipment, (iv) control system, (v) network communication system, and (vi) monitoring system.

As shown in Figure 25, the computer simulation can provide randomized customer and attacker arrival schedule. The researcher can play the role of customer or attacker based on randomized job schedule, place an order or penetrate into the customer database. The order can be fabricated within only one testbed or between testbed supplier and demander. The physical flow in Figure 25 shows a part first fabricated in testbed supplier then transferred to testbed demander and assembled with another part fabricated by demander testbed.

The physical testbed audit system collects cyber data such as network activities and host log, along with physical data such as image, acoustic, acceleration, temperature, power consumption, part dimensions to the isolated database for intrusion detection analysis.

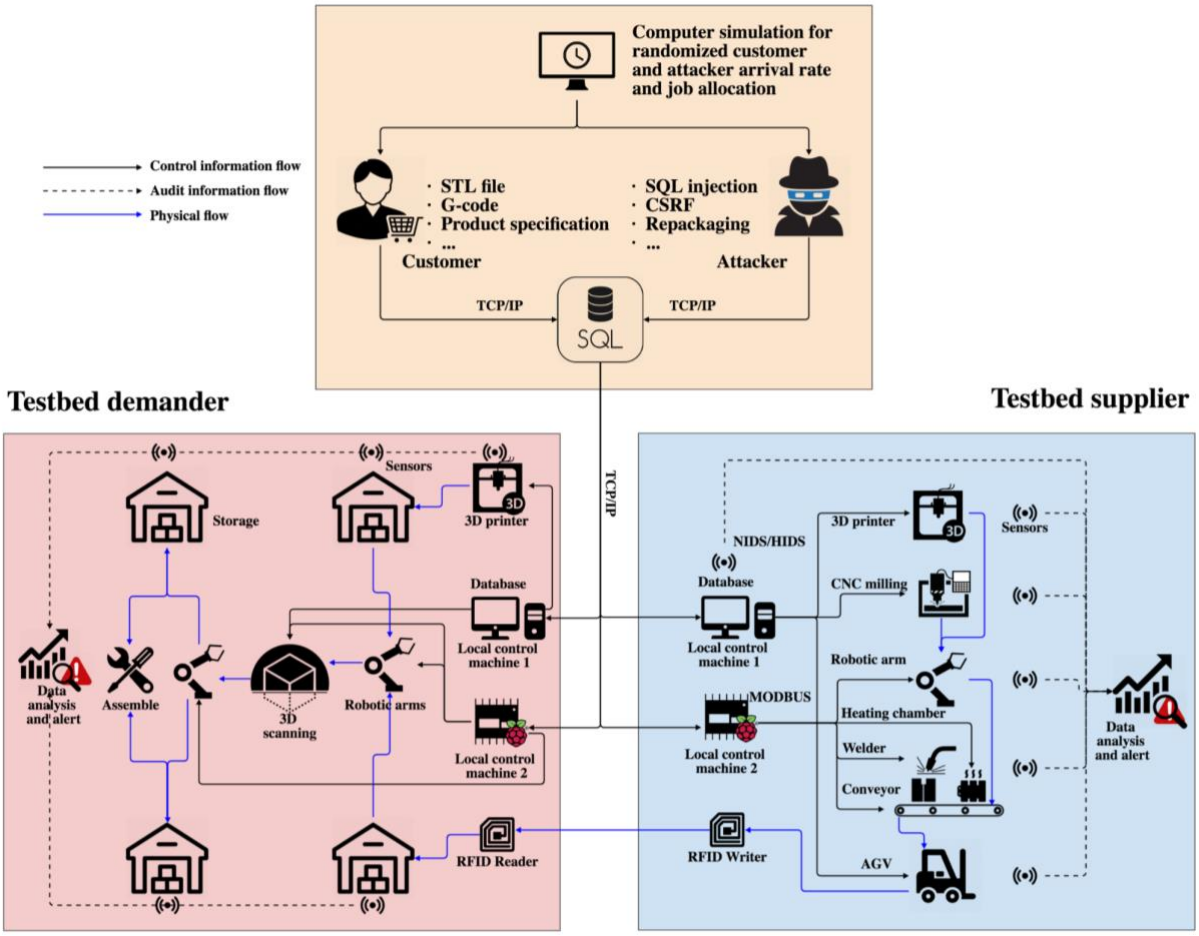
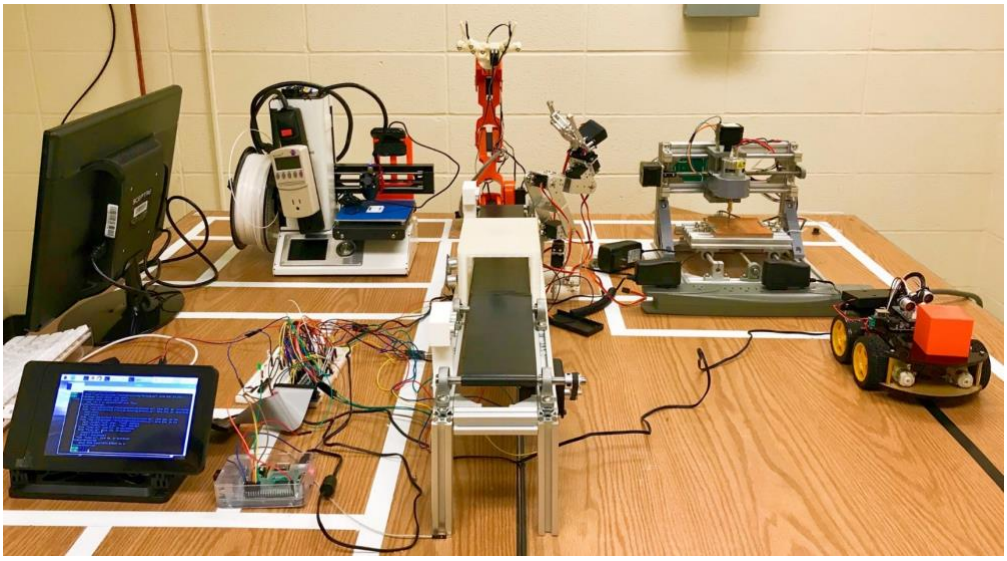
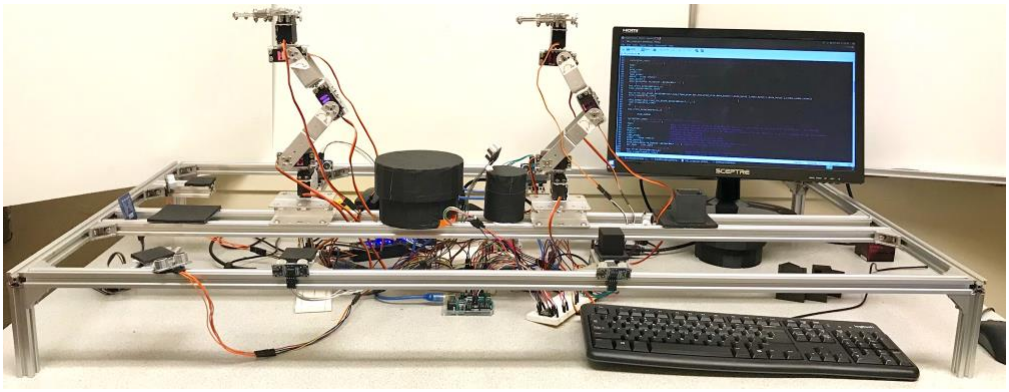


Figure 25 CMS IDS testbed diagram

In general, these five components have following functionalities: (i) mimic CMS production flow; (ii) generate and collect cyber and physical data for analysis; (iii) implement and validate cyber-physical intrusion detection and correlation method such *as Define, Audit, Correlate, Disclose, and Improve - DACDI* (Wu and Moon 2017a), and develop countermeasures for preventing, mitigating cyber-physical attacks. As shown in Figure 26, the testbed is setup in a separate environment to simulate CMS production flow.



(a) Testbed supplier setup

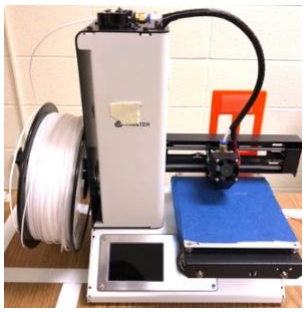


(b)Testbed demander setup

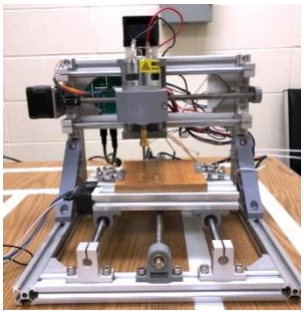
Figure 26 Testbed setup

6.1.2 Physical Manufacturing Processes

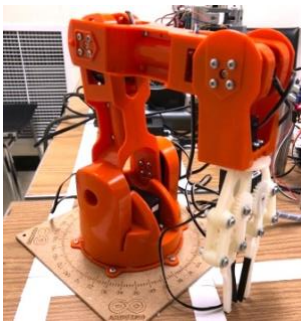
The physical system consists of two 3D printers, a CNC milling machine, four robotic arms, a conveyor, and heating chamber, an Automated Guided Vehicle (AGV), 3D scanner and RFID reader/writer as shown in Figure 27. The equipment (a-g) is set up as a supplier testbed. The equipment (h-j) is set up as a demander testbed.



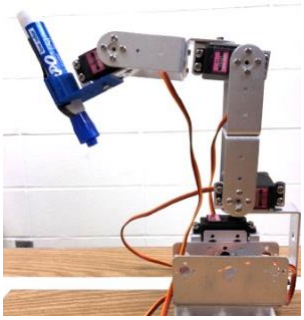
(a) 3D Printer



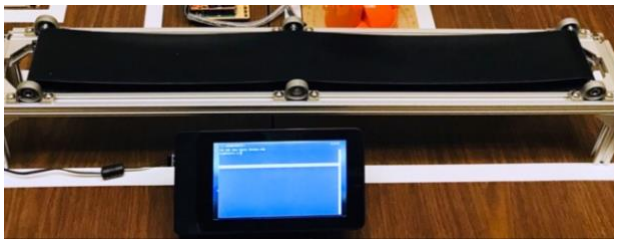
(b) CNC Milling Machine



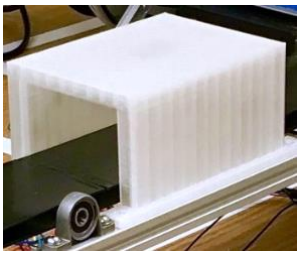
(c) Robotic Arm for Moving



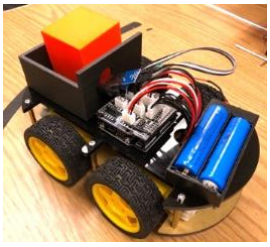
(d) Robotic Arm for Welding



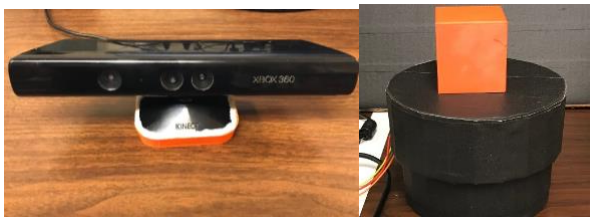
(e) Conveyor



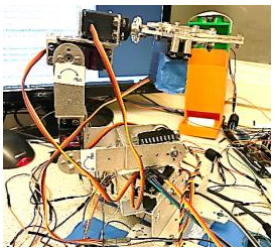
(f) Heating Chamber



(g) Automated Guided Vehicle



(h) 3D Scanner and Turn Table



(i) Robotic Arm for Assemble

(j)RFID Reader/Writer

Figure 27 Testbed physical environment

6.1.2.1 Additive Manufacturing

As shown in Figure 27 (a), the testbed integrates an MP Select Mini 3D printer V2 to represent the additive manufacturing process. The machine is capable of constructing design under the dimensions of 120 x 120 x 120 mm with ABS or PLA material. The machine can print “STL” files via a connected Windows 10 desktop machine with Cura 3D printing software.

6.1.2.2 CNC milling machine

As shown in Figure 27 (b), the testbed integrates a three-axis CNC milling machine to represent the subtractive manufacturing process. The machine can read “Gcode” file from same desktop machine with Grbl Controller 3.0 software.

6.1.2.3 Robotic arm for carrying/moving

As shown in Figure 27 (c), the testbed integrates an Arduino Braccio six degree of freedom robotic arm for carrying sample objects from storage area to conveyor. The robotic arm executes pre-defined code in Arduino UNO R3 micro-controller. The Arduino UNO is connected to a Raspberry Pi as a control machine.

6.1.2.4 Conveyor

As shown in Figure 27 (e), the testbed integrates a custom built conveyor for conveying sample object through welding process and heat treatment process. The conveyor is powered by a step motor controlled by the same Raspberry Pi.

6.1.2.5 Robotic arm for welding

As shown in Figure 27 (d), the testbed integrates a custom built robotic arm for simulating the welding process. The robotic arm has six degrees of freedom with the hand attached to a marking pen. The pen simulates the welding pattern instead of the real welder. Similarly, the welding robotic arm executes pre-defined code in Arduino UNO R3 micro-controller connected to Raspberry Pi as the control machine.

6.1.2.6 Heat treatment

As shown in Figure 27 (f), the testbed integrates a heating chamber for simulating the heat treatment process. The heating chamber consists of a tunnel that encloses the heating environment,

an ultrasonic sensor detects sample object arrival, and 3 heating elements controlled by Raspberry Pi.

6.1.2.7 Transporter AGV

As shown in Figure 27 (g), the testbed integrates a AGV as a final transporting device. It carries sample object coming out of conveyor to the packaging area. The AGV is self-controlled by an Arduino UNO connected wired/wirelessly to local control machine.

6.1.2.8 3D Scanner and turntable

As shown in Figure 27 (h), The 3D scanner and turntable setup is a function that attempts for checking the part dimensions. The part is placed on turntable and rotate 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 original design file.

6.1.2.9 Robotic arm for assemble

As shown in Figure 27 (i), the robotic arm for assembling shares the same model type as the robotic arm for welding, expect in total six servo motors are integrated into the arm for multiple axis movements. The robotic arm can only accomplish simple assemble job but enough for demonstrative purpose.

6.1.2.10 RFID reader/writer

As shown in Figure 27 (j), the RDIF read/write function utilizes Mifare RC522 sensor module and NTAG 215 NFC sticker. The NFC sticker is thin and compact in size for attaching on part with 540 bytes of memory. Information such as part ID, customer ID can be stored and attached on parts.

6.1.3 Control System

There are two local computers each testbed: a Windows 10 based desktop machine connected to Ethernet, and 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 connections are wired but with wireless potential.

6.1.3.1 Control machine for CNC and 3D printer

The Windows 10 based desktop machine 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 Grbl Controller software sending "Gcode" to the machine. In Grbl Controller, milling setting such as feed speed, spindle speed can be modified.

6.1.3.2 OpenPLC for conveyor

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

6.1.4 Communication

This testbed utilizes an Ethernet-based communication control system. The local control machines such as the Windows 10 based desktop machine or Linux based raspberry pi are connected to the Internet for the purpose of connectivity to the CMS database via TCP/IP

communication protocol. Within the testbed, Modbus is used between Raspberry Pi and its connected actuators.

The reason of adopting TCP/IP and MODBUS communication protocol in the testbed is: (i) the popularity of them in nowadays manufacturing systems; (ii) the abundant resource of available network monitoring system and (iii) the communication protocol standardization of future manufacturing visions are not yet accomplished (Bitkom, Vdma, and Zvei 2016). Regardless of which type of protocol is chosen here, the proposed methodology of intrusion detection on cyber-physical attacks function the same.

6.1.4.1 TCP/IP

The connection between customer and website frontend, database, and local control machine are 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 today (Boyle and Panko 2013).

6.1.4.2 MODBUS

MODBUS is a free and open source protocol that developed by Modicon for PLCs. It is popular among the industrial control. As the OpenPLC with raspberry Pi 3 has the capability of Ethernet over TCP/IP, it was implemented support for the MODBUS TCP/IP protocol (Alves et al. 2014). The Modbus TCP/IP is the Modbus protocol with a TCP interface that runs on Ethernet (Goldenberg and Wool 2013).

6.1.4.3 I2C

The connection between the Raspberry Pi and the robotic arm controller Arduino UNOs are over I2C 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).

6.1.5 Monitoring System

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 needed to be collected. Between those two, the physical data analysis is the novel part for intrusion detection of cyber-physical attacks.

Cyber data is capable of: (1) detecting amateur and known attacks, and (2) use as evidence to correlate with physical anomaly occurrence. Physical data is capable of detecting cyber-physical data quickly with high accuracy (Wu, Song, and Moon 2019; Wu et al. 2017; Song et al. 2017), and can also prevent machine malfunction and human mistakes as a byproduct.

6.1.5.1 Cyber data auditing

Cyber audit data includes the data from network activity and host. Snort network-based intrusion detection system (NIDS) software is used to tap network activity log data, such as login attempts, network connections, or every data packet that appeared on the wire (Kemmerer and Vigna 2002). As a packet sniffer, it can monitor network traffic in real time on local control machines and database host of the testbed. The standard rules are used to checking abnormal data in packet traffic (Khamphakdee, Benjamas, and Saiyod 2014).

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

6.1.5.2 Physical data auditing

The physical data is collected from the manufacturing processes and equipment on the testbed. At least two types of sensor are used on each machine/process for the security and alert accuracy, as shown in Table 16.

Table 16 Physical data auditing list

Equipment	Sensor #1	Sensor #2
3D Printer	**Power Meter**	**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**

In following paragraphs, sample data collected form tested bed with analysis method such as feature extraction for machine learning are presented.

6.1.5.2.1 Power consumption data from power meter

The power consumption data is recorded by a Kill-A-Watt P4400 power meter. The malicious and legitimate data is created by print a malicious infill defect (Wu et al. 2016) that can weaken the part structurally (Sturm et al. 2017b).

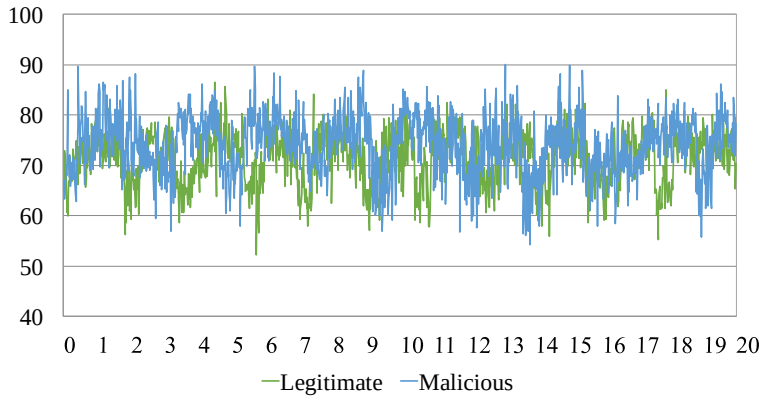


Figure 28 Power consumption data analysis

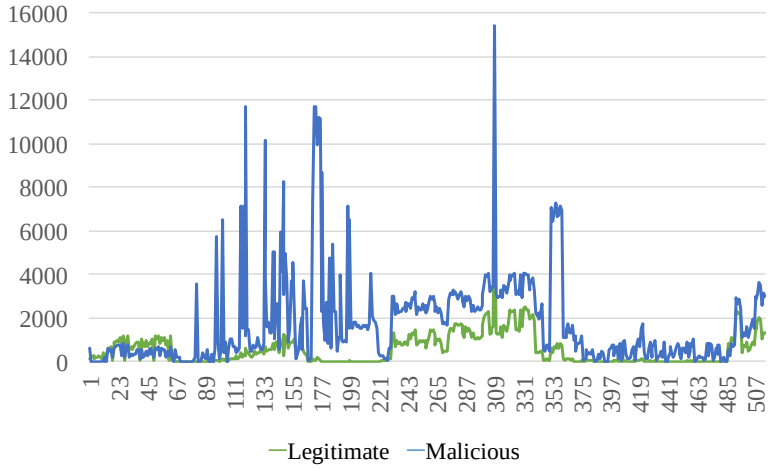
To analysis the power consumption data of 3D printing process as shown in Figure 28, the window time of feature extraction is set for every 100 seconds. During the window time period, the mean value, standard deviation, maximum, medium, minimum, skewness, kurtosis, number of power data points over 80, 82 and 85 kWh are calculated as features. In total, there are ten features are used. In Figure 28, the unit for y-axis is kWh, and the unit for x-axis is minute.

6.1.5.2.2 Image data from the camera

The image for 3D printing process sample part infill and welding process quality is taken by two similar Logitech C310 and C525 cameras. The greyscale value is used for data analysis.

(a) Welding mark

(b)Infill detect



(c) Number of pixels over the grayscale threshold

Figure 29 Image data analysis

The images collected from the welding mark and 3D printing infill are shown in Figure 29 (a) and (b). Each image as a dataset can be divided vertically into eight equal areas. In each area, the greyscale mean value, standard deviation, and the number of pixels over the grayscale threshold are obtained. As shown in Figure 29 (c), the number of pixels over the grayscale threshold between malicious and legitimate images defecate immensely.

6.1.5.2.3 Acceleration data

The accelerometer installed on CNC milling machine and AGV for monitoring the dynamic activity. They are MMA7361 accelerometer sensor with a sampling rate of 115200 bauds,

controlled by Arduino UNO microcontroller. Figure 30 shows the acceleration data at the first 120 seconds with legitimate and malicious settings via manipulating the feed speed.

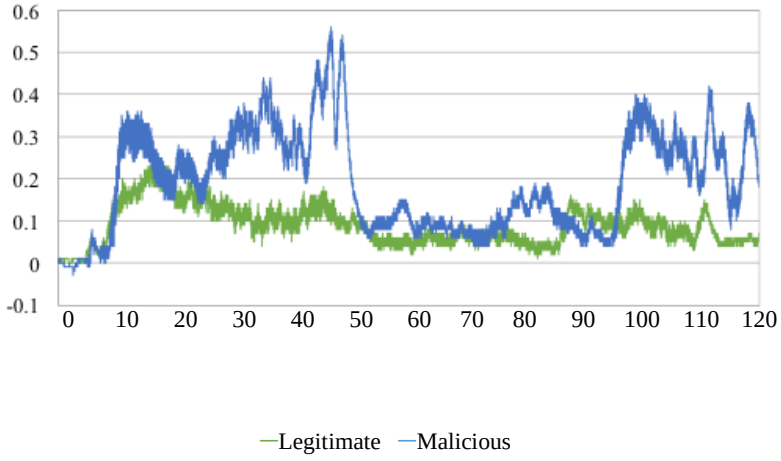


Figure 30 Acceleration data analysis

For acceleration data analysis, the window time for feature extraction is set to 1.5 seconds when the sensor collects around 108 acoustic signals per second. During every 1.5 seconds, the acceleration mean, standard deviation, 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 30, the unit for y-axis is g (G-forces), the unit for x-axis is seconds.

6.1.5.2.4 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 9600 bauds, controlled by Arduino UNO microcontroller. Figure 31 shows the first 2 minutes of CNC milling process with legitimate and malicious settings via manipulating the spindle speed.

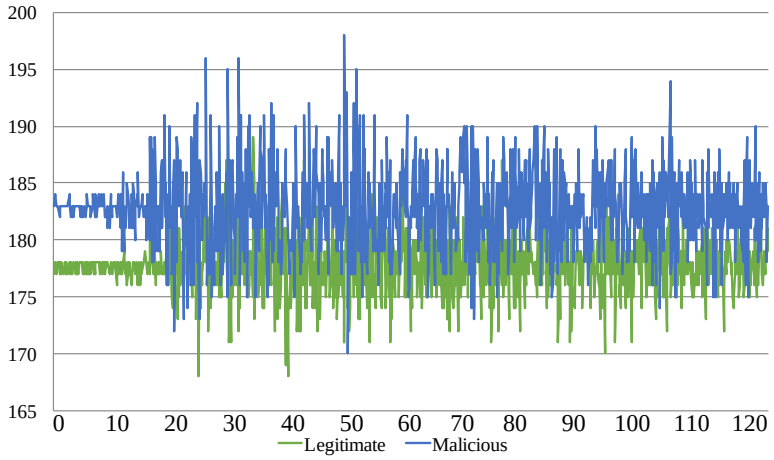


Figure 31 Acoustic data analysis

To analyze the acoustic data, the window time for feature extraction is set to 1 second when the sensor collects 20 acoustic signals per second. During 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 31, the unit for y-axis is dB, the unit for x-axis is seconds.

6.1.5.2.5 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 sensor controlled by Arduino UNO microcontroller. Figure 32 shows the current data from the conveyor.

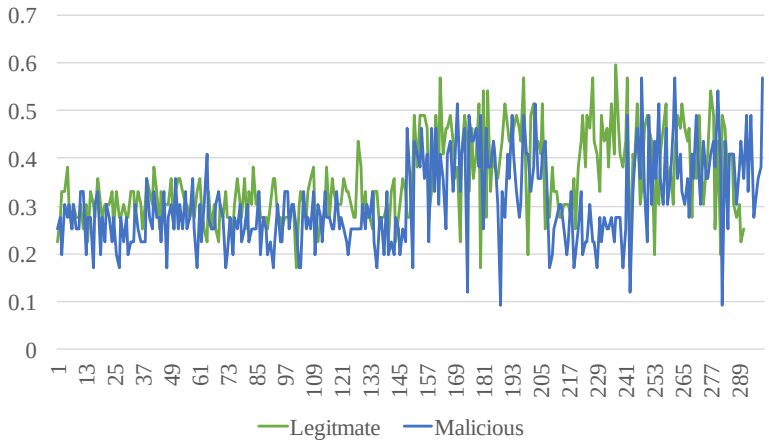


Figure 32 Current sensor analysis

To analyze the current data, the window time for different machine or process varies. During each window time, the current mean, standard deviation, maximum, medium, minimum, number of acoustic data amplitude over the threshold are calculated as features for detection. In Figure 32, the unit for y-axis is Amp, the unit for x-axis is seconds.

6.1.5.2.6 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 a rate of 1 Hz. Figure 33 shows the temperature data from the heating treatment process.

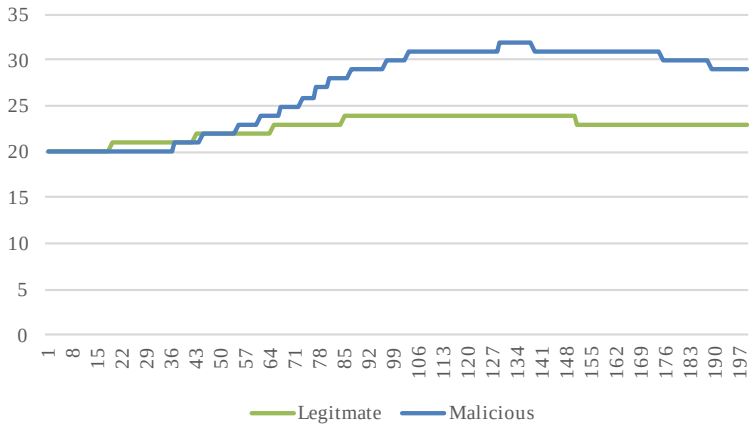


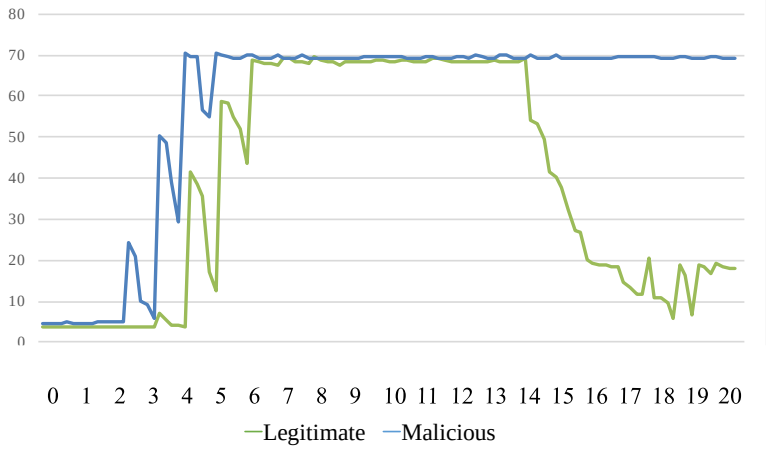
Figure 33 Temperature data analysis

To attack on heating treatment process, the intrude manipulate the heating element power voltage from 5 volts to 9 volts. The potential physical consequence is the overload of the power system and also change the physical character of treated part. As shown in Figure 33, the malicious heating process has a greater increasing trend. In Figure 33, the unit for y-axis is Celsius, the unit for x-axis is seconds.

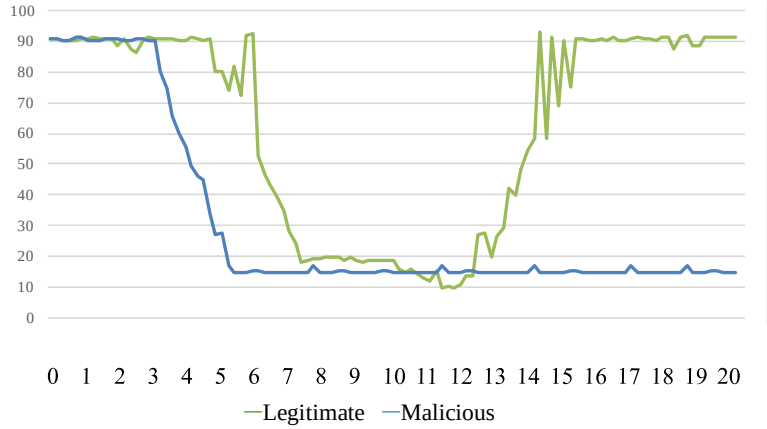
To analyze the temperature data, the window time is set for 7 seconds. During each window time, the temperature means, standard deviation, maximum, medium, minimum, number of acoustic data amplitude over 25 ℃, 28 ℃ and 30 ℃ are calculated as features for detection.

6.1.5.2.7 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 a rate of 5 Hz. Figure 34 shows the ultrasonic data from the AGV route area.



(a) Ultrasonic sensor at loading zone



(b) Ultrasonic sensor at unloading zone

Figure 34 Ultrasonic sensor analysis

To attach the AGV, the intruder changes the control code of the AGV, making it leave assigned route. As shown in Figure 34, both sensors read maximum distance value when AGV leave route. The potential consequence of this attack can be damage to vehicle, production environment or even human safety. In Figure 34, the unit for y-axis is mm, the unit for x-axis is seconds.

To analyze the ultrasonic data, the data from two ultrasonic sensors are added up and set the window time for 1 second. During every window time, the ultrasonic mean, standard deviation,

maximum, medium, minimum, number of acoustic data amplitude over the threshold are calculated for detection.

6.1.5.2.8 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 35 shows the avoidance sensor data.

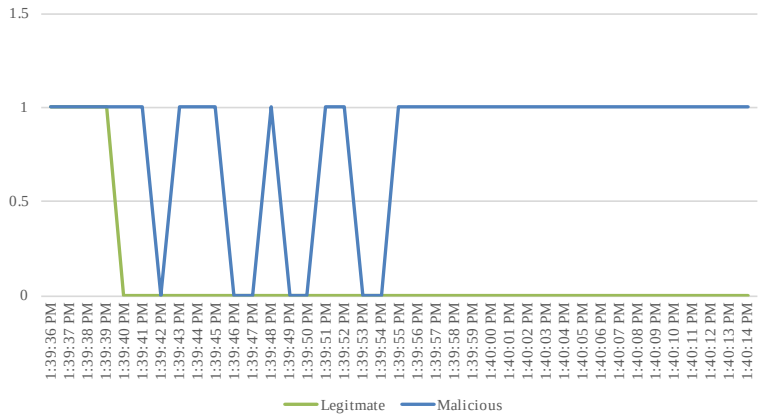


Figure 35 Avoidance sensor data analysis

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 not expecting robotic arm to intrude. It could be areas with equipment or human. When the robotic arm under attack, its working logic is changed, and the potential consequence could be damage to equipment or even human safety. For avoidance sensor data, anytime the sensor gives back value of 1 can be classified as alert.

6.1.5.2.9 3D scanner data

The 3D scanner is located before the assemble process in demander testbed. It is an Xbox 360 Microsoft Kinect Sensor controlled by Skanect 3D Scanning Software. The software will

capture object and export to a “STL” file. The “STL” file will be post-processed in Cloud compare software to check the dimension difference to the original design.

6.2 Cyber-Physical Attack Scenario Design

The attack scenarios are designed to simulating the cyber-physical attacks in the CMS environment. The cyber-physical attack is defined as “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).

As analyzed in Chapter 3, the existing confirmed and published cyber-physical attacks are limited: Stuxnet (Langner 2011) and German steel attack (R. M. Lee, Assante, and Conway 2014). The reasons behind is phenomena including: the unawareness of cyber-physical attack, the trend of under reporting (IBM-Security 2017), and business reputation, confidentiality, etc.

As a result, the cyber-physical attacks are decomposed into two components based on its definition: cyber-attack method and physical consequence.

6.2.1 Cyber-Attack Method

The cyber-attack methods are various and growing every day. In general, they can be categorized into two types: known attack and unknown attack. This work select SQL injection (SQLi) as an example of sophisticated known attack, and 3D printing repackaging attack as an example of unknown attack.

6.2.1.1 SQL Injection (SQLi)

According to a 2016 security report from IBM, 74 percent of their manufacturing clients is 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 percent. Among those code injection attacks in manufacturing, SQL injection made up 45 percent of these attacks ranks the most frequent cyber-attack vectors among all code injection attacks (IBM-Security 2017).

With 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 CMS testbed, the intruder can spoof into the MySQL 5.7 customer database without known the customer's password when the "magic quote" countermeasure turned off.

For example, the CMS customer with username "UID001" and password "1234" can login 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 will have full access to download, edit, upload, and remove the customers' order.

One example of cyber-physical attack via SQL injection is change CAD/CAM file or manufacturing specification. A hacker can access into 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.

6.2.1.2 Repackaging

The repackaging attack originated from smartphone applications. An attacker can download an online banking application (Jung et al. 2013), decompile the application, add malicious functions and upload back to 3rd party application store to obtain any user's information.

In a CMS, a designer can upload their finished design into online marketplace for customers to choose from. The customer pays to get the design file. The file can be an "STL" file for 3D printing, G-code for CNC machining, or any other types of CAD file. The customer can either upload the file to CMS database directly or revise it further and generate a self-designed file. When customers select their designs and products from a 3rd party store, the repackaging attack can happen 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 online marketplace. The repackaging attack occurs in the CMS's customer layer.

The customers can be easily fooled to purchase and download the design from the online market because it is difficult for them to notice the difference between the modified design and original design. Once the modified design is uploaded to CMS database, it will go through a typical process—certification check, model check, order confirmation, and distribute to the specific physical provider for manufacturing. However, results are defective parts, machine malfunction, etc.

6.2.2 Physical Payload

In this section, four types of physical payload are introduced and will be integrated in later case studies.

6.2.2.1 3D printing quality manipulation

The potential attack payload of a repackaging attack can be design alteration, such as embedding malicious infill void defect via change the "STL" file as shown in Figure 36. It is difficult to observe visually by customers or inspectors. Moreover, Sturm (Sturm et al. 2014)

proved that the structural stiffness of a 3D printing test piece with infill void could reduce by at least 14%. As shown in 10, the malicious seam-shape infill defect can be embedded in the connector part, and cause early breakage in use.

To simulate this attack in CMS testbed, a malicious "STL" file can be sent to the database. 3D printer will proceed to manufacture the part with malicious void while sensors collect physical data in the process.

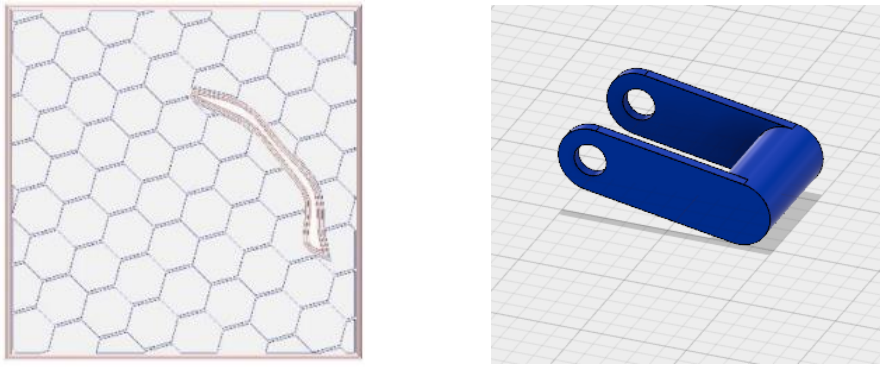


Figure 36 Repackaging on "STL" file with malicious infill void

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 monitored during the production process; and structural health data monitored in post-production stage. Using the pre-defined architecture and correlation, the physical alert will be traced all the way back to the customer's design file. An alert will be sent to the customer, and the corresponding design and designer will be added to the blacklist.

Other 3D printing settings, such as 3D printer's heating bed and nozzle temperature, can also be attacked on the local control machine. It can cause problems such as low quality, gaps between infill and an outer wall, the high defective rate at inspection.

6.2.2.2 CNC milling equipment manipulation

To conduct a cyber-physical attack on CNC milling, an attacker can download the original design from the customer, modify it with malicious parameter or structure, then upload to the data server. For example, the spindle speed and feed speed change in a G-code file can be modified by the attacker. The user could only focus on the appearance of the design and satisfied with the malicious file. When a corrupted file is sent to the physical provider and being manufactured, the change of the spindle speed and feed speed can cause the breakage of drilling bit.

6.2.2.3 Robotic arm attack

To conduct a cyber-physical attack on robotic arm, an attack can alter the user perceived robot state to cause operator injuries (Quarta et al. 2017). The attacker can manipulate the status information, so the operator is not aware of the true status of the robot.

The operator interface must provide timely information at least on the motor state (on/off) and operational mode. Moreover, standards mandate that safety-critical conditions require deliberate user confirmation. Unfortunately, some of these conditions are communicated and require user interaction via software, not through electrical components.

To apply a robot state interface attack safely without cause any damage, two operational modes were defined: normal mode and maintenance mode. When an intruder manipulates the robotic arm control system, the robotic arm will operate under maintenance mode, which will change the operating speed.

6.2.2.4 Supply chain attack

In CMS environment, geographically distributed manufacturing equipment are controlled by the global business center. The global business center will make job allocations depending on the customer order priority and physical provider availability.

The attacker will gain super user privilege via race condition and change the customer orders and physical provider data. A wrong part could be sent to the assembly manufacture. This attack simulates the supply chain attacks and tests the aftermath countermeasure to mitigate such an attack.

6.3 Experimental Design

In this section, the repeatability of the relation between cyber alerts and the cyber-attack vector is discussed. Second, the duties for different roles in the experiment are designed. The roles will be played by engineering students independent from the intrusion detection student team. The randomness and credibility of the experiment is guaranteed by this practice.

6.3.1 Factorial Design

The network environment is based on an Ubuntu 14.04 operating system, with the magic quote function turned off so as to be vulnerable to SQL injection attack. A web application-based customer login front end is hosted and connected to user data based on MySQL 5.7 and Apache HTTP Server 2.4.

To simulate a CMS network environment, a two-level three-factor experiment were designed to test the NIDS software Snort and HIDS software OSSEC. The attack scenarios use SQL injection to get into the CNC machining database, change the feed speed and spindle speed of a G-code, and trigger alerts from both the acoustic sensor and accelerometer.

Snort is an open source, lightweight, cross-platform software, originally developed by Martin Roesch in C language in 1998. It uses predefined rules for checking abnormal data in packet traffic (Roesch 1999a). In our experiment, Snort is equipped with standard rule along with additional SQL injection rules, as follows.

Snort SQL Injection Local Rules

alert tcp any any -> any 80 (msg: “Error Based SQL Injection”; content: “%27” ; sid:100000011;)

alert tcp any any -> any 80 (msg: “Error Based SQL Injection”; content: “22” ; sid:100000012;)

OSSEC is an open source, multi-platform, scalable host-based intrusion detection system (HIDS). It analyzes the host log, file, windows registry, and provides real-time alert responses. The OSSEC is equipped with standard rules.

The three factors in the experiment are: normal customer activity, SQL injection attack, and noise by NMAP software (Orebaugh and Pinkard 2011).

6.3.1.1 Factor one: Cyber-physical attacks

Let students simulate hacker use commands such as “UID_xxxx'; -- ” or “' or 1=1; --” to directly access a customer account or administrator account without knowing the password. Such an act will trigger alerts from Snort software.

6.3.1.2 Factor 2: Network scan noise

Let students use Nmap, a free open source network scanning utility (Orebaugh and Pinkard 2011), to intensely scan a customer website and database host to create false alarms.

6.3.1.3 Factor 3: Customer activity

Let students simulate customers use computer visit customer front-end website, and create events such as login, uploading orders, deleting orders, editing orders and log out.

6.3.2 Role and Duty Design

As a result, there will be at least two roles in the experiment for students to act as. The attack guideline will be provided to student to follow, but the key factors, such as attack time, event order, the attack payload will be decided by the student.

6.3.2.1 Hacker

The hacker will be responsible for attacking the CMS test bed and also adding interference for the intrusion detection system. The hacker needs to be trained with basic knowledge or cyber-attack knowledge as well as manufacturing knowledge, such as CAD/CAM software skills.

Table 17 Attack guideline example

Actions	Order	Begin Time	End Time
User: MTW	1	9:00	9:03
User: BYD	3	9:09	9:15
User: YPL	5	9:22	9:24
Attack: SQLi	4	9:21	9:23
Interference: Scan	2	9:05	9:10

6.3.2.1.1 Duty

The duty of a hacker includes the use of SQLi attack vector to intrude into the CMS database, and downloading and editing the customer's CAD/CAM file, such as "STL" file or "Gcode." The editing of the CAD/CAM file should contain malicious influence to the part or

manufacturing process. The malicious influence should be reasonable and logical so that the machine, such as a 3D printer, can still fabricate the part.

The hacker also needs to use another machine to scan the CMS hosts to give the interference to the intrusion detection system. If there are multiple hosts, multiple scans are preferred.

6.3.2.1.2 Randomness

There are many factors that need to be decided by the role player himself/herself; for example, the timing of attacking, the physical payload of the attack, the times of attack, the design of the attack, etc.

6.3.2.2 Customer

The customer role is comparatively easy. The customer can be trained with the introduction of a/the CMS customer interface. The customer is used to create normal traffic for the web frontend and to create a/the target for cyber-physical attack.

6.3.2.2.1 Duty

The duty for the customer is to select the CAD/CAM file for manufacturing, login into the CMS testbed web frontend, and upload the design to the customer database for placing an order.

6.3.2.2.2 Randomness

The factors such as customer arrival timing, order type, and order design can be randomly decided by the customer role.

6.3.2.3 Security administrator

The security administrator will be independent from the customer and hacker roles. The administrator will process the audit data from both the cyber and physical environments and give alarms when cyber-physical alerts are identified.

6.4 Case Study 1: 3D Printing Infill Structure Attack

The case study 1 is on the additive manufacturing process, or the 3D printing process. Over recent years, there are a growing number of researches (Zeltmann et al. 2016a; Chhetri, Canedo, and Faruque 2016; C. Song et al. 2016; Wu et al. 2016; Yampolskiy et al. 2016; Belikovetsky, Yampolskiy, et al. 2017) involving the 3D printing processes. In this case study, SQL injection and infill seam defect (Vincent et al. 2015) were used to test and validate the proposed methodology.

6.4.1 Cyber-Physical Attack Design

The cyber-physical attack on 3D printing uses attack vector SQLi and unknown attack repackaging. Three orders in total will be placed to the CMS testbed: a legitimate order, a repackaged order, and a malicious order by SQLi. The physical attack payload is the infill design alteration. More specifically, malicious infill defects are caused by changing the "STL" file. It is difficult to observe by inspectors or customers as the change cannot be observed from the exterior. However, the structural stiffness of a 3D printing test piece with infill void may be reduced by 14% (Sturm et al. 2014). As shown below, there are five different designs of malicious defect infill shape.

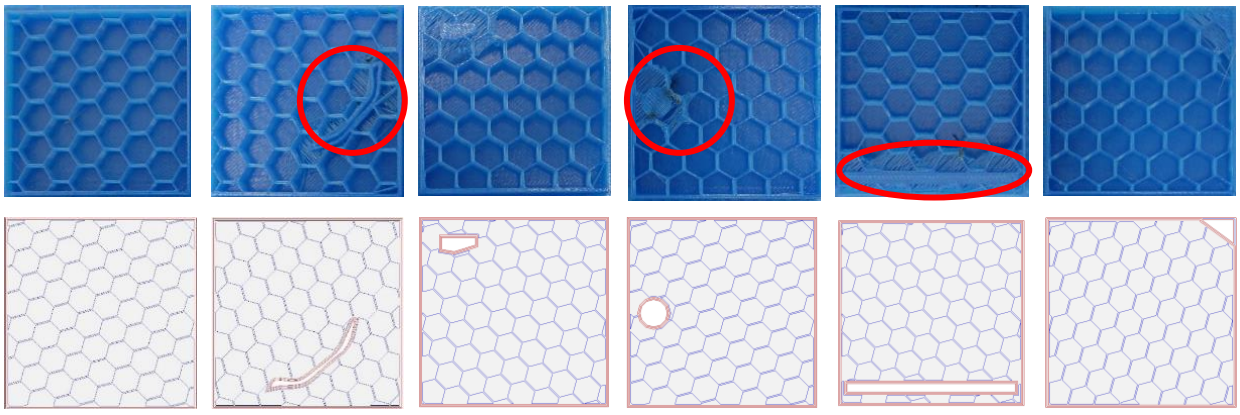


Figure 37 Five Types of Infill Defect Patterns Camera & Simulation View

To simulate this attack in CMS testbed, a malicious "STL" file is sent to the database. 3D printer proceeds to manufacture the part with malicious void while sensors collect physical data in the process. To make the attack realistic and randomized, the design selected for intrusion detection training set and the design for repackaging and SQLi attack is different. As shown in Figure 37, the design 2-2 seam defect is selected for the training set.

6.4.2 Attack Guideline

An attack guideline is provided to a person who operates as an attacker with the training of cyber-security knowledge. The attacker can switch between different roles during the experiment to create normal as well as malicious network activities. Each role is played on a different virtual machine. For example, Table 18 shows that five actions are executed from five different computers.

Table 18 3D printing attack guideline for student attacker

Actions	Tasks	Guideline
User #1	1. Log into user MTW account. 2. Upload one of the legitimate "STL" file. 3. Sign off MTW account	1. Execute user #1-3 action in random order and random time. 2. Make note of order, begin and end time for evaluation.
User #2	1. Log into user BYD account. 2. Upload one of the legitimate "STL" file. 3. Sign off BYD account.	
User #3	1. Log into user YPL account. 2. Upload one of the repackaged "STL" file. 3. Sign off YPL account.	
SQLi	1. SQLi database from login interface: Enter "jws'; -- " (include space) as user name, leave password blank. 2. Randomly choose a legitimate user. 3. Download its legitimate "STL" file. 4. Edit with a random malicious design previously defined. 5. Upload the malicious "STL", delete the previous order. 6. Submit order. 7. Logoff user account.	1. Execute SQLi anytime between the completion of first legitimate user and the end of the experiment. 2. Make note of username, begin and end time for evaluation.
Interference	1. Use Nmap intense scan attack on customer host.	1. Execute Nmap scan interference anytime during the experiment. 2. Make note of username, begin and end time for evaluation.

6.4.3 Attack Detection Result Analysis

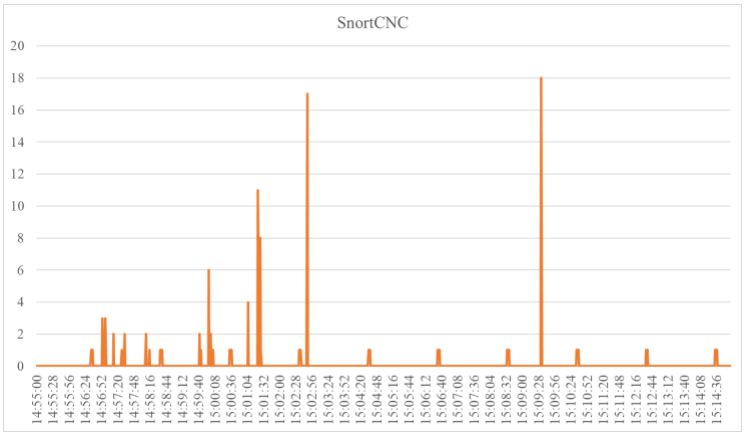
In the experiment, two cyber alert sources and 15 physical alert sources monitored CMS testbed activity in real time. The cyber alert sources are the analysis results of network traffic and host log file changes. The network-based intrusion detection software (SNORT) is using a rule-based algorithm to detect cyber-attack vectors whereas the host-based intrusion detection software

(OSSEC) monitors the change in the critical file directory and alerts if there is any suspicious change.

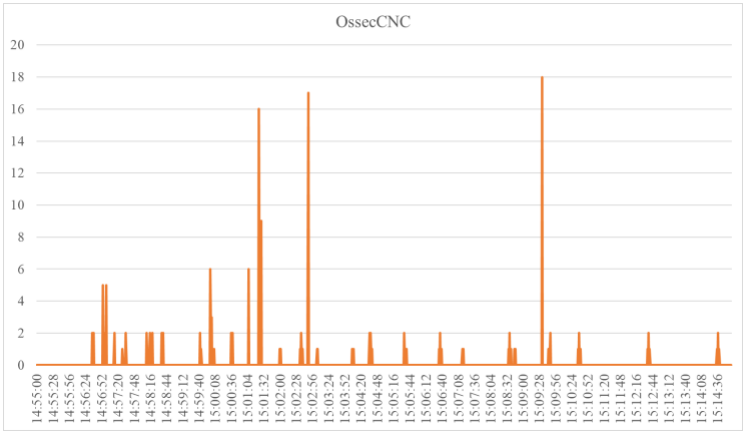
The physical alert sources are the analysis results of physical data from multiple sensors installed on the CMS testbeds. In the supplier testbed, there are (1) conveyor microphone, (2) conveyor current sensor, (3) heating chamber current, (4) heating temperature, (5) CNC microphone, (6) CNC three-axis accelerometer, (7) avoidance sensor, (8) mover robotic arm three-axis accelerometer, (9) welder robotic arm three-axis accelerometer, (10) 3D printing power meter, and (11) 3D printer camera. In the demander testbed, there are (12) robotic arm #1 three-axis accelerometer, (13) 3D scanner, and (14) robotic arm #2 three-axis accelerometer. In each of three-axis accelerometers, there are three channels of data: x-axis, y-axis and z-axis. As a result, there are 24 channels of data in total. Each channel of data is fed to the supervised machine learning algorithm, same as the rule-based detection method. Any suspicious activities are alerted.

6.4.3.1 Cyber alerts

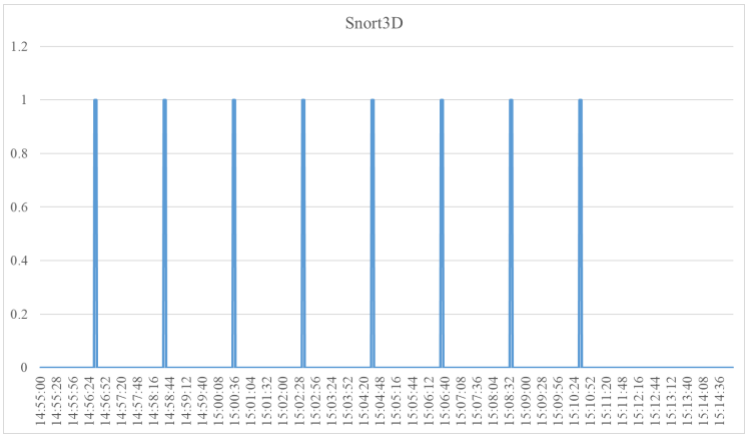
The cyber alerts collected from both CNC and 3D printing hosts are visualized in Figure 38. Clearly, the CNC milling host suffers heavier traffic with discrete cyber alert peaks, with a total of 137 Snort alerts and 173 OSSEC alerts. Using alert correlation with the attribute of IP address and time, meta-alerts are created. This activity reduces the investigation workload and prioritize the alerts. As shown in Table 2, total 5 meta-alerts are aggregated from 62 single Snort cyber alerts.



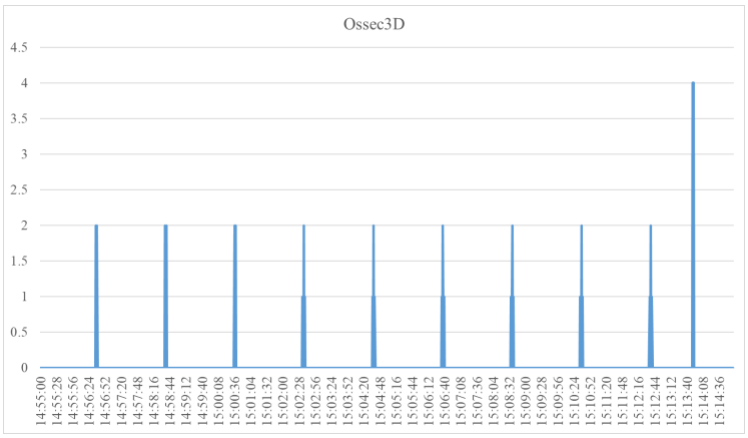
(a) CNC Host Snort Alert Number Per Second



(b) CNC Host Snort Alert Number Per Second



(c) 3D Printing Host Snort Alert Number Per Second



(d) 3d Printing Host Ossec Alert Number Per Second

Figure 38 CNC Attack Cyber Alert Distribution

Below are cyber meta-alerts correlated from 3D printing data host Snort software. The meta-alerts were listed that correlated more than at least 2 alerts, listing them in a hierarchy based on correlated attributes and correlated alerts. As shown in Table 19, five meta-alerts are aggregated from 62 single Snort cyber alerts.

Those five meta-alerts are high-level Snort alerts. They are strong evidence of an intrusion, but not necessarily a successful cyber-physical attack. As a result, further correlation with physical alerts is necessary.

Even though five meta-alerts is a small amount of work to investigate the alerts one by one, in a real network environment, the number of alerts can exponentially increase because of the network traffic complexity and noise.

Table 19 3D printing database Snort Cyber-Meta alerts

Start Time	End Time	Correlated Alerts	Correlated Attributes	Source/Destination IP	Alert Content
15:01:23	15:01:28	32	3	192.168.56.107 ->192.168.56.102	Executable code was detected; Attempted Information Leak; Misc activity.

15:02:48	15:02:48	17	3	192.168.56.103:49388->192.168.56.102:80	Error Based SQL Injection
14:59:58	14:59:58	6	3	192.168.56.105:41678->192.168.56.102:80	Error Based SQL Injection
15:01:06	15:01:06	4	3	192.168.56.102:10009->192.168.56.107:45814	Attempted Denial of Service

As shown in Table 20 below, the host-based intrusion detection software OSSEC's meta-alerts are a reflection of alerts from the Snort. It is because the alert log directory of Snort "ubuntu->/var/log/snort/alert" is under the monitoring of OSSEC. Moreover, the OSSEC provides additional alerts from directory of authentication log "ubuntu->/var/log/auth.log" potentially because of a '*SSH insecure connection attempt (scan)*'.

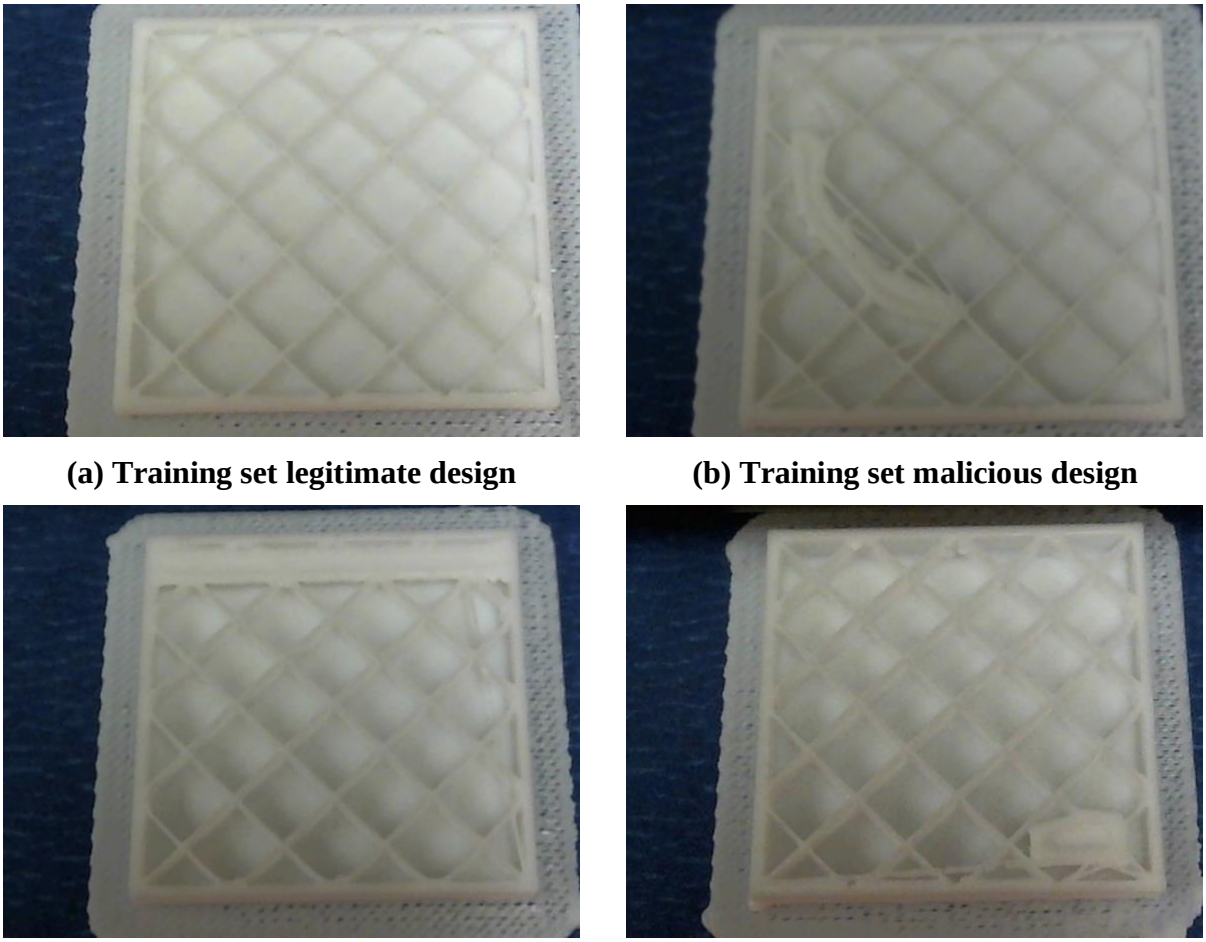
Table 20 3D printing database OSSEC Cyber-Meta alerts

Start Time	End Time	Correlated Alerts	Correlated Attributes	File Directory	Alert Content
15:01:24	15:01:28	32	2	ubuntu->/var/log/snort/alert	ICMP PING undefined code; SHELLCODE x86 inc ebx NOOP; SCAN nmap XMAS
15:02:49	15:02:49	17	2	ubuntu->/var/log/snort/alert	Error Based SQL Injection
15:00:00	15:00:02	9	2	ubuntu->/var/log/snort/alert	Error Based SQL Injection
15:01:06	15:01:06	6	1	ubuntu->/var/log/auth.log; ubuntu->/var/log/snort/alert	'SSH insecure connection attempt (scan).'; Did not receive identification string from 192.168.56.107; COMMUNITY SIP TCP/IP message flooding directed to SIP proxy

The correlated alerts represented the attack activities planned in the attack scenarios well. The rest of the alerts, which are uncorrelated, are a mixture of false alarms caused by normal

customer traffic, or true positives from network scans that didn't correlate because the gap time exceeded the window time reduced its priority. For Snort, between experiments 14:58 and 15:08 a total of 176 alerts were generated. The correlation reduced the alerts to 4 meta-alerts consisting of 59 single alerts. For OSSEC, the experiment also generated a total of 176 alerts, which reduced to 4 meta-alerts consisting of 64 single alerts.

6.4.3.2 Physical alerts



(a) Training set legitimate design **(b) Training set malicious design**

(c) Training set malicious design #1 **(b) Training set malicious design #2**

Figure 39 Training and testing product sample

The camera is the main source of the physical alerts in this experiment. As shown in below Figure 39 , the training set design for intrusion detection machine learning algorithm is different

from the experiment testing set. Even though the original pictures are taken in an imperfect condition—shadows and dark edges from 3D printing heating plate are shown in the picture—reasonable feature extraction technique can reduce the false alarms and reach a detection rate to nearly 100%.

As shown in Table 21, the 3D printing process generated 61 alerts. Each experiment was carried out for 1 hour, with a total of 3 hours for three customer orders. The order from customer_1 started at 15:48 aggregated 23 alerts into a meta-alert while the order from customer_3 started on Day 2 at 13:40 aggregated 38 alerts into a meta-alert.

Table 21 3D Printing Physical Alert List

Start Time	End Time	Number of Alerts	# of Correlated Attributes	Correlated Attributes	Alert Content
13:40	15:10	38	4	Time similarity; Sensor_sim_equi; Manu_Process; UID.	<PIDA-Message_NA> <Create_Time_2018-10-12 13:40:00> <Analyze_Time_NA> <3D_Camera_1_1> <KNN_classifier_k_1_feature_12> <UID_Customer 3> <Order_20181010_3D_T2> <3D_PLA_1> <SupID_NA> <Additive_ Plastic > <Potential_infill_defect>
15:48	16:52	23	4	Time similarity; Sensor_sim_equi; Manu_Process; UID.	<PIDA-Message_NA> <Create_Time_2018-10-10 15:48:00 > <Analyze_Time_NA> <3D_Camera_1_1> <KNN_classifier_k_1_feature_12> <UID_Customer 1> <Order_20180708_3D_T1> <3D_PLA_1> <SupID_NA> <Additive_Plastic> <Potential_infill_defect >

By now, cyber and physical alerts have been successfully extracted from cyber and physical domains. The next step is to correlate those alerts to cyber-physical meta-alerts via pre-defined attributes.

6.4.3.3 Cyber-Physical Meta-Alerts

The available pre-defined similarity attributes to correlate cyber and physical alerts include (1) destination IP with the manufacturing process and (2) source IP with user ID. The network environment setup and correlation of IP address, manufacturing process and customer ID is listed in Table 22. Each independent entity is assigned an IP address. The CNC and 3D printing customer database have the addresses 192.168.56.101 and 192.168.56.102. They belong to the destination IP type in cyber alerts because customers or hackers visit the hosts via Internet. The customers and unknown traffic are assigned with IP addresses from 192.168.56.103 to 192.168.56.107.

It is worth noting that customer 1 has two IP addresses. It is because the SQLi attacker maliciously logged into customer 1's account from 192.168.56.105. Even though the true identities (SQLi attacker and Nmap Scan) are labeled in brackets in Table 22, they were unknown for security administrators in the experiment during data analysis.

Table 22 Network environment and cyber-physical correlation for case study 1

IP address	Type	Manufacturing Process	Customer ID
192.168.56.101	**Destination IP**	**CNC milling**	**-**
192.168.56.102	**Destination IP**	**3D printing**	**-**
192.168.56.103	**Source IP**	**-**	**Customer 1**
192.168.56.104	**Source IP**	**-**	**Customer 2**
192.168.56.105	**Source IP**	**-**	**Customer 1 (SQLi)**
192.168.56.106	**Source IP**	**-**	**Customer 3**
192.168.56.107	**Source IP**	**-**	**Unknown (Nmap Scan)**

With the network environment clearly defined in Table 22, the correlations of "destination IP - manufacturing process" and "source IP - customer ID" are yielded as well, as shown in Figure 40 below.

Similarity	Subtractive_Manufacturing_Wood	Additive_Manufacturing_PLA
192.168.56.101	1	0
192.168.56.102	0	1

(a) Destination and Manufacturing Process Correlation

Similarity	UID_Customer_1	UID_Customer_2	UID_Customer_3
192.168.56.103	1	0	0
192.168.56.104	0	1	0
192.168.56.105	1	0	0
192.168.56.106	0	0	1

(b) Source IP and Customer ID Correlation

Figure 40 Correlation Matrix based on case study 1 network environment

The cyber meta-alerts aggregated from the cyber domain and physical meta-alerts aggregated from the physical domain can be further correlated based on the correlation matrix. The correlation process can be visualized according to Figure 41 below.

The cyber and physical alerts are generated by IDS and machine learning algorithms along the main timeline as the production proceeds in the CMS environment.

The cyber and physical alerts are first correlated into cyber meta-alerts and physical meta-alerts. The single low-level alerts are generated in the hundreds, if not thousands, which are time-consuming to process with many false alarms. This step can reduce false alarms randomly generated from network traffic and the physical production environment.

The cyber meta-alerts and physical meta-alerts are further correlated to the cyber-physical meta-alert. As shown in Figure 41 below, the cyber physical correlation process generated: (i) a cyber-physical meta-alert with highest priority, (ii) a physical meta-alert with medium priority, and (iii) a cyber meta-alert with low priority.

The cyber-physical meta-alert has a clear source and consequence from the information combined from previous cyber and physical alerts: the order comes from customer_1 from IP addresses 192.168.56.103 and 192.168.56.105. The IP addresses provide multiple cyber alerts, with clear attempt of SQL injection attack. The order caused a 3D printer alert with potential infill defect. The cyber domain attack alert lasted from 14:59:58 to 15:02:48. The physical domain attack alerts began at 13:40:00 on day two and didn't end until the job finished.

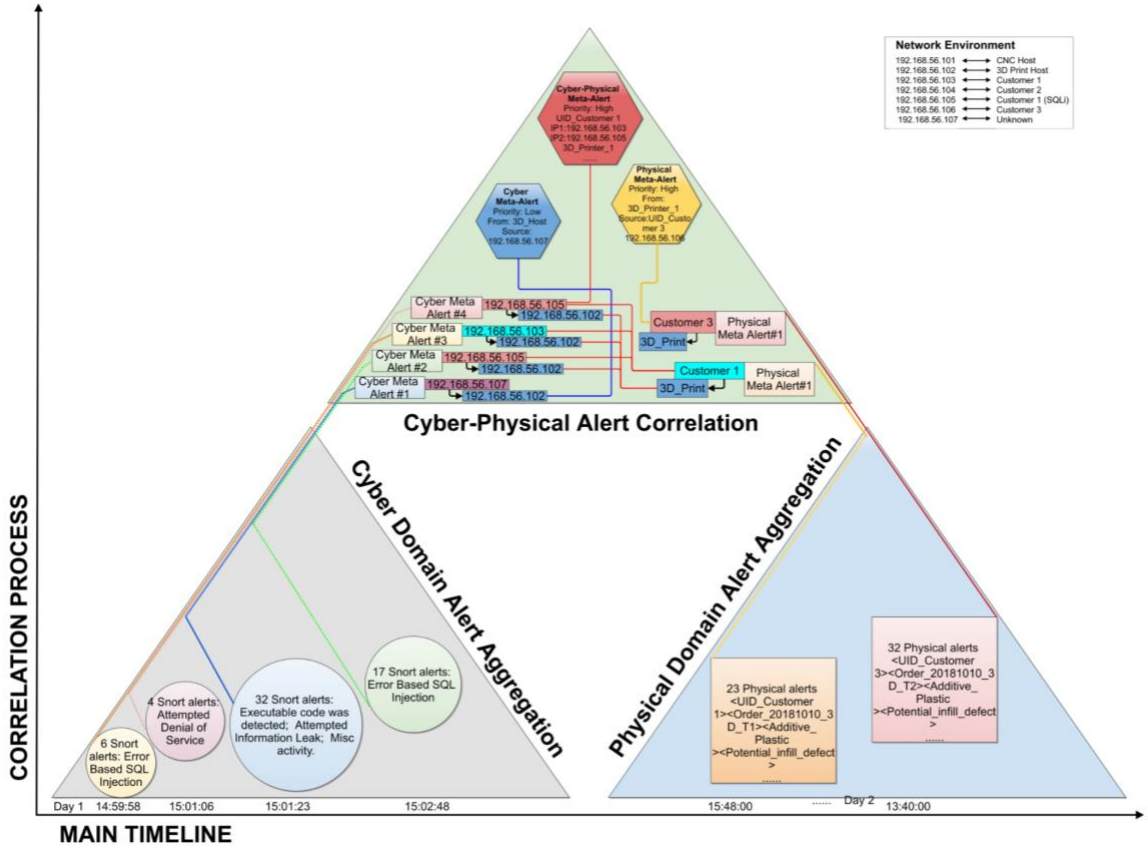


Figure 41 Correlation process diagram

The physical meta-alert has a clear consequence from the 3D printing process—potential infill defect. However, there is not any high-level cyber meta-alert that shows the intrusion. It is possibly coming from a unknown exploit that will not trigger the cyber alert. It is also possible that the physical production process met operation mistakes or defects. The physical domain attack alerts began at 15:48:00 and ended when the job finished.

The cyber meta-alert has a clear attack source without any clear physical consequence. It is possible to be a cyber-attack from outside without a physical domain intrusion. The cyber domain attack alert lasted from 15:01:23 to 15:01:28.

6.4.3.4 Evaluation

To evaluate the attack detection results, the original experiment record is presented in table below. The experiment operating students that played customers and hackers kept a record of attack sequence and timing. The table was not used for previous analysis, but can be used to verify the correctness of the prediction made from a cyber-physical meta alert.

In order to present the experiment attack record to the correlated meta-alerts, a comparison between attack and alert is listed in Table 24 below. From the comparison, the meta-alerts can effectively reflect the attack source, target and timing. For the repackaging attack, there are not specific intrusion times because the malicious activity to repackage the design happened outside the CMS environment, and the malicious pattern was unknown to NIDS (Snort) and HIDS (OSSEC).

Table 23 Experiment operator record on regular activity and attack

Task summary	Task Type	Begin Time	End Time
· Normal customer with legitimate design. · Customer ID: 1 · User ID: MTW	**Regular Customer activity**	**2:59**	**3:00**
·Normal customer with legitimate design. · Customer ID: 2 · User ID: BYD	**Regular Customer activity**	**3:02**	**3:03**
· Customer under repackaging attack with malicious design. · Customer ID: 3 · User ID: YPL	**Attack**	**3:04**	**3:07**
· Act as a hacker, use SQLi attack randomly attack legitimate customer. · Attacker detection: Attack customer 1.	**Attack**	**3:00**	**3:02**
· Act as hacker, use Nmap intense scan attack on 3D printing and CNC milling data host.	**Noise**	**3:01**	**3:06**

The alert priority also provides a useful guideline for investigating the attack. Clearly, the SQLi and repackaging attacks have a higher priority over the Nmap scan, which did not bring any physical consequence. Between SQLi and repackaging attack, both are malicious to the CMS environment, while the SQLi has a strong cyber evidence shown at the alert level. In a real production environment, it also depends on the company's security policy in deciding the alert priority of the different type of alerts.

Table 24 Attack VS Alert Comparison

Attack	Alert
SQL Injection · Target: 3D host, 192.168.56.102. · Source: Hacker, 192.168.56.105. · First attempt: 15:00, download design. · Second attempt: 15:02, upload malicious design.	**Cyber-physical meta-alert (Priority: high)** · Cyber Domain: 14:59:58 to 15:02:48, SQLi, DoS; from Snort on 192.168.56.102. · Physical Domain: 13:40:00 (D2), additive manufacturing infill. · Possible Source: Customer_1, 192.168.56.103 and 192.168.56.105.
Repackaging attack (unknown) · Target: 3D printer. · Source: customer_3, 192.168.56.106.	**Physical meta-alert (Priority: medium)** · Physical Domain: 15:48:00, additive manufacturing infill defect, 3D printer. · Possible Source: customer_3, 192.168.56.106.
Nmap scan (Noise interference) · Target: 3D host, 192.168.56.102. · Source: Hacker, 192.168.56.107. · Attack attempt: 15:01	**Cyber meta-alert (Priority: low)** · Cyber Domain: 15:01:23 to 15:01:28, Executable code was detected; Attempted Information Leak; Miscellaneous activity; from Snort on 192.168.56.102. · Possible Source: unknown source, 192.168.56.107

6.4.4 Summary

The case study 1 is a comprehensive example to show to entire process from cyber to physical domain, from low level to high-level alert. It is a case study proved the following points:

- The cyber and physical domain monitoring can detect various types of attacks.
- The cyber-physical alert correlation method precisely detects the cyber-physical attacks and correlate to root cyber alert.

- The correlation process reduced the number of alerts from 371 alerts (cyber alerts 310, physical alerts 61) to 3 meta-alerts, with a reduction rate of 99.1%.
- The detection accuracy is improved from 49.6% (correlated alarms 184, total cyber 371) to 100%.
- In the cyber domain, the known attack SQLi and unknown attack repackaging both can be detected with the assistance of physical detection.
- In the physical domain, the physical payload can be detected in real time with high accuracy, no false alarm is reported.

In case study 1, 3D printing is selected as an example for cyber-physical attack detection. The physical detection reaches an accuracy of 100% even though different training and testing data sets and product design were used. In the next case study, CNC machining will be used as an example to evaluate how the cyber-physical alert correlation will perform when the physical detection rate is relatively low with a large amount of physical false alarms.

6.5 Case Study 2: a CNC Spindle Speed and Feed Speed Attack

In case study 2, a representative subtractive manufacturing process: CNC milling process with wood material is attacked. Similarly, the cyber-physical attack on CNC milling can use attack vector SQLi and unknown attack repackaging. Three orders in total will be placed to the CSST: a legitimate order, a repackaged order, and a malicious order by SQLi. By exploiting the system vulnerability, a hacker can inject malicious specifications, such as spindle speed, feed speed, or even tool path into the CMS environment. The change can be harmful to the tool's life, equipment safety and design structure.

Similary to case study 1, 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 could generate higher amplitude of acoustic data during the milling process.

Different from case study 1, the physical detection accuracy for both accelerometers is relatively lower compared to the 3D printing image classification. The unavoidable consequence is the false positive alarms: the actual product is legitimate but the monitoring system gives alarms. The alert correlation and prioritization analysis for the physical domain will be emphasized.

6.5.1 Cyber-Physical Attack Design

In this section, a change in spindle speed in the milling operation is captured for further research. In the real case, fast rotation speed can cause over wear of a tool—a tool with a too slow rotation will risk being broken by shear force in the feeding direction. In the scenario, spindle speed is maliciously altered from 1200 rpm to 2000rpm.

The CNC milling cyber-physical attack will be involved with both SQLi and repackaging. Three orders in total will be placed to the CSST: a legitimate order, a repackaged order, and a malicious order by SQLi.

The physical payload will involve the change of two critical milling parameters: spindle speed and feed speed in the milling operation. The manipulation of CNC spindle and feed speed can cause over wearing, tool breakage and a rough finish without an obvious change in the dimensions.

In this experiment, the normal range of spindle speed and feed speed were defined in a range shown in Table 25. The customer order design should be within the legitimate range, while the attack will try to destroy the machine using the malicious range.

Table 25 The CNC Milling Process Feed and Spindle Speed Range

	Legitimate Range	Malicious Range
Feed Speed	$65 \pm 5\ in/min$	$75 \pm 5\ in/min$
Spindle Speed	$1000 \pm 100\ rpm$	$2000 \pm 100\ rpm$

Four sets of training data are collected within with the reference of range. However, the attack parameters are randomly decided by the student hacker without notice of the security administrator.

Table 26 The CNC Milling Training Dataset Parameter

	Feed Speed	Spindle Speed
Training Set Legitimate 1	66.5	1000
Training Set Legitimate 2	61.5	900
Training Set Malicious 1	77.5	2050
Training Set Malicious 2	72.5	1950

6.5.2 Attack Guideline

Similarly, the attack guideline is provided to the student hacker. The student is isolated from the CMS testbed operators and security administrators. The student attacker switch between different roles during the experiment to create normal and malicious network activities according to Table 27.

Table 27 CNC milling process attack guideline

Actions	Tasks	Guideline
User #1	**1. Log into user MTW account.**	**1. Execute user #1-3 action in random order and random time.**
	2. Upload one of the legitimate "Gcode" file. 3. Sign off MTW account	2. Make note of order, begin and end time for evaluation.
User #2	1. Log into user BYD account. 2. Upload one of the legitimate "Gcode" file. **3. Sign off BYD account.**	
User #3	**1. Log into user YPL account.**	
	2. Upload one of the repackaged "Gcode" file. **3. Sign off YPL account.**	
SQLi	**1. SQLi database from login interface: Enter "jws'; -- " (include space) as user name, leave password blank.**	**1. Execute SQLi anytime between the completion of first legitimate user and the end of the experiment.**
	2. Randomly choose a legitimate user.	2. Make note of username, begin and end time for evaluation.
	3. Download its legitimate "Gcode" file. 4. Edit with a random malicious feed speed within $75 \pm 5\ in/min$ and spindle speed within $2000 \pm 100\ rpm$ range. 5. Upload the malicious "Gcode", delete the previous order. **6. Submit order.**	
	7. Logoff user account.	
Interference	**1. Use Nmap intense scan attack on customer host.**	**1. Execute Nmap scan interference anytime during the experiment.**
		2. Make note of username, begin and end time for evaluation.

6.5.3 Attack Detection Analysis

Compared to case study 1, case study 2 was carried out in the same testbed environment, with the same monitoring systems, cyber-attack and detection methods. There are differences in the physical target and payload; the physical attack targets the CNC milling machine with

malicious manufacturing specifications. The physical monitoring method use accelerometers and acoustic sensor, which are different from previous image analysis. As a result, the highlight of this section is how to use the correlated method in the physical domain to mitigate the false positive alerts.

6.5.3.1 Cyber alerts

In case study 2, the cyber domain attack methods, monitoring system and analysis method are same.

6.5.3.2 Physical alerts

To give a better overview, Table 28 shows all 24 channels of alert sources from the physical sensor. Among those, the heat treatment current sensor and the CNC milling accelerometer sensor give a total of 4011 alerts through the three orders' production period.

Table 28 CNC process Physical Alert List

	Sensor Name	Sensor ID	Number of Alerts
1	conv_micro	1001	0
2	conv_current	2001	0
3	heat_current	2002	526
4	heat_temp	3001	0
5	cnc_micro	1002	0
6	cnc_acc_x	4001	1006
7	cnc_acc_y	4002	1260
8	cnc_acc_z	4003	1219
9	avoid_sens	5001	0
10	mover_acc_x	4004	0
11	mover_acc_y	4005	0
12	mover_acc_z	4006	0
13	weld_acc_x	4007	0
14	weld_acc_y	4008	0
15	weld_acc_z	4009	0
16	3D_power	6001	0
17	3D_img	7001	0

18	tb2_arm1_acc_x	4010	0
19	tb2_arm1_acc_y	4011	0
20	tb2_arm1_acc_z	4012	0
21	3D_scan	8001	0
22	tb2_arm2_acc_x	4013	0
23	tb2_arm2_acc_y	4014	0
24	tb2_arm2_acc_z	4015	0

Among the 4011 alerts, the distribution of alerts according to three different orders is shown in Table 29. Knowing two out of three orders is malicious; the heat-current gives clear guidance in detecting the intrusion. Customer_3's order only had 1 alert, possibly a false alarm. However, for CNC accelerometer x, y and z axis data, each customer received hundreds of alerts.

Table 29 Physical alerts distribution based on customer's order

	Customer_1	**Customer_2**	**Customer_3**
heat_current	256	269	1
cnc_acc_x	389	342	275
cnc_acc_y	208	485	567
cnc_acc_z	461	422	336

Without further analysis of the data, the physical alert correlation methods were directly implemented based on sensor similarity, manufacturing process similarity, time similarity and user identification similarity.

As shown in Table 30, 4011 of the physical alerts are correlated into three physical meta-alerts. The meta-alerts are listed in hierarchical order based on number of correlated attributes and correlated alerts. The meta-alert with higher order in the hierarchy will be granted higher priority for alert correlation and investigation. The accumulation of the alert numbers makes a simple

"voting" system: the more alerts correlated from the entire system, the higher priority it will be granted.

Table 30 CNC Milling Meta-Alert List

Start Time	End Time	Number of Alerts	Correlated Attributes	Alert Content
16:53	17:05	1518	Time; Sensor ID; User ID.	<PIDA-Message_NA> <Create_Time_2018-10-12 16:53:00> … <CNC_Acc_1_x, CNC_Acc_1_y , CNC_Acc_1_z , Heat_Current_1> <UID_Customer 2> …
16:38	16:53	1314	Time; Sensor ID; User ID.	<PIDA-Message_NA> <Create_Time_2018-10-12 16:38:00> … <CNC_Acc_1_x, CNC_Acc_1_y , CNC_Acc_1_z , Heat_Current_1> <UID_Customer 1> …
17:05	17:19	1179	Time; Sensor ID; User ID.	<PIDA-Message_NA> <Create_Time_2018-10-12 17:05:00> … <CNC_Acc_1_x, CNC_Acc_1_y , CNC_Acc_1_z , Heat_Current_1> <UID_Customer 3> …

The "voting" system is a part of the alert correlation process. It utilizes the overall accuracy of the physical data analysis to mitigate false alarms coming from one sensor. Case study 2 is an extreme example: one accelerometer comprises three channels of data, one of the channels provides large amount of false alarm. The additional current sensor mitigates and confirms the malicious activity of customer_1 and customer_2's orders, and lowers the priority of the meta-alert from customer_3.

6.5.3.3 Evaluation

Similarly, the experiment attack record and the correlated meta-alerts are listed in Table 31 below. Attack key points such as the attack source, target and timing are highlighted. Different from the previous case study, there is an alarm generated with no corresponding attack activity, which is a false alarm (false-positive).

Table 31 Attack VS Alert Comparison

Attack	Alert
SQL Injection	**Cyber-physical meta-alert (Priority: high)**
· Target: CNC host, 192.168.56.101. · Source: Hacker, 192.168.56.105. · First attempt: 16:17, download design. · Second attempt: 16:18, upload malicious design.	· Cyber Domain: 16:17 to 16:18, SQLi, DoS; from Snort on 192.168.56.101. · Physical Domain: 16:53, CNC milling acceleration. · Possible Source: Customer_2, 192.168.56.104 and 192.168.56.105.
Repackaging attack (Unknown)	**Physical meta-alert (Priority: medium)**
· Target: CNC milling. · Source: customer_1, 192.168.56.103.	· Physical Domain: 16:38:00, CNC milling acceleration, CNC milling process. · Possible Source: customer_1, 192.168.56.103.
No attack activity (False alarm)	**Physical meta-alert (Priority: medium)**
· Any alerts generated are false positives.	· Physical Domain: 17:05:00, CNC milling acceleration, CNC milling process. · Possible Source: customer_3, 192.168.56.106.
Nmap scan (Noise interference)	**Cyber meta-alert (Priority: low)**
· Target: CNC host, 192.168.56.101. · Source: Hacker, 192.168.56.107. · Attack attempt: 16:18	· Cyber Domain: 16:19, Executable code was detected; Attempted Information Leak; Miscellaneous activity; from Snort on 192.168.56.101. · Possible Source: unknown source, 192.168.56.107

With the alert correlation prioritization algorithm and multi-sensor voting effect, the false alarm is listed as a lower priority, after all the true alarms with physical consequences. By prioritizing the meta-alerts, the alert correlation can effectively mitigate the physical false alarms.

Even though this case study is intentionally carried out with false alarms, there are several ways to improve the accuracy of physical detection in general. One way is improving the feature extraction process via feature engineering. Secondly, different algorithms can have a significant difference in accuracy. Thirdly, the training data is also critical for detection accuracy.

For physical detection in the manufacturing process, a feature is a good data representation of a symptom, phenomenon or measurement. It requires domain knowledge and a data processing technique. A good understanding of and experience with different types of manufacturing data can improve the process. For example, the skewness and kurtosis feature can improve the accuracy of power consumption detection accuracy. Moreover, data science techniques such as automated feature engineering can also help creating and selecting features to improve the accuracy.

The machine learning algorithms affect the detection accuracy, false positive rate and speed drastically. In this experiment, kNN algorithm were implemented with comparatively low accuracy and high speed. From our previous work, algorithms such as random forest can greatly enhance the detection accuracy. Different types of manufacturing data also respond differently to various algorithms.

The training data is a critical factor for accuracy because it defines the representation of legitimate and malicious classes of data. In theory, a larger training dataset includes more samples can improve the accuracy. For manufacturing, because of the variation of the design, computer simulation can also generate training data that better matches the actual data collected from the

manufacturing process. Simulation system such as digital twin for Cyber-Manufacturing System, can provide training data for manufacturing process prediction and detection.

6.5.4 Summary

In this case study, CNC milling process specification attack is used to investigate the alert correlation result under the physical alert false alarms. To increase the chances of receiving false alarms, (i) acceleration data is selected because of the lower detection accuracy, (ii) relatively less sophisticated kNN machine learning algorithm is selected with limited detection accuracy, and (iii) different CNC milling specifications (but within a reasonable range) is used during testing and training. As a result, the experiment received around 33.8% of false alarms in CNC accelerometer data set.

The alert correlation process results show that (i) the correlation and prioritization process can decrease the false alarm priority for investigation, (ii) the percentage of false alarm reduced from 33.8% in CNC accelerometer alerts to 25% in meta-alerts.

6.6 Case Study 3: a Multiple Robotic Arm Speed Attack

From the previous two case studies, the effectiveness of cyber-physical alert correlation was proven and the correlation under the condition of false alarm in the physical domain was discussed. However, the previous case studies only attacked a single machine. In this case study, the alert correlation when multiple machines under the same cyber-physical attack will be demonstrated.

For case study 3, robotic arms' user-perceived robot state is attacked. In the CSST testbed, there are two robotic arms on an assembly line controlled by one web-application user interface.

Two robot state - normal mode and maintenance mode - are predefined. An attacker can use the SQLi attack method intrude into the robot mode to change the whole assembly line operating speed.

6.6.1 Cyber-Physical Attack Design

In order to attack the robotic arms without safety risk and damage, the potentially dangerous robot maintenance mode is set up as slower than the normal mode. In fact, changes such as operating angle and pattern can also be changed, but could potentially damage the testbed, and also very explicit from observation and sensor reading.

6.6.2 Attack Guideline

The normal robotic arm state operating speed is set at 10 degrees/second, while the maintenance state is set at 5 degrees/second. The operator can set up the operating state via the SQL based web-application control interface.

The attacker can log into the operator's account via SQLi. Once login, the attacker can freely modify the operating mode while the testbed is operating. However, the operator web-application based interface will not show the change if the webpage is not refreshed manually. Under this situation, the operation and safety could be damaged via the unknown maintenance mode.

6.6.3 Attack Detection Analysis

Compare to case study 1 and 2, the case study 3 carried out in the same testbed environment, and same monitoring systems. The difference of the attack detection in case study 3 is: there are alerts coming from two different equipment within a system. As a result, this section will highlight the correlation for alerts come from different equipment via one cyber-physical attack.

As shown in Table 32 below, two meta alerts with 40 and 28 single alerts are presented. They are fully correlated via all four similarity attributes: time, sensor ID, manufacturing process and user ID. It means all those alerts happen on one machine within one production job from the same customer.

Table 32 Robotic Arm Assembly Line Meta-Alert List

Start Time	End Time	Number of Alerts	Correlated Attributes	Alert Content
16:20	16:22	40	Time; Sensor ID; Manufacturing Process; User ID.	<PIDA-Message_NA> <Create_Time_2018-10-12 16:20:00> … <Arm1_Acc_1_x> <UID_Customer 2> … <Assembly_Process > <Potential_Assemble_Mistake>
16:38	16:40	28	Time; Sensor ID; Manufacturing Process; User ID.	<PIDA-Message_NA> <Create_Time_2018-10-12 16:38:00> … <Arm2_Acc_1_x> <UID_Customer 2> … <Assembly_Process > <Potential_Assemble_Mistake>

The two meta-alerts share a lot of similarity: they are both from same customer_2, they happened in the same manufacturing process, and the sensor is installed on similar equipment. In this case, even though the time similarity is not met, the two meta-alerts should be further correlated into one meta-alert. The similarity based alert correlation process realizes this function automatically.

6.6.4 Summary

The benefit of such an act is: automatically correlate alerts (1) comes from similar equipment or sensors, (2) happens within the same manufacturing process, or (3) comes from the same user. These scenarios are most likely to happen during industrial security incident: the hacker would bring influence on a large scale to increase the physical consequence.

The limitation of this function is it could correlate alerts happened during the same order process within the same environment, but actually are independent. To mitigate this effect, the security investigator should look into each correlated event to see if there is an actual connection.

6.7 Case Study 4: a Supply Chain Attack

In case study 4, the attack from the supply chain is investigated. A cyber-physical supply chain attack is an attack that damages a manufacturing service provider by targeting raw materials, parts or products from its supply network. In a CMS environment, geographically distributed manufacturing equipment is managed by the global business center. The attacker can send a wrong part or manufacture a malicious part that will be sent to the assembly manufacture which can cause further physical consequence.

6.7.1 Cyber-Physical Attack Design

To simulate a supply chain, the CSST supplier testbed manufactures cubes and feeds to demander testbed as shown in Figure 42. The cube will be inspected via 3D scanning for legitimacy. The hacker uses SQLi or any alternative attack vector intrude into the supplier database and changes the order type to make supplier testbed ships cube with a different specification to the demander testbed.

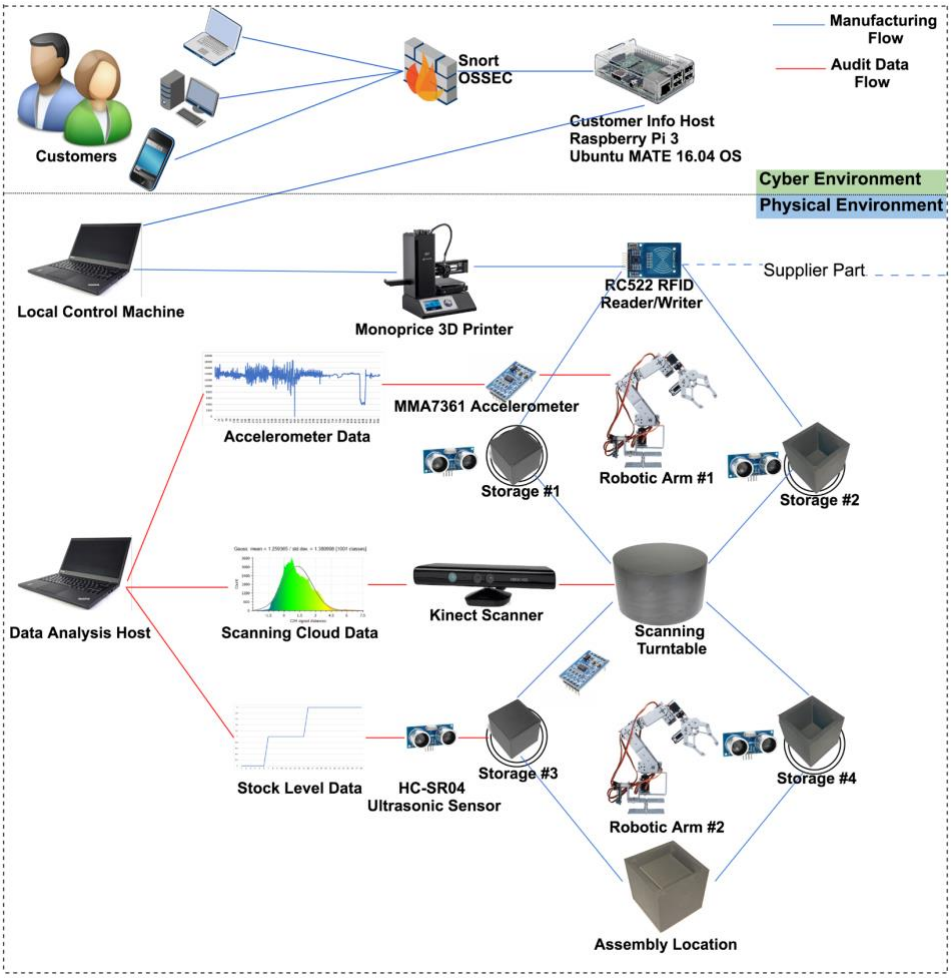


Figure 42 Demander testbed production flow

6.7.2 Attack Guideline

As shown in Figure 43 below, a hacker can send an alternative design product instead of the legitimate design requested by the supplier. The two design are dimensionally similar however largely different in design features. The part shipped from a supplier may in batch or in a package that cannot be discovered until the production process.

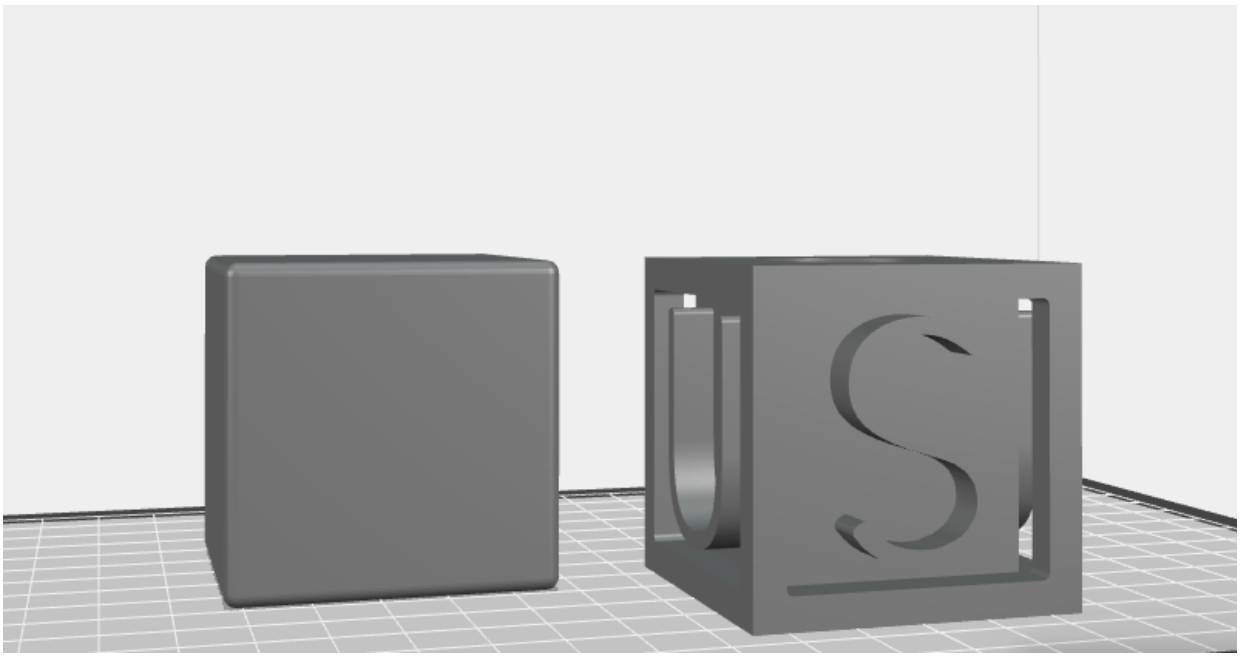
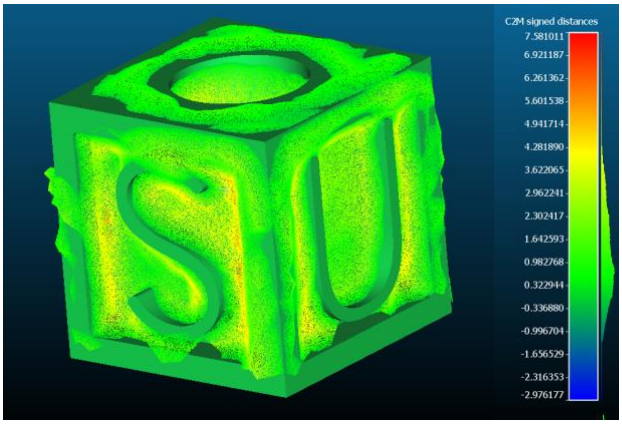


Figure 43 Alternative design (left) and original legitimate design request from supplier

6.7.3 Attack Detection Analysis

The 3D scanner is located before the assembly process in the Demander-Testbed. It is an Xbox 360 Microsoft Kinect Sensor controlled by Skanect 3D Scanning software. The software captures object images and exports to a “STL” file. The “STL” file is post-processed in the cloud, comparing the dimension difference from the original design.



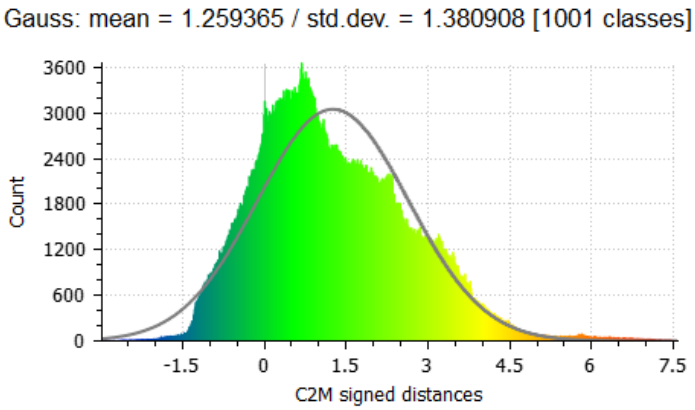


Figure 44 3D scanning model compare

To inspect the overall dimension, as shown in Figure 44 above, the distance between the scanning model and the original model follows a normal distribution with a mean value of 1.26 mm with a standard deviation of 1.38. It means the scanned part is on average 1.26 mm larger than the original design, which is beyond the 0.05 mm tolerance. The part from the supplier is defective.

As shown in Table 33 below, the 3D scanning inspection gives a single physical alert. Because the inspection is a single step procedure, rather than a process, there is only one alert. However, the physical alert provides alert correlation content for root cause: customer_3 or supplier CSST_1.

Table 33 3D Printing Physical Alert List

Start Time	End Time	Number of Alerts	Alert Content
10:58	10:58	1	<PIDA-Message_04-1> <Create_Time_2018-8-14 10:58:00> <Analyze_Time_NA> <3D_Scanner_1_1> <Tolerance_Inspection> <UID_Customer 3> <Order_20180814_3D_ASM> <3D_PLA_1> <SupID_CSST_1> <Additive_ Plastic > <Dimensional_Change>

The investigation will trace back to supplier CSST_1 using the alert correlations. However, because of the data confidentiality and customer privacy, order and security relevance data will not be shared across the supplier and demander. But the request of further investigation can be sent over the CMS network.

6.7.4 Summary

This case study discusses the supply chain attack detection and correlation problem. CMS as a future vision of manufacturing system will broaden the scope of the supply chain from a large corporation to small business. The alert correlation method can provide a potential source of attack through the supply chain when facing such an attack. However, because of the data confidentially and privacy, the investigation will be requested to the supplier.

Inspection before and after the production for raw materials and parts are necessary for the era of Cyber-Manufacturing. It is a combination of both improve the production quality control and cyber-physical security.

6.8 Conclusion

This section presented four case studies based on a Cyber-Manufacturing System Security Testbed (CSST). The four case studies are all cyber-physical attacks but serve different purpose:

- Case study 1 is a comprehensive example. Cyber-physical attacks with three different types of cyber-attack vector: (i) unknown attack, (ii) known attack and (iii) attack influence that causes false alarms were discussed. The case also included a full cyber-physical attack detection based on a highly accurate 3D printing infill detection vision detection example. The result shows that the cyber-physical alert correlation method can accurately correlate cyber and physical domain alerts and find the root cause.
- Case study 2 discussed the situation when a physical detection system generates false alarms and how the cyber-physical alert correlation methods mitigates such a problem. The case used CNC milling process accelerometer data with a less sophisticated kNN algorithm, which reduces accuracy and provides large amount of false alarms. The result shows that the proposed method can reduce the false alarm priority and overall percentage.
- Case study 3 discussed the situation of multiple machines under attack at the same time. It is a common scenario for industrial security incidents as the large scope or attack increases the influence and consequence. The correlation method can correlate the alerts even though they are not from the same machine. Attributes such as user ID, sensor and equipment ID are utilized.
- Case study 4 discussed the situation when an attack comes from the supply chain, which is a common scenario for current and future manufacturing systems. The inspection and correlation method can alert of the problem and provide direction for further investigation.

Potential issues such as data confidentiality, privacy and inspection across supply chain are discussed.

In conclusion, the case studies attempt to prove that the cyber-physical attack detection and correlation method can effectively reduce the alert numbers, reduce the false alarms, and trace to root causes—despite intentional influences on the cyber domain and limitation of detection accuracy in the physical domain. Issues such as supply chain attacks and large domain attacks are discussed as well. The correlation method can provide valuable information for security investigators to trace for the root cause through supply chain of CMS environment.

To generalize and implement the process of intrusion detection and correlation in CMS for cyber-physical attacks, a five-step framework DACDI is presented in next chapter.

Chapter 7

Implementation Framework

Security is a process, not a product. In this chapter, generalize the cyber-physical intrusion detection and correlation method into an implementation framework. This five-step framework aims for help security specialists applying intrusion detection and correlation methods to cyber-physical manufacturing system despite its scope, architecture or manufacturing process type.

A five-stage framework—DACDI (Define, Audit, Correlate, Disclose, Improve)—is proposed as follows. The DACDI framework is a collection of practices, techniques, procedures, and analyses structured for detecting intrusion, reducing the influence of the intrusions, and improving the Cyber-Manufacturing System security after intrusion incidents. This high-level framework allows additional practices and tools to be included. Professionals in manufacturing can follow this framework and adapt it to specific manufacturing environment for cyber-physical intrusions purposes.

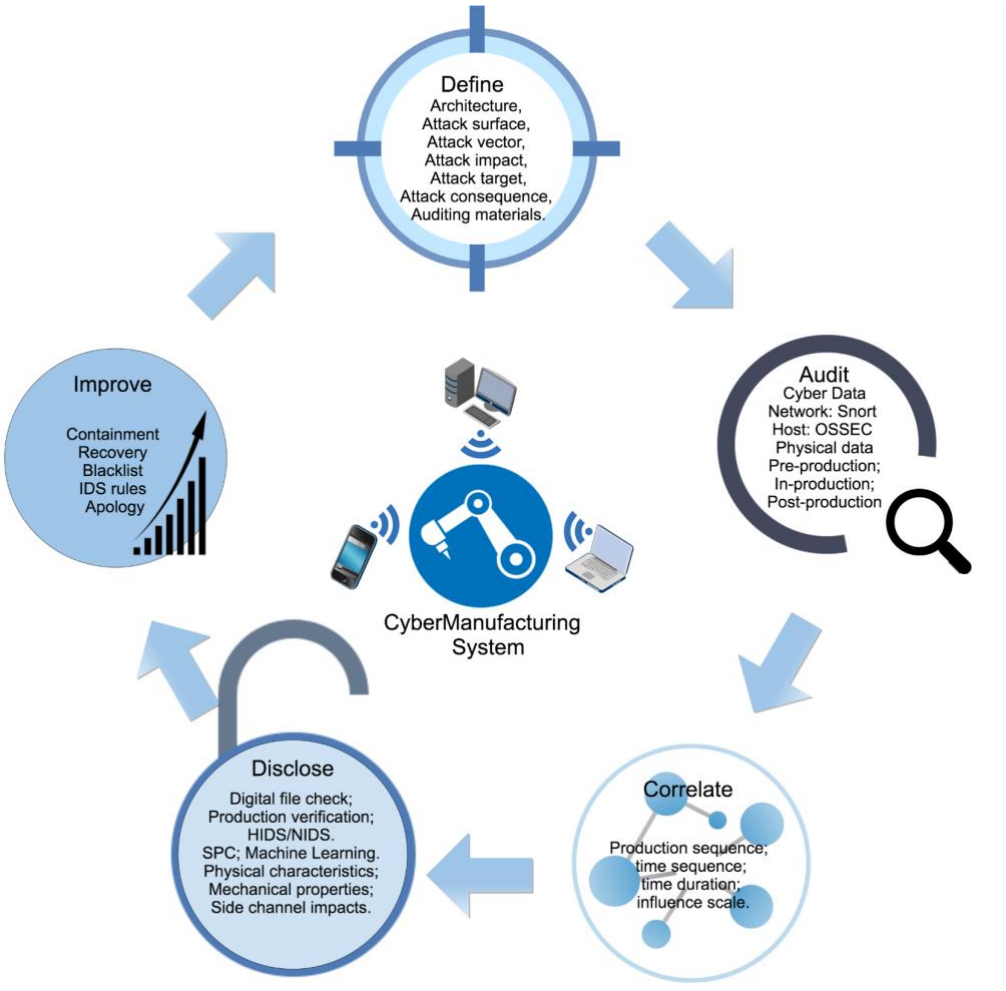


Figure 45 DACDI five-stage intrusion detection approach

The DACDI is a continuous improvement system as illustrated in Figure 45, consisting of five stages:

- Define
- Audit
- Correlate
- Disclose
- Improve

7.1 Define

The first stage identifies seven **A**s: Architecture, Attack surface, Attack vector, Attack impact, Attack target, Attack consequence and Audit material. The objective of this step is to define the kinds of cyber and physical data that need to be selected as audit data for the second stage. The analysis of architecture, attack surface and attack vector can identify cyber data selection from the cyber-security perspective. The analysis of attack impact, attack target and attack consequence can identify physical data from the manufacturing process perspective. Both types of data are summarized and further utilized in audit and disclose stages (2 and 4).

7.1.1 Define the Architecture

The architecture of a victim system needs to be defined in the first place by studying implementing manufacturers. The manufacturers can follow any architecture that fits their business model, manufacturing process and customer needs such as CMS hierarchical five-layer architecture (Z. Song and Moon 2016b), cloud manufacturing concept architecture (Adamson et al. 2015), or reference architecture model industry 4.0 (Bitkom, Vdma, and Zvei 2016).

By defining a flow diagram, a detailed flow and dynamic relationship can be presented. For example, Figure 1 shows a process flow in a standardized CMS architecture. This example architecture of CMS consists of five layers: the Application/User Layer, the Application Interface Layer, the Global Core Service Layer, the Integrated Connection Layer, and the Physical Provider Layer. The first layer—Application/User Layer—includes users and consumers. The second layer—Application Interface Layer—includes support techniques as a buffer of inventory and information processing. The third layer—Core Service Layer—is the global information hub of machine resources, personnel, geographically locations, logistics, user information, etc. The fourth layer—Integrated Connection Layer—is a local analysis and self-control network center. The fifth layer—Physical Provider Layer—is the physical layer which includes all the manufacturing resources in factory floor.

However, different types of business and manufacturing may adopt a different architecture or develop an appropriate architecture. Understanding of the adopted architecture can help defining the attack surface.

7.1.2 Define the Attack Surface

The attack surface of a CMS environment is a list of different points where an attacker can try to enter data or extract data from the environment. To inject the cyber-physical attack, the input data is the only way that attackers can put malicious code, design, or commands to the physical layer. By analyzing the input data, where to set up network or host sensors to monitor the intrusion can be determined. According to Hutchins (Hutchins et al. 2015b), data inside a manufacturing system includes, but is not limited to: design specs, CAD files, financial info, user data, inventory, design feedback, production feedback, etc.

Figure 1 illustrates an attack surface analysis for a CMS five-layer architecture. As shown, the potential place an attacker can inject malicious data into CMS is the input data at the application/user layer. However, the malicious data can flow through the CMS environment by data exchanges.

In following four steps, a taxonomy of cross-domain attacks on CMS (Wu and Moon 2017b) is adopted to define the attack vector, attack impact, attack target and attack consequence.

7.1.3 Define the Attack Vector

The attack vector is the method an attacker can exploit system vulnerabilities. For example, code injection, shellshock, are frequently observed attack vectors in manufacturing (IBM-Security 2017). The purpose of defining the attack vector in intrusion detection is to find measurements in network and host activities and logs to monitor the intrusion. For example, if the CMS customer platform is a web application based uploading system with SQL databases, the following attack vectors are possible: Shellshock, Buffer Overflow, Race Condition, Cross-Site Request Forgery (CSRF), Code Injection, Repackaging, Virus, and Worms. A checklist-style table can be used in such a process and the table should be updated with time as new attack vectors show up. Different CMS enterprises can modify the checklist according to the different network environment.

7.1.4 Define the Attack Impact

According to the cross-domain attack taxonomy, the direct impact from a cyber-attack incident can be but not limited to: privilege compromise, user compromise, file compromise, denial of service, malware installation, etc. The attack impact is the direct result/payload of an attack incident. It can help define cyber audit materials. For example, an attacker can use shellshock attack gaining the super user privilege and change the CNC milling machine

specification. The privilege elevation is the attack impact. Using network/host-based intrusion detection can stop the intrusion at an early stage before causing any consequence. There are open-source tools that can be used to monitor the network and host activities such as SNORT, OSSEC, etc.

7.1.5 Define the Attack Target

The attack target is the ultimate goal attackers aim for. In cyber-physical attacks, the targets for attackers are physical targets. For example, the Stuxnet worm attack's target was the controllers of the centrifuge in a nuclear power plant. In a CMS environment, the physical target can be sensors, controllers, actuators, machines and equipment, manufactured parts, or even human beings.

The targets selected in CMS will be the source of physical audit data. After defining the target, the next step is analyzing the consequences of the target under attack.

7.1.6 Define the Attack Consequence

Defining the attack consequence is a way to identify physical evidence of intrusions in progress or after intrusions. Such evidence is sometimes referred as an attack's manifestation (Kemmerer and Vigna 2002). The physical consequence can be defective product, machine manipulation, malfunction and breakage, or loss of system availability. With respect to the selected attack consequence, physical auditing material such as side-channel monitoring data, inspection rules can be defined in the next step for detection purpose.

One of the methods to explore the attack consequence is Failure Mode and Effects Analysis (FMEA)—a structured systematic technique for failure analysis for reliability study. In intrusion detection in a CMS environment, FMEA can be used for reviewing machines and assembly lines

to identify the vulnerability and failure modes, consequences and effects. By creating a CMS security FMEA worksheet, the potential failure modes and their effects on the whole CMS system can be recorded. It can also give guidelines on the placement of monitoring systems and indicate the criticality of the potential failure.

7.1.7 Define the Audit Data

The result of "Define" step will help define the audit data for intrusion detection. The audit data includes data from cyber environments such as network activities and host log, and data from physical environments such as temperature and energy consumption.

7.2 Audit

Audit, or data auditing, is the second stage of our approach. It is the process of collecting data for intrusion detection. In CMS, two types of data are collected for intrusion detection purpose: cyber data and physical data. Cyber data: (1) are capable of detecting amateur and known attacks, and (2) are used as evidence in sophisticated attacks to correlate with physical anomaly occurrence. Physical data: (1) are capable of detecting cyber-physical data quickly with high accuracy (Wu, Song, and Moon 2019; Wu et al. 2017; Z. Song et al. 2017), and (2) can also prevent machine malfunction and human mistakes as a by-product.

7.2.1 Cyber Data

Cyber audit data includes the data from network activity and host. In a computer network, network activity log data can be information, such as login attempts, network connections, or every data packet that appeared on the wire (Kemmerer and Vigna 2002). It can be monitored by Network-based Intrusion Detection System (NIDS). For example, Software Snort is a packet

sniffer that can monitor network traffic in real time. It checks each packet closely to detect a dangerous payload or suspicious anomalies.

A network host is a computer or other device connected to a computer network. A network host may offer information resources, services, and applications to users, or other nodes, on the network. It can be monitored by a host-based intrusion detection system (HIDS). For example, Software OSSEC can do log analysis, file integrity checking, Windows registry monitoring, centralized policy enforcement, rootkit detection, real-time alerting and active response (Timofte 2008).

7.2.2 Physical Data

The next step is collecting physical data from the manufacturing process in CMS environment. The manufacturing process is the main target of the cyber-physical attacks in CMS. After defining step, it should be clear what manufacturing process is within the scope of intrusion detection. In this section, the 3D printing process is analyzed.

3D printing, or additive manufacturing, is a key technology for future manufacturing systems; it is typically computer-controlled and can be integrated with the Internet. 3D printing systems have unique vulnerabilities presented by the ability to affect the internal layers of an object without affecting the exterior layers. To attack this process, malicious users can change the design or dimensions in the ".STL" file, so malicious defective parts could be manufactured without an alert.

To detect this type of potential attack, multiple types of physical data can be selected as audit data in 3D printing process. The vision monitoring method for additive manufacturing malicious void defect is proven effective with an accuracy of 95.51% (Wu et al. 2016). To collect

images from the additive manufacturing process, engineers can install cameras on top of the object (Wu et al. 2017), so cameras can collect cross-sectional views.

Acoustic emission generated by onboard stepper motors as a side channel data has also been used for monitoring additive manufacturing malicious infill void. By collecting the acoustic data from the 3D printing process, and comparing it to the simulated original design data, the method from Belikovetsky (Belikovetsky, Solewicz, et al. 2017b) can effectively detect the infill defect and stop the printing process for compromised objects.

7.3 Correlate

With the similarity-based alert correlation method defined, an alert correlation process provides a high-level view on the correlating process in a CMS environment. Based on previous research, a five-step process is proposed as a general principle for cyber-physical alert correlation in CMS.

7.3.1 Alert Normalization

The first step is to normalize the cyber and physical alerts collected from different nodes of CMS. The IDS software generates alerts and encodes them in different formats. These alerts are usually received by the correlation process from different software. The primary objective of alert normalization is to translate the features of each sensor alert into a generic format to reduce the number of alerts to be correlated.

For cyber alerts, the Internet Engineering Task Force (IETF) has proposed a generic representation of intrusion alerts to develop a standard known as **Intrusion Detection Message Exchange Format** (IDMEF). The IDMEF "defines data formats and exchange procedures for sharing information of interest to intrusion detection and response systems and to the management

systems that may need to interact with them" (H. Debar, Curry, and Feinstein 2007). An IDMEF alert message is composed of nine different components including create time, detect time, analyzer time, analyzer, source, target, classification, assessment and additional data. For cyber alert generated by software such as Snort or OSSEC, the alert can be translated into IDMEF message format.

For the physical alert, this work proposes a new **Physical Intrusion Detection Alert** (PIDA) for reporting alerts in CMS physical environment. The PIDS alert format is composed of 11 components including Alert message title and ID, Create Time, Analyzer Time, Sensor ID, Analyzer ID, User ID, Order ID, Equipment ID, Supplier ID, Manufacturing Process, Additional Information.

7.3.2 Alert Aggregation

The purpose of the alert aggregation process is to combine the alert caused by the same event or attack, create cyber meta-alerts and physical meta-alerts as defined in section 3.1 and 3.2. For example, for cyber domain, the same attack caused five NIDS alerts from snort and seven HIDS alerts from OSSEC need to be aggregated in this step into a cyber meta-alert. For the physical domain, the alert from the camera and power consumption meter on a 3D printer should be aggregated into a physical meta-alert.

The decision to aggregate two alerts based on the attribute feature varies between cyber and physical alerts. For the cyber alert, source IP, destination IP can be the attribute to correlate the alerts. For the physical alert, target type/manufacturing process, attack assessment/consequence can be the attribute to correlate the alerts.

7.3.3 Cyber-Physical Alert Correlation

The cyber-physical alert correlation process is to generate a strong meta-alert between the alert aggregation results. In this process, the temporal feature will not work because the window time between a cyber intrusion and a physical consequence is uncertain: an attacker can hide the payload of the malware long enough to make temporal feature fail.

The attribute-based technique can correlate the cyber and physical alerts based on CMS production flow characteristics. For example, the attribute such as customer type, customer ID, order ID, manufacturing process type, local supplier type, supplier ID can correlate the alerts in the customer database, design process, and manufacturing environment. In this step, a high-level cyber-physical meta-alert will be created.

7.3.4 Influence Analysis

To estimate the different impact of meta-alerts, factors such as the number of correlated alerts, number of shared similar attributes, affected entity type and number can be utilized for influence analysis. For example, a meta alert coming from multiple machines has more influence on CMS than alerts coming from a single machine; Similarly, a meta-alert correlated with three shared attributes will have a higher impact on another meta alert has only one shared attribute. The impact analysis can be defined differently based on CMS manufacturing type, scope, etc.

7.3.5 Alert Prioritization

The alert prioritization process can effectively discard alerts that is irrelevant or less important to a particular environment. For different environments, the security requirement and

policy will lead to different alert prioritization policy. Therefore, there is no absolute priority for an attack (Valeur et al. 2004). A general rule for CMS alert prioritization is a higher level of correlation ranks the higher level of priority. For example, the cyber-physical meta alert owns higher priority compared to cyber meta-alert or physical meta-alert; Cyber meta-alert or physical meta-alert also own higher priority than cyber or physical alert.

7.4 Disclose

Disclose is the fourth stage of the approach. The purpose of intrusion detection is to become aware of the intrusion and stop it as early as possible. The collection of methods from cybersecurity, machine learning, and quality control are implemented for CMS and cyber-physical attacks. The detection is therefore divided into three stages: pre-production, in-production and post-production.

7.4.1 Pre-production

In the pre-production stage, the inspecting relies on both human inspection and data analysis. Attacks being detected in the pre-production stage can be terminated before entering the production environment. It can reduce the influence of physical layer processes and reduce the recovery time and cost. No physical material or machines are damaged or wasted. No customer is influenced by the attack.

Digital file check is the first step of intrusion detection in CMS. The check can verify the CAD/CAM file and determine if there are any holes or non-closed shells that could cause malicious influence to the product or machine. The check is a part of the initial communication process with the customer, along with the steps of verifying the compatibility of the design (dimensions, complexity, material, etc.), as well as price estimation according to the material used, labor and urgency.

Production verification is a document sent from a CMS service provider to the customer. The purpose of this document is to: (1) give customers a preview of their order status, (2) approve the integrity of the order detail, such as design or parts, and (3) receive the approval from the customer to proceed to manufacture. As illustrated in section 2.1, network activities and host logs can be analyzed by software packages Snort and OSSEC. The results generated by the software are alerts with some levels of urgency.

7.4.2 In-production

As large amounts of data can be collected to monitor the production process, the in-production stage detection uses data analysis techniques. Attacks being detected during the production stage can be terminated before the attack causes further damage to machines and equipment, and reduces the waste of materials and time. Some material could be wasted, but the machine should not be damaged. Customers' orders can be processed by other machines or suppliers to reduce the schedule delay. Statistical process control (SPC) is also adopted to detect intrusions.

An example in Fig. 10 shows the malicious void attack on 3D printing. The attack makes the number of the pixels whose grayscale value is higher than 120 go beyond its upper control limits. The red dots higher than the upper control limits (UCL) are the defective areas malicious void shows.

Machine learning is a core enabling technology for CMS and other future visions of manufacturing systems. It has been used in quality control (Wuest, Irgens, and Thoben 2014), defect detection (Pernkopf and O'Leary 2003) and attack detection (Wu, Song, and Moon 2019) in manufacturing systems. It is also used in intrusion detection systems in cybersecurity. It can be

categorized as supervised learning and unsupervised learning. The signature-based detection is broadly used in firewall and intrusion detection systems. It uses the rule of supervised learning for the pattern of known attacks, such as detecting the syntax of a SQL code injection attack (Alnabulsi, Islam, and Mamun 2014). In manufacturing systems, supervised learning can also detect issues such as a malicious void in the 3D printing process (Wu, Song, and Moon 2019). Unsupervised learning assumes that an intrusion can be detected by observing a deviation from the normal or expected behavior of the system of the users using the rules of unsupervised learning. Compared to signature-based detection, it can provide more protection facing unknown exploit.

7.4.3 Post-production

In the post-production stage of detection, quality control (QC) measures in manufacturing processes are used. Attacks being detected in the post-production stage can be terminated before the final product is delivered to the customer and causes further influence. Material and time are wasted, and an apology could be needed for notifying the customer of the delay in manufacturing.

The physical domain inspection can be classified into three groups: physical characteristics, mechanical properties and side-channel impacts (Pan et al. 2017a). QC measures for physical characteristics, including visual inspection, dimension measure, weight measure, 3D laser scanning, X-rays, and CTs. As nondestructive tests, it can be implemented on all products for inspection. Mechanical property tests, such as the tensile test, can be used to test wire, strip or machined samples with either circular or rectangular cross-section. It is a destructive test that can be used in sampling inspection rather than 100% inspection.

Side-channel impacts are mostly discussed in cryptography and refer to cases where attackers do not leverage information from plaintext or ciphertext, but from physical characteristics

of cryptosystems (Pan et al. 2017a). Side channel information such as temperature, power signature, timing, utilization rate, and queue time can be used for analyzing intrusion detection.

7.5 Improve

The last stage of DACDI approach is improving. It is a collection of countermeasures based on the disclosed result to terminate the intrusion, improve the security of the victim system, and respond to any damage to the system and the customers.

The first step after detection is stopping the damage. The methods of containment include, but are not limited to, disconnection, blacklisting the attacker, and adding detection rule.

A radical way to contain the situation is to disconnect the host from the local network, or even to disconnect the whole site from the connection; for example, disconnect a manufacturer temporarily from CMS environment before the security team finds the cause of the problem. The business impact can be migrated because the service can be taken over by other suppliers before the recovery.

Security teams can collect attack information such as: IP address, attack payload, and packet information. By putting the attacker's IP address into blacklist, the CMS environment will drop all future packets from that IP address. The attack information can be added to the blacklist, and Snort/OSSEC detection rules can be used to improve the intrusion detection.

Once the attack is contained, the next step is the recovery stage. The attack has left the CMS environment with backdoors and vulnerabilities. Before recovery production, the security team must fix the bug before the attacker comes back. Reporting the problem to the service provider, and updating software and the operating system is necessary. If the attacker manipulated the database, the staff might be able to restore program and data files from the last trusted backup.

In general, the system needs to be more secure than before, so that the attacker cannot come back in CMS. Once an attacker has cracked a system, he or she often invites other attackers in to prove his or her skills (Boyle and Panko 2013).

After recovery of the software, the next step is to recover the hardware. The machine and equipment that have been manipulated need to go through a thorough inspection, repair, maintenance, or even a replacement could be needed.

If the attack has not been detected before production, it is possible that it has caused a delay in the customer's production schedule. It could also cause harm to an employee. It is important to give a prompt and sincere apology. From experience, downplaying the severity of the incident can cause worse influence than being honest.

7.6 Summary

The DACDI framework is an implementational guideline for manufacturing enterprise detecting and correlating cyber-physical attacks. Based on different operational structure, the data type and algorithm for detection and correlation should be adjusted for optimal results.

Chapter 8

Conclusion and Future Work

This dissertation presented a cyber-physical detection and correlation system for a cyber-physical manufacturing system. In this chapter, the conclusion, contribution and broad impact of this work were summarized. The limitations of this work are presented along with necessary future work.

8.1 Summary

This dissertation presented a cyber-physical attack detection and correlation system. This system is proposed for cyber-physical manufacturing systems, such as the Cyber-Manufacturing System. It is designed for the detection of cyber-physical attacks: an emerging attack intrudes via cyber-attack vector but causes physical consequences. To detect such an attack, this work utilizes available network and host-based intrusion detection software to monitor the cyber security domain, while applies supervised and unsupervised machine learning algorithms on physical security domain. To reduce the false alarms via integrating cyber and physical alerts, this work applies the similarity-based alert correlation method.

To better understand cyber-physical attacks, this work took a close look at cyber-physical attacks in manufacturing systems. Existing cyber-physical industrial security incidents are analyzed and further generalized into attack taxonomies to characterize cyber-physical attacks. Systemic cyber-physical attack scenarios are designed based on the taxonomies targeting manufacturing system physical domain in experiments.

To achieve a dataset to validate the method, this work also presented a security-oriented cyber-physical manufacturing testbed. The testbed consists of a cyber network environment, as well as physical manufacturing equipment and processes. Monitoring systems are integrated in both cyber and physical domains. The data collected from the testbed with attack scenarios are used in validation.

The intrusion detection heavily relies on training data. For this work, the historic data (Wu et al. 2017, 2018; Wu, Song, and Moon 2019) is used as training data for manufacturing processes

to detect generic issues in the manufacturing process; as the historic data accumulates by time, the detection accuracy will also increase. The historic data is a reliable source for the manufacturing process as the process is more under control compared to network activities. Attacks that act within manufacturing constraint but change the design features, tolerances, or/and accuracy cannot be guaranteed to be detected by historic data. This type of detection will reply more on design specific training data generated by computer simulation, such as the 3D printing process image simulation (Wu et al. 2016), CNC acoustic emission simulation (Wu, Song, and Moon 2019), and the growing popular digital twin (Tao et al. 2018) concept. KCAD method (Chhetri, Canedo, and Faruque 2016) proved the simulation data can be used in cyber-physical attack detection.

The experiments show that machine learning methods in physical domain detection give high accuracy—overall higher than 90%, with some cases reaching 100%. The alert correlation method can effectively reduce the total amount of alerts, especially the cyber alerts, by 99.1%; it can correlate physical alerts to cyber alerts for root cause analysis; and it can prioritize the true alarms and deprioritize the false alarms. It improves the overall detection accuracy from 49.6% to 100%, shortens the detection (investigation) time, and reduces the false alarm rate from 33.8% to 25% for the case studies.

The DACDI framework generalizes such a method into a five-step process. The process can be applied to cyber-physical manufacturing systems such as Cyber-Manufacturing System, Industry 4.0, or Smart Manufacturing system. Moreover, it could be modified to apply to different or more general cyber-physical systems where cyber-physical attacks could intrude.

8.2 Contribution

Machine Learning in cyber-physical intrusion detection. This work applied supervised and unsupervised machine learning (Wu, Song, and Moon 2019) to the physical data from the manufacturing process. For the 3D printing process, vision and the power consumption data source for machine learning were used. Three different machine learning algorithms were implemented with image classification. The anomaly detection method returned the highest accuracy of 96.1% in detecting a malicious defect in the printing process. In the CNC milling process example, two attack modes changing the part design and manufacture operation were designed. Acoustic signal is selected as source of physical data for the machine learning process. The same three machine learning algorithms implemented with the random forest algorithm returned the highest average accuracy of 91.1%. The technique of detecting malicious activities during the manufacturing process is validated with both simulation (Wu et al. 2016) and physical experiment (Wu et al. 2017).

Cyber-physical attack in manufacturing system analysis. The cyber-physical attack as one of the cross-domain attacks was not well understood (Yampolskiy et al. 2013) at the beginning of this research. To give an in-depth review, existing documentaries about Stuxnet and German steel mill cyber-physical security incidents were studied for characterizing the cyber-physical attacks. This work also provided the definitions, taxonomies and scenarios to provide a better understanding of cyber-physical attacks. Two taxonomies are proposed from both intrusion detection (Wu and Moon 2017b) and attacker (Wu and Moon 2018) perspectives. Moreover, a discussion about the detection period of known attacks and unknown exploits regarding the production period in a manufacturing system is presented.

Similarity-based alert correlation method for cyber-physical attack. An alert correlation methodology is developed for cyber-physical attacks in cyber-physical manufacturing system. The method applies a similarity-based alert correlation technique with newly defined attributes. It comprises three steps: cyber-alert correlation, physical-alert correlation, and cyber-physical alert correlation. In each step, attributes are defined based on the characteristic of cyber-physical attacks and CMS environment. The distinct manufacturing attributes, such as the sensor, manufacturing process are created and employed for similarity-based alert correlation. Moreover, a physical alert format PIDA (Physical Intrusion Detection Alert) is defined for cyber-physical alert correlation. The format is defined with reference to Intrusion Detection Message Exchange Format (IDMEF) with distinct manufacturing information. The manufacturing specific information, such as user ID, machine ID, sensor ID, and manufacturing process establishes a bridge between cyber and physical alert correlation.

Cyber-Manufacturing System Security Testbed (CSST). For CMS, the benchmark dataset to evaluate the intrusion detection system is not available. One of the reasons is that the cyber-physical attacks are new; researchers have limited knowledge and examples from real production systems. Moreover, current manufacturing systems are not designed to monitor cyber-physical attacks. As a result, the CSST testbed is established for data collection and validation. The physical components include equipment, controllers, sensors, and actuators from the component level in CMS shop floor. The computational components include web interfaces, IDS, and a discrete event simulation model from the system level of CMS. Both components collect data for intrusion detection analyses. The testbed can illustrate the process of customer orders, job allocation, manufacturing, post-processing, conveying, and transporting. The data collected from this testbed is shared in the research community.

Implementation framework for cyber-physical IDS in CMS. Previous intrusion detection studies focus on individual manufacturing processes, such as additive manufacturing or CNC machining, rather than considering the manufacturing system as a whole. Critical components for intrusion detection such as network, host, and quality control inspections are neglected in these works. A five-step intrusion detection framework—DACDI (*D*efine, *A*udit, *C*orrelate, *D*isclose, and *I*mprove)—is designed specifically for the CMS. A model CMS is used to collect cyber as well as physical audit data, and to demonstrate the feasibility of operating the intrusion detection system. It is a framework to implement the cyber-physical intrusion detection system in different CMS environments. It is also a collection of systematic and statistical analysis for detecting intrusions, reducing their influences, and improving the level of security after detection. Professionals in manufacturing, cyber-security, and control systems can adapt it as a guideline to detect intrusions in a CMS environment.

8.3 Limitations

While this work attempts to improve the intrusion detection of cyber-physical attacks in CMS with presented method, it can never achieve 100% security. This work is presented with some limitations.

Firstly, this work cannot effectively detect one of the attack types in manufacturing system: intellectual property theft. This attack type is a broadline problem; in some cases, such as Dragon fly attack, they are cyber-attacks, while in other cases, such as 3D printing smartphone (C. Song et al. 2016) eavesdropping, they act like a cyber-physical attack. However, in both cases, the attack payload does not leave consequences in the physical domain of CMS. As a result, this type of attack is not compatible with the cyber-physical attack detection and correlation method.

Secondly, the machine learning algorithms implemented were selected based on accuracy and previous work. More algorithms, such as deep neural networks, could be implemented to compare with current results in terms of accuracy, false-positive rate, and detection speed.

8.4 Future Work

For the similarity-based alert correlation method, the cyber-physical attributes need to be further developed and refined. Moreover, other alert correlation methods, such as sequential-based alert correlation methods, can be applied in the CMS cyber-physical alert correlation domain.

For the machine learning in the detection system, alternative algorithms, such as anomaly detection, and feature extraction techniques may be implemented to increase the physical alert accuracy.

For the CSST tested, computer simulation can be integrated to collect long-term, large-scale system level data from the physical domain for intrusion detection. Technologies such as wireless network, digital twin, and special industrial protocols such as MQTT (S. Lee et al. 2013) and MTconnect (Vijayaraghavan et al. 2008) can be integrated to simulate a cyber-physical manufacturing system with different setups. Security countermeasures such as blockchain or a software-defined network can be implemented with IDS to react to cyber-physical attacks.

To better study cyber-physical attacks, more taxonomies, attack scenarios can be published in the research community. Moreover, a public repository that documents industry security incidents can greatly help manufacturing security research.

To improve on DACDI framework, an application of such a process on a real manufacturing environment can help disclose the limitations and shortcomings. Moreover, the

framework can be extended to a broader audience, including the users of Industry 4.0 and Smart Manufacturing, similar cyber-physical systems, critical infrastructure, and so on.

Finally, the experiments to validate and test the intrusion detection system can be improved with more attack scenarios defined in Chapter four. Factors such as manufacturing defects, different attack vectors, such as malware, and different manufacturing processes can be used to test the effectiveness of the intrusion detection system.

References

Abad, Cristina, Jed Taylor, Cigdem Sengul, William Yurcik, Yuanyuan Zhou, and Ken Rowe. 2003. "Log Correlation for Intrusion Detection: A Proof of Concept." In *Proceedings of Annual Computer Security Applications Conference,* 255–64. Las Vegas, Nevada: Institute of Electrical and Electronics Engineers. doi: 10.1109/CSAC.2003.1254330

Adamson, Göran, Lihui Wang, Magnus Holm, and Philip Moore. 2015. "Cloud Manufacturing - a Critical Review of Recent Development and Future Trends." *International Journal of Computer Integrated Manufacturing* 30, 4-5: 347-380. doi: 10.1080/0951192X.2015. 1031704

Ahmadinejad, S.H., and S. Jalili. 2009. "Alert Correlation Using Correlation Probability Estimation and Time Windows." *2009 International Conference on Computer Technology and Development,* 170-175. Kota Kinabalu, Malaysia: Institute of Electrical and Electronics Engineers. doi: 10.1109/ICCTD.2009.22.

Alnabulsi, Hussein, Md Rafiqul Islam, and Quazi Mamun. 2014. "Detecting SQL Injection Attacks Using SNORT IDS." *Asia-Pacific World Congress on Computer Science and Engineering*. Nadi, Fiji: Institute of Electrical and Electronics Engineers. doi: 10.1109/ APWCCSE.2014.7053873.

Alvarez, Michelle, Nicholas Bradley, Pamela Cobb, Scott Craig, Ralf Iffert, Limor Kessem, Jason Kravitz, Dave McMillen, and Scott Moore. 2017. "X-Force Threat Intelligence Index 2017." *IBM*. www-01. ibm.com/common/ssi/cgi-bin/ ssialias?htmlfid=WGL03140USEN&.

Alves, Thiago Rodrigues, Mario Buratto, Flavio Mauricio De Souza, and Thelma Virginia Rodrigues. 2014. "OpenPLC: An Open Source Alternative to Automation." *Proceedings of the 4th IEEE Global Humanitarian Technology Conference,* 585–89. San Jose, California:

Institute of Electrical and Electronics Engineers. doi: 10.1109/GHTC.2014.6970342.

Belikovetsky, Sofia, Yosef Solewicz, Mark Yampolskiy, Jinghui Toh, and Yuval Elovici. 2017a. "Detecting Cyber-Physical Attacks in Additive Manufacturing Using Digital Audio Signing." http://arxiv.org/abs/1705.06454.

———. 2017b. "Detecting Cyber-Physical Attacks in Additive Manufacturing Using Digital Audio Signing."

Belikovetsky, Sofia, Mark Yampolskiy, Jinghui Toh, and Yuval Elovici. 2017. "Dr0wned - Cyber-Physical Attack with Additive Manufacturing." In *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*. http://arxiv.org/abs/1609.00133.

Bhuyan, Monowar H., Dhruba K. Bhattacharyya, and Jugal K. Kalita. 2017. "Alert Management and Anomaly Prevention Techniques." In *Network Traffic Anomaly Detection and Prevention. Computer Communications and Networks*, 171–99. Springer, Cham. https://doi.org/10.1007/978-3-319-65188-0_5.

Bilge, Leyla, and Tudor Dumitras. 2012. "Before We Knew It: An Empirical Study of Unknown Attacks in the Real World." *Proceedings of the 2012 ACM Conference on Computer and Communications Security -- CCS'12*, 833–44. https://doi.org/10.1145/2382196.2382284.

Bitkom, V, V Vdma, and V Zvei. 2016. "Implementation Strategy Industrie 4.0."

Bosch, Anna, Andrew Zisserman, and Xavier Munoz. 2007. "Image Classification Using Random Forests and Ferns." In *2007 IEEE 11th International Conference on Computer Vision*, 1–8. IEEE. https://doi.org/10.1109/ICCV.2007.4409066.

Boyle, Randall J, and Raymond R Panko. 2013. *Corporate Computer Security*. Prentice Hall Press.

Bradley, Nicholas, Michelle Alvarez, John Kuhn, and David McMillen. 2015. "IBM 2015 Cyber Security Intelligence Index." *IBM*, 24. https://doi.org/SEW03039-USEN-02.

Chandola, Varun, Arindam Banerjee, and Vipin Kumar. 2009a. “Anomaly Detection: A Survey.” *ACM Computing Surveys* 41 (3): 1–58. https://doi.org/10.1145/1541880.1541882.

———. 2009b. “Anomaly Detection.” *ACM Computing Surveys* 41 (3): 1–58. https://doi.org/10.1145/1541880.1541882.

Chhetri, Sujit Rokka, Arquimedes Canedo, and Mohammad Abdullah 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. https://doi.org/10.1145/2966986.2967050.

Clarke, Justin, and Rodrigo Marcos Alvarez. 2012. “SQL Injection Attacks and Defense.” https://books.google.co.uk/books?hl=en&lr=&id=Spm7UgBwzjIC&oi=fnd&pg=PR3&dq=sql+injection+hacking&ots=k-3DTHhjeJ&sig=fitmTZpZU1UDXBKWJp-2AiYO6Gg#v=onepage&q=sql injection hacking&f=false.

Contag, Moritz, Guo Li, Andre Pawlowski, Felix Domke, Kirill Levchenko, Thorsten Holz, and Stefan Savage. 2017. “How They Did It: An Analysis of Emission Defeat Devices in Modern Automobiles.” *Proceedings - IEEE Symposium on Security and Privacy*, 231–50. https://doi.org/10.1109/SP.2017.66.

Cuppens, Frédéric, and Alexandre Miège. 2002. “Alert Correlation in a Cooperative Intrusion Detection Framework.” *Proceedings - IEEE Symposium on Security and Privacy* 2002–Janua: 202–15. https://doi.org/10.1109/SECPRI.2002.1004372.

Debar, H., D. Curry, and B. Feinstein. 2007. “The Intrusion Detection Message Exchange Format (IDMEF).” https://doi.org/10.17487/rfc4765.

Debar, H, and A Wespi. 2001. “Aggregation and Correlation of Intrusion Detection Alerts.” *International Workshop on Recent Advances in Intrusion Detection* 2212: 85–103.

https://doi.org/10.1007/3-540-45474-8_6.

Debar, Herve. 2017. “What Is Behavior Based Intrusion Detection?” SANS. 2017. https://www.sans.org/security-resources/idfaq/what-is-behavior-based-intrusion-detection/2/6.

Delio, T., J. Tlusty, and S. Smith. 1992. “Use of Audio Signals for Chatter Detection and Control.” *Journal of Manufacturing Science and Engineering* 114 (2): 146. https://doi.org/10.1115/1.2899767.

Duro, João A, Julian A Padget, Chris R Bowen, and H Alicia Kim. 2016. “Multi-Sensor Data Fusion Framework for CNC Machining Monitoring.” *Mechanical Systems and Signal Processing* 67: 505–20.

Elhabashy, Ahmad E., Lee J. Wells, Jaime A. Camelio, and William H. Woodall. 2018. “A Cyber-Physical Attack Taxonomy for Production Systems: A Quality Control Perspective.” *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-018-1408-9.

Elhabashy, Ahmed. 2018. “Quality Control Tools for Cyber-Physical Security of Production Systems.” Virginia Polytechnic Institute and State University.

Elshoush, Huwaida Tagelsir, and Izzeldin Mohamed Osman. 2012. “An Improved Framework for Intrusion Alert.” In *Proceedings of the World Congress on Engineering*, I:4–9. http://www.iaeng.org/publication/WCE2012/WCE2012_pp518-523.pdf.

Garcia, Ramon Ferreiro, J. L C Rolle, and Javier Perez Castelo. 2011. “A Review of SCADA Anomaly Detection Systems.” *Advances in Intelligent and Soft Computing* 87 (August 2015): 405–14. https://doi.org/10.1007/978-3-642-19644-7.

Gates, Dominic. 2018. “Boeing Hit by WannaCry Virus, but Says Attack Caused Little Damage.” The Seattle Times. 2018. https://www.seattletimes.com/business/boeing-aerospace/boeing-

hit-by-wannacry-virus-fears-it-could-cripple-some-jet-production/.

Giraldo, Jairo, Esha Sarkar, Alvaro A. Cardenas, Michail Maniatakos, and Murat Kantarcioglu. 2017. "Security and Privacy in Cyber-Physical Systems: A Survey of Surveys." *IEEE Design and Test* 34 (4): 7–17. https://doi.org/10.1109/MDAT.2017.2709310.

Giraldo, Jairo, David Urbina, Alvaro Cardenas, Junia Valente, Mustafa Faisal, Justin Ruths, Nils Ole Tippenhauer, Henrik Sandberg, and Richard Candell. 2018. "A Survey of Physics-Based Attack Detection in Cyber-Physical Systems." *ACM Computing Surveys* 51 (4): 1–36. https://doi.org/10.1145/3203245.

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

Hadžiosmanović, Dina, Robin Sommer, Emmanuele Zambon, and Pieter H. Hartel. 2014. "Through the Eye of the PLC." In *Annual Computer Security Applications Conference*, 126–35. https://doi.org/10.1145/2664243.2664277.

Hansman, Simon, and Ray Hunt. 2005. "A Taxonomy of Network and Computer Attacks." *Computers and Security* 24 (1): 31–43. https://doi.org/10.1016/j.cose.2004.06.011.

Hutchins, Margot J., Raunak Bhinge, Maxwell K. Micali, Stefanie L. Robinson, John W. Sutherland, and David Dornfeld. 2015a. "Framework for Identifying Cybersecurity Risks in Manufacturing." *Procedia Manufacturing* 1: 47–63. https://doi.org/10.1016/j.promfg.2015.09.060.

———. 2015b. "Framework for Identifying Cybersecurity Risks in Manufacturing." *Procedia Manufacturing* 1: 47–63. https://doi.org/10.1016/j.promfg.2015.09.060.

IBM-Security. 2016. "2016 Cyber Security Intelligence Index." *IBM*. http://www-

01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&htmlfid=SEW03133USEN&attachment=SEW03133USEN.PDF.

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

———. 2018. “IBM X-Force Threat Intelligence Index 2018.” *IBM*. https://securityintelligence.com/2018-ibm-x-force-report-shellshock-fades-gozi-rises-and-insider-threats-soar/.

———. 2019. “IBM X-Force Threat Intelligence Index 2019.” *IBM*. https://www.ibm.com/security/data-breach/threat-intelligence.

IETF. 2018. “IETF | Internet Engineering Task Force.” 2018. https://www.ietf.org/.

Jain, Anil K, Arun Ross, and Salil Prabhakar. 2004. “An Introduction to Biometric Recognition.” *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY* 14 (1). https://doi.org/10.1109/TCSVT.2003.818349.

Jakobson, Gabriel, and Mark Weissman. 1995. “Real-Time Telecommunication Network Management: Extending Event Correlation With Temporal Constraints.” *Im*, 290–301. https://doi.org/10.1007/978-0-387-34890-2.

Jia, Hongbin, Yi Lu Murphey, Jianjun Shi, and Tzyy Shuh Chang. 2004. “An Intelligent Real-Time Vision System for Surface Defect Detection.” *Proceedings - International Conference on Pattern Recognition* 3 (February): 239–42. https://doi.org/10.1109/ICPR.2004.1334512.

Jie, Ma, Zhi Tang Li, and Wei Ming Li. 2008. “Real-Time Alert Stream Clustering and Correlation for Discovering Attack Strategies.” *Proceedings - 5th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2008* 4: 379–84.

https://doi.org/10.1109/FSKD.2008.522.

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

Julisch, Klaus, and Marc Dacier. 2004. “Mining Intrusion Detection Alarms for Actionable Knowledge.” In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM*, 366–75. https://doi.org/10.1145/775047.775101.

Jung, Jin Hyuk, Ju Young Kim, Hyeong Chan Lee, and Jeong Hyun Yi. 2013. “Repackaging Attack on Android Banking Applications and Its Countermeasures.” *Wireless Personal Communications* 73 (4): 1421–37. https://doi.org/10.1007/s11277-013-1258-x.

Kabiri, Peyman, and Ali A. Ghorbani. 2007. “A Rule-Based Temporal Alert Correlation System.” *International Journal of Network Security* 5 (1): 66–72. http://ijns.femto.com.tw/contents/ijns-v5-n1/ijns-2007-v5-n1-p66-72.pdf.

Karnouskos, Stamatis. 2011. “Stuxnet Worm Impact on Industrial Cyber-Physical System Security.” *IECON Proceedings (Industrial Electronics Conference)*, 4490–94. https://doi.org/10.1109/IECON.2011.6120048.

Karthikeyan, K.R., and A. Indra. 2010. “Intrusion Detection Tools and Techniques –A Survey.” *International Journal of Computer Theory and Engineering* 2 (6): 901–6. https://doi.org/10.7763/IJCTE.2010.V2.260.

Kaspersky Lab. 2017. “The State of Industrial Cybersecurity 2017.” *Business Advantage Group Limited*.

Kelley, Michael B. 2013. “The Stuxnet Attack On Iran ’ s Nuclear Plant Was ’ Far More

Dangerous ' Than Previously Thought." Business Insider. 2013. http://www.businessinsider.com/stuxnet-was-far-more-dangerous-than-previous-thought-2013-11.

Kemmerer, R.A., and G. Vigna. 2002. "Intrusion Detection: A Brief History and Overview." *Computer* 35 (4): supl27-supl30. https://doi.org/10.1109/MC.2002.1012428.

Khamphakdee, Nattawat, Nunnapus Benjamas, and Saiyan Saiyod. 2014. "Improving Intrusion Detection System Based on Snort Rules for Network Probe Attack Detection." *2014 2nd International Conference on Information and Communication Technology, ICoICT 2014*, no. May: 69–74. https://doi.org/10.1109/ICoICT.2014.6914042.

Kim, Ae Chan, Won Hyung Park, and Dong Hoon Lee. 2013. "A Study on the Live Forensic Techniques for Anomaly Detection in User Terminals." *International Journal of Security and Its Applications* 7 (1).

Kumar, M, S Siddique, and H Noor. 2009. "Feature-Based Alert Correlation in Security Systems Using Self Organizing Maps." *Proceedings of SPIE - The International Society for Optical Engineering* 7344 (Id). https://doi.org/10.1117/12.820000.

Kumar, Mohit. 2018. "TSMC Chip Maker Blames WannaCry Malware for Production Halt." The Hacker News. 2018. https://thehackernews.com/2018/08/tsmc-wannacry-ransomware-attack.html.

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

Lee, Keunsoo, Juhyun Kim, Ki Hoon Kwon, Younggoo Han, and Sehun Kim. 2008. "DDoS Attack Detection Method Using Cluster Analysis." *Expert Systems with Applications* 34 (3): 1659–65. https://doi.org/10.1016/j.eswa.2007.01.040.

Lee, Robert M, Michael J Assante, and Tim 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.

Lee, Shinho, Hyeonwoo Kim, Dong Kweon Hong, and Hongtaek Ju. 2013. *Correlation Analysis of MQTT Loss and Delay According to QoS Level*. *International Conference on Information Networking*. https://doi.org/10.1109/ICOIN.2013.6496715.

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

Liao, Hung-Jen, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. 2013. “Intrusion Detection System: A Comprehensive Review.” *Journal of Network and Computer Applications* 36 (1): 16–24. https://doi.org/10.1016/j.jnca.2012.09.004.

Lindsay, J. 2013. “Stuxnet and the Limits of Cyber Warfare.” *Security Studies* 22 (3): 365–404. https://doi.org/10.1080/09636412.2013.816122.

Maggi, Federico, and Stefano Zanero. 2007. “On the Use of Different Statistical Tests for Alert Correlation – Short Paper.” In *Recent Advances in Intrusion Detection*, 167–77. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-74320-0_9.

Manadhata, Pratyusa K., and Jeannette M. Wing. 2010. “An Attack Surface Metric.” *IEEE Transactions on Software Engineering* 3: 371–86.

Mary, A Caroline. 2015. “Shellshock Attack on Linux Systems – Bash.” *International Research Journal of Engineering and Technology*, 1322–25.

Mendes, Luís, Anneli Kangas, Kirsi Kukko, Bjarke Mølgaard, Arto Säämänen, Tomi Kanerva, Iñigo Flores Ituarte, et al. 2017. “Characterization of Emissions from a Desktop 3D Printer.” *Journal of Industrial Ecology* 21: S94–106. https://doi.org/10.1111/jiec.12569.

Mitchell, Robert, and Ing-Ray Chen. 2014. “A Survey of Intrusion Detection Techniques for Cyber-Physical Systems.” In *ACM Comput. Surv.* Vol. 46. https://doi.org/10.1145/2542049.

Monroy, Sergio A Salinas, Ieee Member, Ming Li, Ieee Member, Pan Li, and Ieee Member. 2018. “Energy-Based Detection of Defect Injection Attacks in IoT-Enabled Manufacturing.” In *2018 IEEE Global Communications Conference (GLOBECOM)*, 1–6. IEEE. https://doi.org/10.1109/GLOCOM.2018.8647631.

Moore, Samuel, Mark Yampolskiy, Jeffrey T. McDonald, Todd R. Andel, and Jacob Gatlin. 2016. “Buffer Overflow Attack’s Power Consumption Signatures,” 1–7. https://doi.org/10.1145/3015135.3015141.

Orebaugh, Angela, and Becky Pinkard. 2011. *Nmap in the Enterprise: Your Guide to Network Scanning*.

Pan, Yao, Jules White, Douglas C Schmidt, Ahmad Elhabashy, Logan Sturm, Jaime Camelio, and Christopher Williams. 2017a. “Taxonomies for Reasoning About Cyber-Physical Attacks in IoT-Based Manufacturing Systems.” *International Journal of Interactive Multimedia and Artificial Inteligence* 4 (3): 45–54. https://doi.org/10.9781/ijimai.2017.437.

Pernkopf, Franz, and Paul O’Leary. 2003. “Image Acquisition Techniques for Automatic Visual Inspection of Metallic Surfaces.” *NDT and E International* 36 (8): 609–17. https://doi.org/10.1016/S0963-8695(03)00081-1.

Peterson, Leif. 2009. “K-Nearest Neighbor.” Scholarpedia. 2009. https://doi.org/10.4249/scholarpedia.1883.

Qin, Xinzhou. 2005. “Dissertation: A Probabilistic-Based Framework for INFOSEC Alert Correlation.” Georgia Institute of Technology. https://doi.org/http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.4108&rep

=rep1&type=pdf.

Quarta, Davide, Marcello Pogliani, Mario Polino, Federico Maggi, Andrea Maria Zanchettin, and Stefano Zanero. 2017. "An Experimental Security Analysis of an Industrial Robot Controller." *2017 IEEE Symposium on Security and Privacy (SP)*, 268–86. https://doi.org/10.1109/SP.2017.20.

Rabatel, Julien, Sandra Bringay, and Pascal Poncelet. 2011. "Anomaly Detection in Monitoring Sensor Data for Preventive Maintenance" 38 (6): 7003–15. https://doi.org/10.1016/j.eswa.2010.12.014>.

Roesch, M. 1999a. "Snort: Lightweight Intrusion Detection for Networks." *LISA '99: 13th Systems Administration Conference*, 229–38. https://doi.org/http://portal.acm.org/citation.cfm?id=1039834.1039864.

———. 1999b. "Snort: Lightweight Intrusion Detection for Networks." *LISA '99: 13th Systems Administration Conference*, 229–38. https://doi.org/http://portal.acm.org/citation.cfm?id=1039834.1039864.

Roschke, Sebastian, Feng Cheng, and Christoph Meinel. 2011. "A New Alert Correlation Algorithm Based on Attack Graph." In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6694 LNCS:58–67. https://doi.org/10.1007/978-3-642-21323-6_8.

Salah, Saeed, Gabriel Maciá-Fernández, and Jesús E. Díaz-Verdejo. 2013. "A Model-Based Survey of Alert Correlation Techniques." *Computer Networks* 57 (5): 1289–1317. https://doi.org/10.1016/j.comnet.2012.10.022.

Schwab, Wolfgang, and Mathieu Poujol. 2018. "The State of Industrial Cybersecurity 2018."

Shen, Qingming, Jianmin Gao, and Cheng Li. 2010. "Automatic Classification of Weld Defects in

Radiographic Images." *Insight: Non-Destructive Testing and Condition Monitoring* 52 (3): 134–39. https://doi.org/10.1784/insi.2010.52.3.134.

Shittu, Riyanat, Alex Healing, Robert Ghanea-Hercock, Robin Bloomfield, and Muttukrishnan Rajarajan. 2015. "Intrusion Alert Prioritisation and Attack Detection Using Post-Correlation Analysis." *Computers and Security* 50: 1–15. https://doi.org/10.1016/j.cose.2014.12.003.

Simmons, Chris, Charles Ellis, Sajjan Shiva, Dipankar Dasgupta, and Qishi Wu. 2014. "AVOIDIT: A Cyber Attack Taxonomy." *9th Annual Symposium on Information Assurance*, 12–22. http://si.lopesgazzani.com.br/docentes/marcio/SegApp/CyberAttackTaxonomy_IEEE_Mag.pdf.

Siraj, Ambareen. 2006. "A Unified Alert Fusion Model for Intelligent Analysis of Sensor Data in an Intrusion Detection Environment." Mississippi State University. https://dl.acm.org/citation.cfm?id=1269240.

Smith, Reuben, Nathalie Japkowicz, Maxwell Dondo, and Peter Mason. 2008. "Using Unsupervised Learning for Network Alert Correlation." In *Conference of the Canadian Society for Computational Studies of Intelligence*, 308–19. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-540-68825-9_29.

Song, Chen, Feng Lin, Zongjie Ba, Kui Ren, Chi Zhou, and Wenyao Xu. 2016. "My Smartphone Knows What You Print : Exploring Smartphone-Based Side-Channel Attacks Against 3D Printers." In *ACM CCS*, 895–907. https://doi.org/10.1145/2976749.2978300.

Song, Zhengyi. 2018. "Sustainability Benefits Analysis of CyberManufacturing Systems." Doctoral thesis. Syracuse University. Syracuse, New York.

Song, Zhengyi, and Young Moon. 2016a. "Assessing Sustainability Benefits of Cybermanufacturing Systems." *International Journal of Advanced Manufacturing*

Technology 90, 5-8: 1365-1382. doi: 10.1007/s00170-016-9428-0

Song, Zhengyi, and Young B. Moon. 2016b. "Performance Analysis of CyberManufacturing Systems: A Simulation Study." *In the Proccedings of IFIP 13th International Conference on Product Lifecycle Management*, 592–605. Columbia, South Carolina: Springer. doi: 10.1007/978-3-319-54660-5_53

Spring, Tom. 2018. *Leaky Backup Spills 157 GB of Automaker Secrets*. *Threatpost*. https://threatpost.com/leaky-backup-spills-157-gb-of-automaker-secrets/134293/.

Sturm, Logan D., Christopher B. Williams, Jamie A. Camelio, Jules White, and Robert Parker. 2014. "Cyber-Physical Vulnerabilities in Additive Manufacturing Systems." In *International Solid Freeform Fabrication Symposium*, 951–63. http://sffsymposium.engr.utexas.edu/sites/default/files/2014-075-Sturm.pdf.

———. 2017a. "Cyber-Physical Vulnerabilities in Additive Manufacturing Systems: A Case Study Attack on the .STL File with Human Subjects." *Journal of Manufacturing Systems* 44: 154–64. https://doi.org/10.1016/j.jmsy.2017.05.007.

———. 2017b. "Cyber-Physical Vulnerabilities in Additive Manufacturing Systems: A Case Study Attack on the .STL File with Human Subjects." *Journal of Manufacturing Systems* 44: 154–64. https://doi.org/10.1016/j.jmsy.2017.05.007.

Tao, Fei, Jiangfeng Cheng, Qinglin Qi, Meng Zhang, He Zhang, and Fangyuan Sui. 2018. "Digital Twin-Driven Product Design, Manufacturing and Service with Big Data." *International Journal of Advanced Manufacturing Technology* 94 (9–12): 3563–76. https://doi.org/10.1007/s00170-017-0233-1.

Timofte, Jack. 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, Hamilton, Jules White, Jaime A Camelio, Christopher Williams, Brandon Amos, and Robert Parker. 2015. “Bad Parts: Are Our Manufacturing Systems at Risk of Silent Cyberattacks?” *IEEE Security and Privacy* 13 (3): 40–47. https://doi.org/10.1109/MSP.2015.60.

Valdes, Alfonso, and Keith Skinner. 2001. “Probabilistic Alert Correlation.” *Recent Advances in Intrusion Detection*, 54–68. https://doi.org/10.3923/jas.2007.565.569.

Valeur, Fredrik. 2006. “Dissertation: Real-Time Intrusion Detection Alert Correlation.” University of California, Santa Barbara. https://doi.org/10.1016/j.ejor.2004.04.029.

Valeur, Fredrik, Giovanni Vigna, Christopher Kruegel, and Richard A. Kemmerer. 2004. “A Comprehensive Approach to Intrusion Detection Alert Correlation.” *IEEE Transactions on Dependable and Secure Computing* 1 (3): 146–68. https://doi.org/10.1109/TDSC.2004.21.

Verizon. 2017. “2017 Data Breach Investigations Report.” *Verizon Enterprise*. https://doi.org/10.1017/CBO9781107415324.004.

Vijayaraghavan, A, W Sobel, Armando Fox, D Dornfeld, and P Warndorf. 2008. “Improving Machine Tool Interoperability Using Standardized Interface Protocols : MT Connect.” *2008 International Symposium and Flexible Automation,*. https://escholarship.org/uc/item/4zs976kx.

Vincent, Hannah, Lee Wells, Pablo Tarazaga, and Jaime Camelio. 2015. “Trojan Detection and Side-Channel Analyses for Cyber-Security in Cyber-Physical Manufacturing Systems.” *Procedia Manufacturing* 1: 77–85. https://doi.org/10.1016/j.promfg.2015.09.065.

Waslo, René, Tyler Lewis, Ramsey Hajj, and Robert Carton. 2017. “Industry 4.0 and Cybersecurity: Managing Risk in an Age of Connected Production.” *Deloitte University Press*, 1–21.

Wells, Lee J., Jaime A. Camelio, Christopher B. Williams, and Jules White. 2014. "Cyber-Physical Security Challenges in Manufacturing Systems." *Manufacturing Letters* 2 (1): 74–77. https://doi.org/10.1016/j.mfglet.2014.01.005.

WIRED. 2015. "Hackers Remotely Kill a Jeep on the Highway." 2015. https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

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 the *Proceedings of ASME International Mechanical Engineering Congress and Exposition, V014T07A004-V014T07A004*. Phoenix, Arizona: American Society of Mechanical Engineers. doi: 10.1115 /IMECE201667641

Wu, Mingtao, and Young B. Moon. 2017a. "DACDI (Define, Audit, Correlate, Disclose, and Improve) Framework to Address Cyber-Manufacturing Attacks and Intrusions." *Manufacturing Letters 15*, B: 155-159. doi: 10.1016/j.mfglet.2017.12.009

Wu, Mingtao, and Young B. Moon. 2017b. "Taxonomy of Cross-Domain Attacks on CyberManufacturing System." *In Complex Adaptive Systems Conference*, 368-374. Chicago, Illinois: Procedia Computer Science. doi: 10.1016/j.procs.2017.09.050

Wu, Mingtao, and Young B. Moon. 2018. "Taxonomy for Secure CyberManufacturing System." In *Proceedings of the ASME 2018 International Mechanical Engineering Congress and Exposition*, V002T02A067. Pittsburgh, PA: American Society of Mechanical Engineers. doi: 10.1115/IMECE2018-86091

Wu, Mingtao, Jinwoo Song, Long Wang, Lucas Lin, Noé Aurelle, Yapan Liu, Bingyan Ding, Zhengyi Song, and Young B Moon. 2018. "Establishment of Intrusion Detection Testbed for CyberManufacturing Systems." In *Proceedings of the 46th SME North American*

Manufacturing Research Conference, 1053-1064. College Station, Texas: Procedia Manufacturing. doi: 10.1016/j.promfg.2018.07.142

Wu, Mingtao, Zhengyi Song, and Young B. Moon. 2019. "Detecting Cyber-Physical Attacks in CyberManufacturing Systems with Machine Learning Methods." *Journal of Intelligent Manufacturing* 30 (3): 1111–23. https://doi.org/10.1007/s10845-017-1315-5.

Wu, Mingtao, Heguang Zhou, Longwang Lucas Lin, Bruno Silva, Zhengyi Song, Jackie Cheung, and Young Moon. 2017. "Detecting Attacks in CyberManufacturing Systems : Additive Manufacturing Example." In *Proceedings of the International Conference on Mechanical, Aeronautical and Automotive Engineering*, 108-06005. Malacca, Malaysia: EDP Sciences. doi: 10.1051/matecconf/201710806005

Wuest, Thorsten, Christopher Irgens, and Klaus-Dieter Thoben. 2014. "An Approach to Monitoring Quality in Manufacturing Using Supervised Machine Learning on Product State Data." *Journal of Intelligent Manufacturing* 25 (5): 1167–80. https://doi.org/10.1007/s10845-013-0761-y.

Yadegari, Babak, and Paul Mueller. 2012. "The Stuxnet Worm Overview of Stuxnet," 1–12. http://www.cs.arizona.edu/~collberg/Teaching/466-566/2012/Resources/presentations/2012/topic9-final/slides.pdf.

Yampolskiy, Mark, Peter Horvath, Xenofon D Koutsoukos, Yuan Xue, and Janos Sztipanovits. 2013. "Taxonomy for Description of Cross-Domain Attacks on CPS." In *Proceedings of the 2nd ACM International Conference on High Confidence Networked Systems*, 135–142. https://doi.org/10.1145/2461446.2461465.

Yampolskiy, Mark, Anthony Skjellum, Michael Kretzschmar, Ruel A. Overfelt, Kenneth R. Sloan, and Alec Yasinsac. 2016. "Using 3D Printers as Weapons." *International Journal of Critical*

Infrastructure Protection 14: 58–71. https://doi.org/10.1016/j.ijcip.2015.12.004.

Zeltmann, Steven Eric, Nikhil Gupta, Nektarios Georgios Tsoutsos, Michail Maniatakos, Jeyavijayan Rajendran, and Ramesh Karri. 2016a. "Manufacturing and Security Challenges in 3D Printing." *JOM*, May. https://doi.org/10.1007/s11837-016-1937-7.

———. 2016b. "Manufacturing and Security Challenges in 3D Printing." *Jom* 68 (7): 1872–81. https://doi.org/10.1007/s11837-016-1937-7.

Zhang, Lin, Yongliang Luo, Fei Tao, Bo Hu Li, Lei Ren, Xuesong Zhang, Hua Guo, Ying Cheng, Anrui Hu, and Yongkui Liu. 2014. "Cloud Manufacturing: A New Manufacturing Paradigm." *Enterprise Information Systems* 8:2 (February 2015): 167–87. https://doi.org/10.1080/17517575.2012.683812.

Zhang, Xue Wu, Yan Qiong Ding, Yan Yun Lv, Ai Ye Shi, and Rui Yu Liang. 2011. "A Vision Inspection System for the Surface Defects of Strongly Reflected Metal Based on Multi-Class SVM." *Expert Systems with Applications* 38 (5): 5930–39. https://doi.org/10.1016/j.eswa.2010.11.030.

Zhu, Bonnie, Anthony Joseph, Shankar Sastry. 2011. "A Taxonomy of Cyber Attacks on SCADA System." *Internet of Things (IThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing*.

VITA

Mingtao Wu
Phone: (315) 380-8240 | Email: miwu@syr.edu

EDUCATION

Syracuse University, Syracuse, NY **Aug 2015 – Aug 2019**
Ph.D., *Mechanical Engineering*
Dissertation: "*Intrusion Detection of Cyber-Physical Attacks in Cyber-Manufacturing System*"
Advisor: Dr. Young B. Moon
Committee: Dr. J. Park (Chair), Dr. W. Du, Dr. J. L. Romeu, Dr. J. Dannenhoffer, Dr. X. Liu.
Syracuse University, Syracuse, NY **Aug 2013 – May 2015**
M.S., *Mechanical and Aerospace Engineering*
Beijing Forestry University, China **Aug 2009 – Jun 2013**
B.S., *Automotive Engineering*

PUBLICATIONS

- Peer-reviewed Journal

Wu, M., Song, J., Di, B. He, & Moon, Y. B. (Submitted). Development of Testbed for Cyber-Manufacturing Security Issues. *International Journal of Computer Integrated Manufacturing*.

Wu, M., & Moon, Y. (2019). Alert Correlation for Detecting Cyber-Manufacturing Attacks. *Journal of computing and information science in engineering*.

Wu, M., & Moon, Y. (2019). Intrusion Detection System for Cyber-Manufacturing System. *Journal of Manufacturing Science and Engineering*, 141(3), 031007. doi: 10.1115/1.4042053

Wu, M., Song, Z., & Moon, Y. B. (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

Wu, M., & Moon, Y. (2018). DACDI (Define, Audit, Correlate, Disclose, and Improve) framework to address cyber-manufacturing attacks and intrusions. *Manufacturing Letters*, 15, 155-159. doi:10.1016/j.mfglet.2017.12.009

- Conference Proceedings

Wu, M., & Moon, Y. (2019). Intrusion Detection of Cyber-Physical Attacks in Manufacturing Systems: A Review. *2019 International Mechanical Engineering Congress and Exposition*, Salt Lake City, Utah: ASME.

Song, J., **Wu, M.,** & Moon, Y. (2019). Physical Data Auditing for Attack Detection in Cyber-Manufacturing Systems: Blockchain for Machine Learning Process. *2019 International Mechanical Engineering Congress and Exposition*, Salt Lake City, Utah: ASME.

Wu, M., & Moon, Y. (2019). Alert Correlation for Cyber-Manufacturing Intrusion Detection. *North American Manufacturing Research Conference (NAMRC) 47 proceedings*, Erie, Pennsylvania: SME.

Wu, M., & Moon, Y. (2018). Taxonomy for secure CyberManufacturing System. *In Proceedings of the 2018 International Mechanical Engineering Congress and Exposition*, V002T02A067, Pittsburgh, Pennsylvania: ASME. doi: 10.1115/IMECE2018-86091

Wu, M., Song, J., Lin, L. W., Aurelle, N., Liu, Y., Ding, B., . . . Moon, Y. B. (2018). Establishment of intrusion detection testbed for CyberManufacturing systems. *Procedia Manufacturing*, 26, 1053-1064. doi:10.1016/j.promfg.2018.07.142

Wu, M., & Moon, Y. B. (2017). Taxonomy of cross-domain attacks on CyberManufacturing system. *Procedia Computer Science*, 114, 367-374. doi:10.1016/j.procs.2017.09.050

Wu, M., Zhou, H., Lin, L. L., Silva, B., Song, Z., Cheung, J., & Moon, Y. (2017). Detecting attacks in CyberManufacturing Systems: additive manufacturing example. *In Proceedings of the 2017 International Conference on Mechanical, Aeronautical and Automotive Engineering*, 108(06005), Malacca, Malaysia: EDP Sciences. doi: 10.1051/matecconf/201710806005

Wu, M., Phoha, V. V., Moon, Y. B., & Belman, A. K. (2016). Detecting malicious defects in 3D printing process using machine learning and image classification. *In Proceedings of the 2016 International Mechanical Engineering Congress and Exposition*, V014T07A004, Phoenix, Arizona: ASME. doi: 10.1115/IMECE2016-67641

Moon, Y., & **Wu, M.** (2016). Spurring innovation in a sustainable manufacturing course. *In Proceedings of the 2016 ASEE Annual Conference and Exposition*, New Orleans, Louisiana: ASEE. doi: 10.18260/p.25861

Moon, Y. B., & **Wu, M.** (2015). Innovation within the constraints of sustainability: Analysis of product development projects. *In Proceedings of the 2015 IEEE Frontiers in Education Conference*, 15652747, El Paso, Texas: IEEE. doi: 10.1109/FIE.2015.7344083

AWARDS & HONORS

- Syracuse University Summer PhD Dissertation Fellowship (2019)
- Syracuse University ECS-MAE Research Day Poster Competition Award (2018 & 19)
- NSF Travel Grant for NAMRC 47 Conference (2018 & 19)
- Syracuse University GSO Travel Grant (2016 & 18)
- SPARK Innovation Competition Second Prize (2015)
- Demo Day Student Entrepreneurship Competition 1st Prize (2014)
- Invention and Creativity Competition Most Creative Award (2014)
- Invention and Creativity Competition 2nd Funding Award (2014)

RESEARCH EXPERIENCE

Syracuse University, Syracuse, NY **PhD research (2018-19)**

Intrusion Detection for cyber-physical attacks in Cyber-Manufacturing System (CMS)

- Utilized machine learning (classification, anomaly detection) into CMS physical domain data to achieve detection on the cyber-physical attack with physical payloads, with accuracy over 95%.
- Defined similarity-based alert correlation method for cyber-physical attack, reached alert reduction rate of 99.1%, realize rapid respond and root cause analysis.

Establish a Cyber-Manufacturing System Testbed **PhD research (2017-18)**

- Designed and developed a CMS testbed for intrusion detection research, including a 3D printer, CNC milling machine, conveyor, robotic arms, web frontend, database, and sensors.
- Simulated cyber-physical attack on CMS Testbed with systemic attack taxonomies.
- Collected and processed data for intrusion detection and alert correlation theory validation.
- Maintained testbed lab for experiment, course demonstration, and external visiting.

Detecting cyber-physical attacks in manufacturing process **PhD research (2015-16)**

- Analyzed cyber-physical attacks in 3D printing "STL" file and CNC milling "Gcode".
- Adapted random forest, Naïve Bayes, anomaly detection machine learning algorithms to detect physical consequence after attack, realized accuracy over 91%.
- Utilized both computer simulation and physical experiment for theory validation.

Taxonomy of cyber-physical attacks **PhD research (2015-16)**

- Created a cross-domain attack taxonomy from detection perspective with four dimensions: attack vector, impact, target, and consequence
- Created a cyber-physical attack taxonomy from attack perspective with six dimensions: human, product, equipment, intellectual property, environment, operation;

- Enhanced understanding for cyber-physical attack, clarified vulnerabilities in CMS.

Innovation course with sustainability constraint in engineering education**Graduate research (2014)**

- Participated a sustainable manufacturing course as TA for engineering education research
- Created rubrics to evaluate openness and courage to explore Ideas for students in the class.
- Analyzed five years of data to evaluate the effect of constraints on students' creativity.

TEACHING EXPERIENCE

Production System Design and Control, Graduate Level **Teaching assistant (2018)**

- Conceived lecture materials and lectured on forecasting methods in production system.

Data Analysis for Engineers, Undergraduate Level **Teaching assistant (2017-18)**

- Developed and led lab exercises on R programming, topics included: R basics, vectors, R markdown, discrete and continuous distribution, expectation, permutation, and combination.
- Proposed and graded homework, quiz and exam materials.

Simulation and Data Analytics, Graduate Level **Teaching assistant (2017-19)**

- Lectured on topics including: Arena Simulation Software, Design of Experiment, R language.
- Generated and led lab exercises on Arena modeling, Input analysis, and Process Analyzer.
- Created and led lab materials on R language: DoE, ANOVA, and Fractional factorial design.
- Proposed and graded homework, quiz and exam materials.

Productivity and Quality Engineering, Graduate Level **Teaching assistant (2016-17)**

- Co-supervised graduate students on group projects about quality control.
- Trained students on quality engineering software: Minitab and Quality Companion.
- Evaluated assignments on each lesson and lab.

Invention Factory, Undergraduate Level **Teaching assistant (2017)**

- Mentored student groups ideation, design and conceive an original invention in 6 weeks.
- Instructed students filing provisional patent applications for their invention at the end of class.
- Coordinated among faculty, students, and guest evaluators.

Introduction to ECS, Undergraduate Level **Teaching assistant (2016-17)**

- Led undergraduate group project based on LEGO Mindstorms RCX robotic package.

Advanced Mechanics of Materials, Graduate Level **Teaching assistant (2015-16)**

- Evaluated assignments, exams, and quiz.

Sustainable Manufacturing, Graduate Level **Teaching assistant (2015)**

- Lectured on sustainable product development and entrepreneurship.
- Guided student projects on sustainable product development and evaluated assignments.
- Assisted research in engineering education on students' innovation and creativity.

STUDENT ADVISING EXPERIENCE

Advised 17 undergrad/graduate students from US, Brazil, France, China, India, Korea. Participated advising students from program including Independent Research, Research Experience Undergrad, Exchange Program, and Undergrad Leadership Program.

- 2016 Summer: Lucas Lin, Jackie Cheung, Heguang Zhou, Bruno Silva.
- 2017 Summer: Noé Aurelle, Yapan Liu, Bingyan Ding, Jinwoo Song, Lucas Lin.
- 2018 Spring: Jinwoo Song, Jupeng Di, Benliu He, Ziming Wang, Jingkai Zhang.
- 2018 Summer: Emily Greaney, Claire Baron, Côme Butin, Merouane Alliouche, Snehav Sharma.