The Ohio State Model of ICS Cybersecurity

Theodore Allen, John McCarty, Tu Feng, Shih-Hsien Tseng,
Vimal Buck and Robert Pardee

December 7, 2021

The Ohio State Model For ICS Cybersecurity

Theodore T. Allen
Integrated Systems Engineering
The Ohio State University
Columbus, Ohio
ORCID: 0000-0002-9522-3252

John McCarty
Information Security
The Ohio Technology Consortium
Columbus, Ohio USA
ORCID: 0000-0002-2031-9815

Tu Feng
Integrated Systems Engineering
The Ohio State University
Columbus, Ohio
feng.1039@buckeyemail.osu.edu

Shih-Hsien Tseng
Department of Industrial Management
National Taiwan University of Science & Technology
Taipei, Taiwan
shtseng@mail.ntust.edu.tw

Vimal Buck
The Center For Design & Manufacturing Excellence
The Ohio State University
Columbus, Ohio
buck.80@osu.edu

Robert Pardee
Enterprise Security
The Ohio State University
Columbus, Ohio
pardee.10@osu.edu

***Abstract*—We propose a simple framework for Industrial Control System (ICS) system cybersecurity. The proposed system is based on considerations which include known vulnerabilities, safety issues, and the centrality of assets in hypothetical attack vectors. We relate the proposed system to the Purdue Model and two optimization formulations from the literature. We also relate our point system to the results of a recent penetration testing exercise on a manufacturing robotic cell. Finally, we discuss multiple challenges including that posed by legacy equipment and threats to manufacturing uptime.**

Keywords—Purdue Model, manufacturing, ICS410, vulnerabilities

I. Introduction

Manufacturing is the second most attacked industrial sector (after finance and insurance). The number of vulnerabilities affecting Industrial Control System (ICS) jumped significantly in 2014 and has been increasing sharply [1]. More generally, our buildings and street signs are battlegrounds with smart thermostats offering access points with potential implications for comfort and safety and even electric grid stability. The original Purdue Model\ was proposed to clarify the layers in a modern Industrial Control System (ICS) [2]. The layers shown in Fig. 1 relate the concepts of Information Technology (IT) and Operational Technology (OT), which are still potentially relevant phrases. Yet, the concept of a simple air gap or strong firewall is increasingly outmoded.

Operational Technology (OT) is acting more like IT and increasingly demands access and connectivity. The firm Dragos estimates that only a minority (~30%) of manufacturers claimed to use air gaps in 2020 [3]. At the same time, the ease of micro-segmentation within and between layers is increasing. The concept of zero trust would seem to imply effective segmentation around every device within a network [4]. Yet, such an extreme set of limitations brings cost and complication. Such segmentation can also be impossible with legacy equipment. The question that we study here is how to support segmentation decision-making to achieve an appropriate level of risk and trust. The contribution is an application framework that seeks to balance direct costs, simplicity, and the chances and severity of plausible intrusions.

II. Review of Modeling Frameworks

In this section, selected methods from operations research and reliability engineering are reviewed. These methods relate cybersecurity with general system maintenance and economic and other consequences [4]. We begin by elaborating on some of the challenges of Fig. 1 and the Purdue Model. Next, we consider an integer programming adaptation of a proposed vulnerability or threat coverage model [5]. Also, we describe the possible adaptation of a reinforcement learning model [6]. Reinforcement learning offers the promise of addressing incomplete information but brings significant complications.

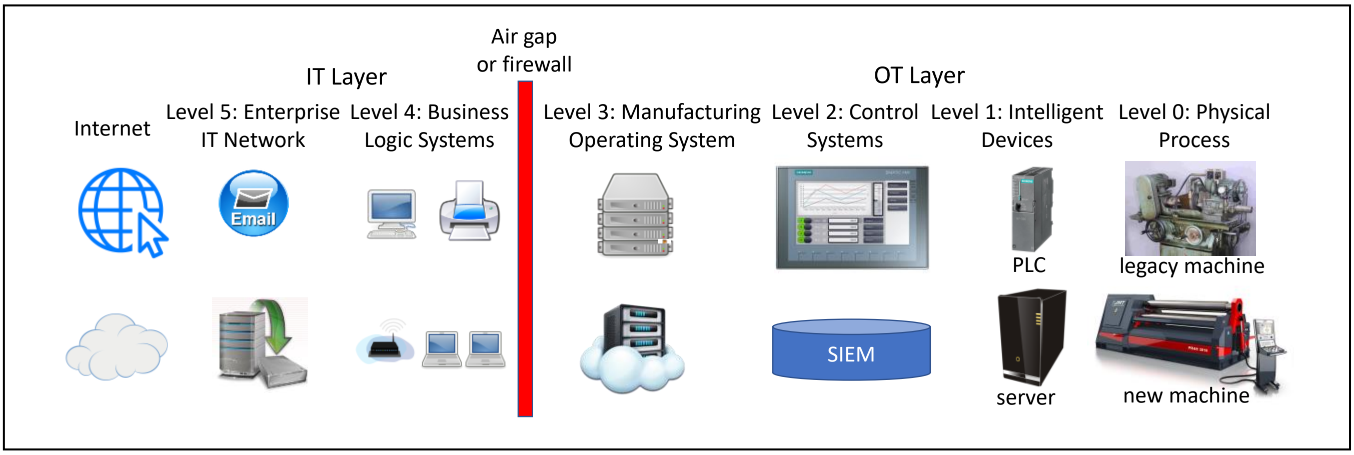

Air gap
or firewall
IT Layer
OT Layer
Internet
Level 5: Enterprise IT Network
Level 4: Business Logic Systems
Level 3: Manufacturing Operating System
Level 2: Control Systems
Level 1: Intelligent Devices
Level 0: Physical Process
Email
PLC
legacy machine
SIEM
server
new machine

Fig. 1. The layers in the Purdue model with hypothetical devices.

A. *Model-Based Bayesian Reinforcement Learning*

Although the Purdue Model was not intended as a cybersecurity framework, it has created a language and set of widely recognized concepts (Fig. 1). The discussion of this model helps in clarifying the changing ICS cybersecurity landscape. At the same time, there are many factors that are contributing to making the Purdue Model misleading or obsolete. To elaborate:

- Only approximately 30% of manufacturers claim to use an **air gap** in 2021 [3]. The need for sharing information is apparently too strategically relevant to support an air gap.
- The concept of "**zero trust**" means that a single gap or wall is no longer relevant. The connection between any two nodes (i.e., an "arc") offers the potential for an effective firewall that may be implemented in software, requiring no physical division [7]. This paper relates to decisions within a zero trust environment.
- While some segmentation or effective firewalls have been enabled by modern systems, some new and **legacy devices** may be impossible to isolate because of hardware or other issues [8]. These offer constraints in any security design.
- Modern systems offer options in addition to simple boundaries. For example, **unidirectional** gateways between the OT and IT networks or any two devices are possible.
- Implementing **multifactor authentication** for Virtual Private Networks is a potentially important step for all jump servers and potentially all remote access points. It represents another set of options for each arc [9].
- **Penetration tests** that are scheduled regularly can be critical to achieving reasonable levels of safety. The costs of such activities may need to be balanced with other potential costs.
- There are many other opportunities for system hardening including: building a Security Information and Event Management (**SIEM**), Intrusion Detection System (IDS), and deploying **honeypots** for monitoring, confusing attackers, and/or understanding intruder intentions [10].

Creating incentives for these hardening activities and their maintenance is a critical challenge which we all face in our homes and the systems for which we have authority.

B. *Prioritizing Vulnerability Repairs*

An integer programming model was proposed to optimize maintenance and other investment activities in a generic environment with vulnerabilities. Here, we review the proposed model and describe its potential adaptation to support ICS system decision-making [5].

Let *S* denote the set of attack paths and let *N* denote the set of devices with vulnerabilities (nodes). These paths could be enumerated in table top or penetration testing activities. Yet, one can simply imagine a kill chain reaching from the higher layers toward the physical layer in Fig. 1. Let $n \in N$ denote the subset of vulnerabilities (real devices with vulnerabilities on them) in attack path $s \in S$: Let $c_n \in \Re_+$ represent the criticality of vulnerability $n \in N$. In our own work, we have identified what we call "supercritical vulnerabilities" in that they are many times more likely to be compromised than a median Common Vulnerability Scoring System (CVSS) critical. In general, a higher weight c_n corresponds to a more important vulnerability. We use these critical vulnerability levels to quantify the coverage of each attack path. Let *M* denote the set of available mitigations, and let $M_n \in M$ denote the set of mitigations that cover vulnerability node $n \in N$: Let $B \in \Re_+$ be the total mitigation budget. Each mitigation $m \in M$ has an implementation cost $b_m \in B$: In many situations, much of this underlying data may be collected from subject matter experts (SMEs) who have knowledge of the underlying processes, possible mitigations, and mitigation effectiveness.

Let variable x_m have a value of one if mitigation $m \in M$ is chosen, and zero otherwise. Let z_n have a value of one if node $n \in$ N is covered by a mitigation, and zero otherwise. Let y_s be the number of vulnerability nodes in attack path $s \in S$ that are covered, weighted by criticality level. Specifically, $y_s = \sum_{n \in N} c_n z_n$. The authors [5] introduced functions $f_s(y_s)$ that capture the coverage of attack path *s*. By assumption, $f_s(y_s)$ is nondecreasing and concave in y_s for each $s \in S$: Note that $f_s(y_s)$ might not be identical across all attack paths $s \in S$; since it can reflect the likelihood of an attack occurring and the perceived consequence of the attack. A decision problem is formulated as a mixed-integer programming model:

$$\begin{aligned} &\max_{x,y,z} \textstyle\sum_{s \in S} f_s(y_s). \qquad (1) \\ &\quad s.t. \textstyle\sum_{m \in M} b_m x_m \leq B, m \in M, \\ &\quad y_s = \textstyle\sum_{n \in N} c_n z_n, s \in S, \\ &\quad z_n \leq \textstyle\sum_{m \in M} x_m, n \in N, \\ &\quad x_n \in \{0,1\}, m \in M, \\ &\quad z_n \in \{0,1\}, n \in N. \end{aligned}$$

The objective of covering attack paths in part 1a might be revisited in the context of designing optimal segmentation. Other objectives might relate directly to costs incurred through maintenance or intrusions.

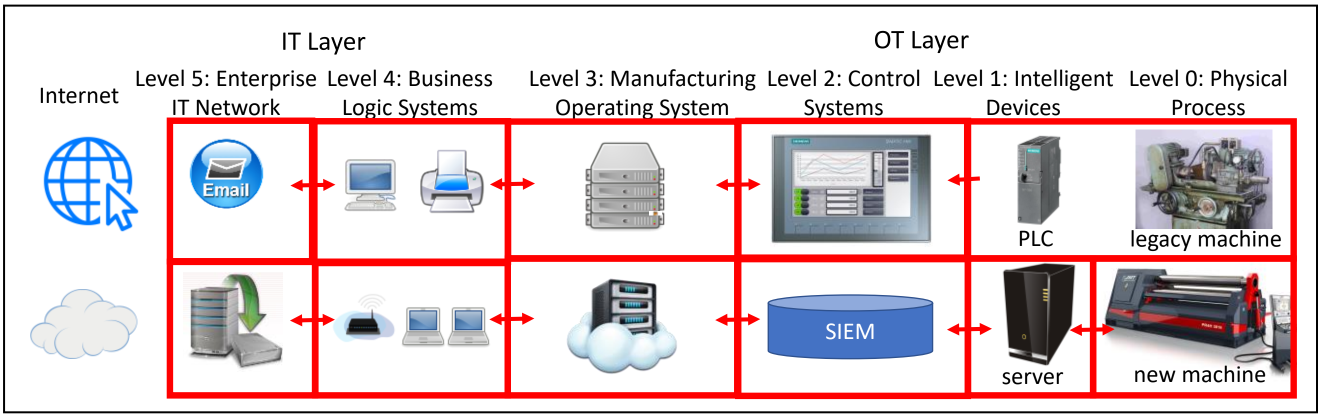
Further, the focus on the nodes and not the connections or "arcs" between entities is likely as or more relevant for ICS security than vulnerabilities or hosts as indicated in Fig. 2. Each boundary can allow travel one way or two ways as indicated in the figure. Also, legacy equipment such as on the right-hand side of Fig. 2 may simply not support detailed segmentation. The model in Equation (1) was extended to address uncertainties in remediation and two time stages [5]. Yet, another important consideration relates to learning the parameters in the model.

C. *Learning Models For Cybersecurity*

In our own vulnerability modeling, we considered the possibility that the parameters and transitions are unknown but can be represented by discrete model scenarios [6]. Then, as time evolves and actions are taken, the probabilities converge toward certainties about which scenario applies. We also considered the possibility of learning from multiple systems at-a-time. Yet, the readiness of these approaches for large-scale ICS decision support is limited both by conceptual complexity and computational issues at present. This motivates in part the exploration of a simple point-based system.

D. *Timely Modeling*

Besides the uncertainties of the success of remediation actions and about the unknown model parameters, there is also the possibility that situations can change. New devices can be added to the network and vulnerabilities can be discovered.


IT Layer
OT Layer
Internet
Level 5: Enterprise IT Network
Level 4: Business Logic Systems
Level 3: Manufacturing Operating System
Level 2: Control Systems
Level 1: Intelligent Devices
Level 0: Physical Process
Email
PLC
legacy machine
SIEM
server
new machine

Fig, 2. Hypotetical system with (nearly) full segmentation and decisions about allowable information flows built into firewall rules.

Even known vulnerabilities can have their severities changed as exploits are discovered with growing capabilities. In our own work, we explored how social media analytics and other Open Source Intelligence (OSINT) can play a role. The system risk levels and cost estimates can be updated [11]. Here, we simply assume that the key stakeholders are doing some monitoring. With a simple point-based system, manual point adjustments are enabled because all participants understand the implications on an intuitive level.

E. *Point Systems In Cybersecurity*

The concept of a point system is simple. Entities with large numbers of points are treated differently. For example, the ProofPoint commercial spam filter gives points to users who click on many phishing emails. For these users with high point values, the filtering settings are adjusted to change the false positive and false negative rates accordingly. Similarly, the Common Vulnerability Scoring System values for vulnerabilities range from 1 to 10. In our own work, we have explored how the scores can function as demerits with implications for system monitoring [12] and control [13].

III. Description of an ICS Penetration Test

Our team observed a penetration test of an ICS system. This system involved a robotic arm and commands generated in the Robot Operating System (ROS) language. In some respects, the system being studied resembles Fig. 1 except that there were no air gaps and the firewall was a router with some degree of vulnerable firmware. Also, some holes had been intentionally punched in the effective firewall to permit working at home. Despite some promising leads for penetrating the router, the team failed.

The way penetration was achieved was in-person entry into the (supposedly) secure laboratory. There was no segmentation within layers 0-3, i.e., inside the router. Further, there were operational ethernet ports within the network. Simply plugging into those ports, the team was able to observe the packet traffic to the legacy robot from a desktop computer using Wireshark. Using TcpRelay, the penetration team was able to shut down legitimate commands and replace these commands with alternative controlling inputs. Because of limited knowledge of ROS, the replaced commands only amounted to a shutdown order. With superior ROS skills, the team is confident that a successful replay attack could achieve arbitrary control.

IV. A Proposed Simple Point System

In this section, we propose a point system. The intent is to approximately capture the risks similar to the formulation in Equation [1]. At the same time, we want a system that is adjustable to address not merely the centrality of the nodes but also uncertainties about costs (learning) and the fact that threats change over time (timeliness).

A. *Subjective Considerations and Point Values*

There are many aspects that define the importance of connections or “arcs” in ICS networks. If a device on either side of the connection has a “supercritical” vulnerability (defined in our on-going research using Artificial Intelligence and extreme gradient boost tree modeling), for example, the connection in question is assigned 5 points. Similarly, if a device is associated with the potential for severe physical impacts such as harming people, we suggest adding 5 points. Further, if a device on either side is associated with a known intrusion, we propose to add 3 points. Similarly, 1 point is added for known critical vulnerabilities.

Some factors can also be considered which subtract points. These factors either balance or mitigate risks, suggesting that a more open and less restricted system is appropriate. These include high observed costs, routine penetration testing, and the remediation of known vulnerabilities.

TABLE I. Proposed Point System

#	Table Column Head		
	Table column subhead	***Points***	***Range***
1	Known active intrusion on subsystem.	+10	(0,10)
2	Super-critical vuln. is known on subsys.	+5	(0,5)
3	Physically dangerous or business critical.	+5	(0,5)
4	Recent intrusion is known on on subsys.	+3	(0,5)
5	Important functions are on subsys.	+2	(0,3)
6	Critical vuln. Is known on subsys.	+1	(0,2)
7	Costs are considered too high on install.	-1	(-2,0)
8	Critical vuln. is patched or remed.	-1	(-2,0)
9	Penetration testing issues are resolved.	-2	(-3,0)
10	Super-critical vuln. is patched or remed.	-5	(-3,0)

B. *Possible Implications of The Point Values*

Intuitively, higher point values should necessitate stronger enforcement of system boundaries. With some arbitrariness, we propose that 10 or more points on an arc should cause a complete restriction similar or identical to an air gap. In the range of 6 to 9 points it should permit one way travel towards the internet. Also, such access might generally benefit from a multifactor authentication requirement and the use of ICS aware firewalls. This follows because instructions traveling away from the internet could cause physical harm. Arcs with point values less than 5 might be unrestricted or permit two-way communication at the discretion of the system administrator or other key stakeholders.

With newer systems, these controls could hypothetically be implemented as part of the normal change management process as point values are adjusted through software-defined firewalls. More commonly perhaps, the system could involve significant setup and maintenance costs. In this case, the points could be used to update your treat model and then be implemented in the next mantainence cycle With this in mind, some adjustability of point scores to take cost issues into account may be relevant and appropriate.

V. A Toy Example Sub-Network

To illustrate how this system might work, consider the sub-network indicated in Fig. 3. The dashboard and SIEM control software are up-to-date and well-maintained. They require access to the internet with pinholes in firewalls for their own updating processes (still avoiding even outbound direct IP connections if proxied access is possible). No immediate segmentation is indicated for their left-hand connections or arcs. The legacy software on the right-hand side cannot be segmented from its connecting PLC because of its inherent limitations. Yet, the combination of PLC and legacy machines has multiple vulnerabilities and poses physical dangers. Complete disconnection or isolation in this case is warranted. The more modern machine at the bottom poses less risk. The device also supports a firewall. The system administrator judges that a two-way connection is appropriate with the server with some regulation from a software-based firewall. There is no two-factor authentication selected at this time.

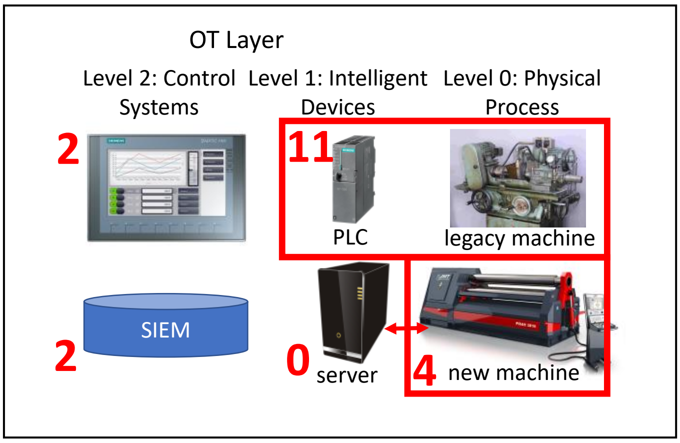


Fig. 1. Example of a figure caption. (*figure caption*)

Discussion and Future Work

We have reviewed the modern challenges of ICS cybersecurity and related decision problems. The Purdue Model is still helpful in creating a language to discuss ICS layers. Yet, the concept of a single air gap is largely obsolete. In its place, we have offered a simple point system with intuitive thresholds regulating device accessibilities.

A number of topics remain for future research. First, variants of the model in Equation (1) can be developed together with realistic cost estimates and applied to real world systems. The benefits of such applications can be measured including in terms of setup and maintenance costs and counts of any intrusions. Second, feedback on the point system can be obtained from a variety of stakeholders. A version 2.0 can be developed and explored using simulated and real world test networks. Third, real world Industrial 4.0 capabilities can be described in detail while clarifying related cybersecurity challenges also in greater detail. The initial penetration test described here is only a beginning example. Fourth, the framework may be regarded as “tentative” pending applications and quantitative analyses which can be performed. Fifth, the concept of measuring risk at the arcs can be related to data flow diagramming and associated threat modeling frameworks like STRIDE or PASTA. The development of a comprehensive threat modeling methodology that seeks to support the Ohio State Model point value estimation can be explored.

Acknowledgments

Michael Haas, Sayak Roychowdhury, and Matt Stvartak provided valuable inputs. Julia Armstrong, Anish Arora, Mark Bartman, Eylem Ekici, and Duane Wegener provided important encouragement and leadership. Satish Bukkapatnam, Ranjana Mehta, Prabhakar Pagilla, and Cathy Xia collaborated on related work. Christopher Lacy and Zhiqiang Lin offered helpful suggestions. We thank Cheri Rice and the Ohio Department of Higher Education grants AWD-108590 and AWD-109559 for partially supporting this work. We also thank Dean Schneider and SecureAmerica for grant AWD-109371, which partially supported this work.

References

[1] Bakuei, M., et al. "Securing smart factories threats to manufacturing environments in the era of Industry 4.0." Trend Micro Res 41 (2019).

[2] Williams, T. J. (1994). The Purdue Enterprise Reference Architecture. *Computers in Industry*, *24*(2-3), 141–158

[3] Dragos. (2021). ICS Cybersecurity Year in Review 2020. Hanover, MD; Dragos.

[4] Allen, T. T. (2019). *Introduction to engineering statistics and lean six sigma: statistical quality control and design of experiments and systems* (Vol. 3). London: Springer.

[5] Zheng, K., Albert, L. A., Luedtke, J. R., & Towle, E. (2019). A budgeted maximum multiple coverage model for cybersecurity planning and management. IISE Transactions, 51(12), 1303-1317.

[6] Allen, T. T., Roychowdhury, S., & Liu, E. (2018). Reward-based Monte Carlo-Bayesian reinforcement learning for cyber preventive maintenance. *Computers & Industrial Engineering*, 126, 578-594.

[7] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). “Zero trust architecture.” *Special Publication (NIST SP) - 800-207*.

[8] McLaughlin, Stephen, et al. "The cybersecurity landscape in industrial control systems." Proceedings of the IEEE 104.5 (2016): 1039-1057.

[9] Mathezer, S. (2021) Introduction to ICS Security Part 2, https://www.sans.org/blog/introduction-to-ics-security-part-2/ Accessed 10-31-21.

[10] Byres, Eric, and Justin Lowe. "The myths and facts behind cyber security risks for industrial control systems." Proceedings of the VDE Kongress. Vol. 116. 2004.

[11] Allen, T. T., Sui, Z., & Parker, N. L. (2017). “Timely decision analysis enabled by efficient social media modeling.” *Decision Analysis*, 14(4), 250-260.

[12] Afful-Dadzie, A. & Allen, T.T. (2014) “Data-driven cyber-vulnerability maintenance policies.” *Journal of Quality Technology*, 46(3), 234-250.

[13] Afful-Dadzie, A., & Allen, T. T. (2016). “Control charting methods for autocorrelated cyber vulnerability data.” *Quality Engineering*, 28(3), 313-328.