future internet

MDPI

Article

Leveraging Digital Twin Technology for Enhanced Cybersecurity in Cyber–Physical Production Systems

Yuning Jiang [1,*,†], Wei Wang [2], Jianguo Ding [3], Xin Lu [4] and Yanguo Jing [4]

[1] School of Computing, National University of Singapore, Singapore 639798, Singapore
[2] Department of Production and Automation Engineering, University of Skövde, 541 28 Skövde, Sweden; wei.wang@his.se
[3] Department of Computer Science, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden; jianguo.ding@bth.se
[4] Faculty of Business, Computing and Digital Industries, Leeds Trinity University, Leeds LS18 5HD, UK; x.lu@leedstrinity.ac.uk (X.L.); y.jing@leedstrinity.ac.uk (Y.J.)
* Correspondence: yuning_j@nus.edu.sg
† Part of the work by the contact author was done while at the University of Skövde, 541 28 Skövde, Sweden.

Abstract: The convergence of cyber and physical systems through cyber–physical systems (CPSs) has been integrated into cyber–physical production systems (CPPSs), leading to a paradigm shift toward intelligent manufacturing. Despite the transformative benefits that CPPS provides, its increased connectivity exposes manufacturers to cyber-attacks through exploitable vulnerabilities. This paper presents a novel approach to CPPS security protection by leveraging digital twin (DT) technology to develop a comprehensive security model. This model enhances asset visibility and supports prioritization in mitigating vulnerable components through DT-based virtual tuning, providing quantitative assessment results for effective mitigation. Our proposed DT security model also serves as an advanced simulation environment, facilitating the evaluation of CPPS vulnerabilities across diverse attack scenarios without disrupting physical operations. The practicality and effectiveness of our approach are illustrated through its application in a human–robot collaborative assembly system, demonstrating the potential of DT technology.

Keywords: cybersecurity; asset visibility; dependence analysis; mitigation prioritization; cyber–physical system (CPS); digital twin (DT); manufacturing system

Citation: Jiang, Y.; Wang, W.; Ding, J.; Lu, X.; Jing, Y. Leveraging Digital Twin Technology for Enhanced Cybersecurity in Cyber–Physical Production Systems. *Future Internet* **2024**, *16*, 134. https://doi.org/10.3390/fi16040134

Academic Editor: Gyu Myoung Lee

Received: 10 March 2024
Revised: 11 April 2024
Accepted: 13 April 2024
Published: 17 April 2024

1. Introduction

The integration of cyber and physical systems has profoundly transformed the manufacturing sector, leading to the development of cyber–physical production systems (CPPSs) and driving a shift toward intelligent manufacturing [1]. CPPSs, renowned for their adaptive capabilities to varying operational contexts, have yielded substantial enhancements in production processes, thereby augmenting efficiency and productivity [1]. Nonetheless, this integration also introduces increasing cybersecurity vulnerabilities, as underlined by incidents such as the "WannaCry" ransomware attack in 2017 [2]. Moreover, vulnerabilities often remain undetected until they are exploited in such environments. Even when vulnerabilities are identified and reported in databases like the common vulnerabilities and exposures (CVE), they may not be promptly addressed due to various operational constraints [3,4].

Cybersecurity challenges in information technology (IT) and smart manufacturing share some similarities but exhibit distinct approaches due to the unique nature of the systems they protect and the threats they encounter [5]. While IT security primarily focuses on safeguarding digital assets such as data and cloud services, smart manufacturing security extends its scope to protect operational technology (OT) systems, such as industrial control systems and physical machinery [6]. The convergence of IT and OT systems

presents significant complexities in asset management and dependence analysis, posing significant challenges to vulnerability mitigation [7,8]. A primary concern in IT security is the identification and prioritization of vulnerable assets, which is complicated by the sheer volume of components and their intricate interconnections. This challenge is further magnified in the OT domain, where maintaining continuous production often takes higher priority over asset scanning and vulnerability patching, as the latter may pose operational disruption risks [9,10]. This difficulty is heightened in IT and OT converged environments, like CPPSs, where gaining a comprehensive understanding is crucial for systematically assessing vulnerabilities [11,12]. Moreover, the threat landscapes faced by IT security and smart manufacturing security differ significantly. IT security contends with cyber threats, like malware, phishing, and data breaches, while smart manufacturing security faces unique challenges such as sabotage, physical tampering, and supply chain attacks. Consequently, IT security emphasizes data confidentiality, integrity, and compliance with regulations, while smart manufacturing prioritizes operational continuity, safety, and reliability [13].

Digital twin (DT) [14]-based methods emerge as a solution to address the gaps of asset management complexities [9,10] and vulnerable component patch prioritization [15], while protecting the monitoring of industrial assets [16,17]. DT methods, by functioning as virtual replicas of physical components, offer critical insights by aggregating asset-specific data and enabling analytics [18,19], or act as service providers through additive manufacturing [20]. Furthermore, they support enterprise security efforts by simulating attacks and evaluating potential impacts on virtual counterparts [21]. However, the adoption of DT security simulations within broader enterprise security frameworks, typically overseen by security operations centers, remains largely unexplored and underutilized [22].

This paper contributes to the field by presenting a DT-centered security framework tailored to enhance asset visibility and prioritize mitigation in CPPS. Our framework-oriented methodology consists of three key modules: a reference architecture that represents various unique CPPS assets; dependence rules within cyber–physical layers that facilitate component criticality analysis; and a virtual patch tuning and component vulnerability score calculation algorithm that enables patch prioritization. The framework enhances collaboration between the manufacturing and cybersecurity domains and bridges various organizational departments to ensure comprehensive monitoring and prediction of potential cybersecurity threats. Our approach leverages the capabilities of DT technology to support a simulation environment for vulnerability assessment across various attack scenarios without compromising the integrity of the physical system. Note that this paper follows a DT framework methodology as adopted in [16,23,24] to allow flexibility and modularity.

The practicality efficacy of our proposed framework is substantiated through its application in a human–robot collaborative (HRC) assembly system, illustrating how DT can strengthen the cybersecurity posture of CPPS. This study emphasizes the utility of DT in component criticality analysis, vulnerability retrieval, and attack simulation, thus positioning DT technology as a pivotal instrument in advancing cybersecurity measures within the manufacturing sector. Our methods also show a capability of being integrated with existing solutions in practical settings. Our contributions are summarized as follows:

- We present a flexible DT-centered framework that supports security assessment such as vulnerable component mitigation prioritization in CPPS without compromising operations.
- We identify critical assets through comprehensive dependence rules within the cyber–physical layers.
- We validate the framework's utility and effectiveness through an industrial case study involving an HRC assembly system, showcasing DT's potential to enhance CPPS cybersecurity.

The structure of this paper is organized as follows: Section 2 firstly introduces CPPSs, outlines the threats these systems encounter, and then presents related works while highlighting existing research gaps. Section 3 details the proposed DT design and the associated dependency rules. Section 4 delves into a case study that demonstrates the applicability

of the DT within an HRC system. The results of the case study are analyzed in Section 5. Finally, Section 6 concludes the paper, summarizing our findings and contributions.

2. Background and Related Works

The integration of information and communication technology highlights the importance of cybersecurity for manufacturing systems. This critical issue has received considerable attention from academics and industry, emphasizing the urgent need for strong cybersecurity measures in modern manufacturing environments [3,4,25].

2.1. Common Vulnerabilities in CPPSs

CPPSs encounter distinct security challenges that set them apart from traditional IT systems, stemming from their intricate networks and heterogeneous embedded components [26]. We conducted a static analysis of existing vulnerabilities affecting common CPPS assets (i.e., human–machine interface (HMI), programmable logic controller (PLC), remote terminal units (RTUs), and intelligent electronic devices (IEDs)). For this analysis, we primarily utilized two data sources: NVD, and Shodan. We then summarized the vulnerabilities commonly exploited in CPPSs, classifying them according to the common weakness enumeration (CWE) [27]:

- Software and firmware vulnerabilities: Flaws in application software and operating systems are prevalent yet challenging to mitigate. Vulnerabilities such as outdated firmware (e.g., CWE-1277) open back doors for attackers. Programming errors leading to buffer overflow can enable unauthorized code injection and elevated system access.
- Data communication security: The use of unencrypted protocols for data transmission risks, exposing sensitive information to unauthorized interception; vulnerable to man-in-the-middle (MiTM) attacks, as exemplified by the deployment of HTTP basic authentication for sensitive data (CWE-319).
- Access control issues: Inadequate access control mechanisms, such as improperly granting administrative permissions to guest accounts, can compromise critical system files. This encompasses flaws in identity management (e.g., CWE-1294), resource isolation (e.g., CWE-1189), authentication (e.g., CWE-261), and authorization (e.g., CWE-732). The integration of IoT devices introduces further hardware-targeted threats, including improper resource control (e.g., CWE-125 and CWE-787).
- Cybersecurity awareness and training: Insufficient cybersecurity training and awareness among employees can facilitate phishing attacks and internal breaches. Weak password policies (e.g., CWE-521) and communication gaps within organizations heighten the risk of data leaks and spoofing attacks.
- Cloud and edge computing vulnerabilities: Transitioning to cloud services brings forth vulnerabilities in edge computing and cloud architectures. Application programming interface (API) with insecure default configurations can inadvertently expose critical databases to the public internet, as highlighted by CWE-648, indicating the incorrect use of privileged APIs.

2.2. Advanced Persistent Threats in CPPS

The vulnerabilities detailed in the previous section can be sequentially exploited, leading to the formation of advanced persistent threats (APTs) [28].

Figure 1 illustrates the vulnerability chain within CPPS, where *V-x* denotes various vulnerabilities and *A-x* signifies stages of an advanced attack. An attacker could exploit the default password setting (*V-1*) of a design engineer's account, leading to account compromise (*A-1*). This breach could enable the attacker to target a related designer workstation in a follow-up attack (*A-2*). The likelihood of success for this follow-up attack (*A-2*) increases if the designer workstation suffers from poor authentication management (*V-2*). Furthermore, the attacker might launch an additional attack (*A-3*) to gain entry into a database server via the compromised workstation. This entry could be facilitated by weak access control (*V-3*), paving the way for another attack (*A-4*) aimed at altering geometry

computer-aided manufacturing (CAM) programs on the control server, especially if the database lacks encryption (*V-4*). The absence of effective communication between CAM engineers and security personnel (*V-5*) might allow such alterations to CAM program codes to remain unnoticed. Communication barriers between IT and OT staff, exacerbated by their physical separation, could result in the modified CAM file being used without correction. This oversight could lead to numerical control (NC) machines operating incorrectly, potentially causing significant disruptions in the manufacturing process due to these communication failures.

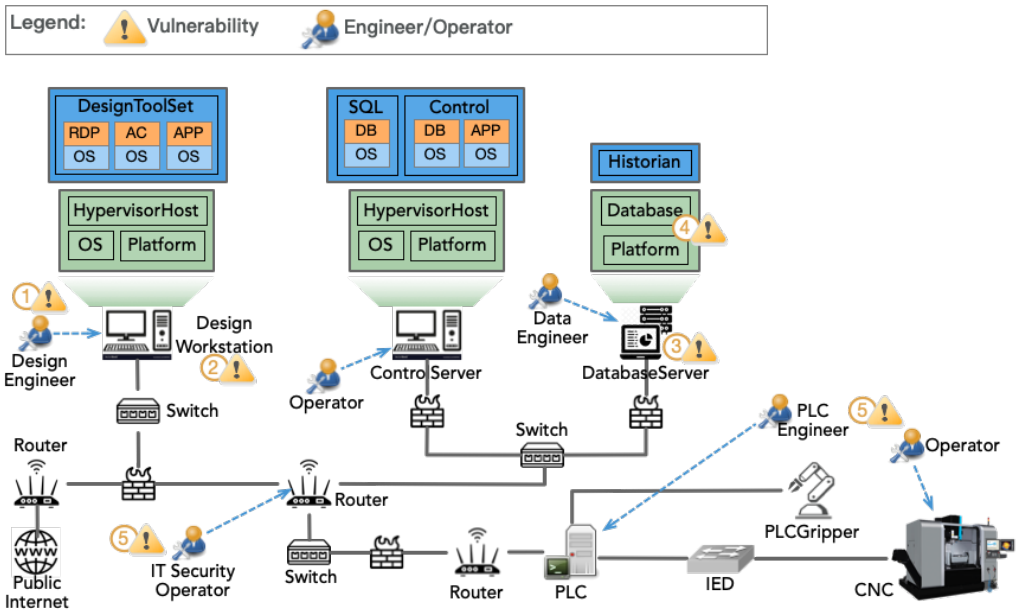

Figure 1. A scenario of advanced persistent threats in the manufacturing system. (Note that the numbers with circles indicate different vulnerabilities).

The chained vulnerabilities discussed can be pinpointed either through the insights of security experts or by tracing the interrelations among *CWE*, common attack pattern enumeration and classification (*CAPEC*) [29], and MITRE adversarial tactics, techniques, and common knowledge (*ATT&CK*) [30] attributes associated with specific vulnerabilities. The sequence of vulnerabilities illustrated in Figure 1 is effectively represented using these standardized enumerations, as depicted in Figure 2. Particularly, *CAPEC-560* is capable of advancing beyond both *CAPEC-49* and *CAPEC-70*, showcasing a complex attack graph that evolves along the vulnerability chain. Recognizing such sequences is crucial for conducting a comprehensive vulnerability assessment at the system level.

2.3. Cybersecurity Research on Manufacturing System

Research studies in this domain primarily focus on exploring potential attack scenarios, illustrating the tangible risks posed to manufacturing integrity and security. For instance, Wells et al. [31] demonstrated how NC files could be altered in an attack, leading to the production of defective parts. Similarly, Sturm et al. [32] examined the implications of tampering with STL files, which is crucial for additive manufacturing processes, while Desmit et al. [33] introduced a qualitative approach for assessing vulnerabilities in cyber–physical manufacturing systems, using compromised CAD files as a case study.

Moving beyond isolated incident analysis, some researchers have concentrated on developing comprehensive taxonomies of attacks within manufacturing systems. Elhabashy et al. [34] proposed an attack taxonomy centered around quality control systems, suggesting that securing these systems could mitigate a broad spectrum of potential attacks. Yampolskiy et al. [12] focused on creating an attack taxonomy for the additive manufacturing process. These contributions showcase the perspective of manufacturing engineers,

highlighting the operational impact of cyber security threats. However, from a cyber security engineering standpoint, these scenarios represent specific instances of broader attack methodologies, such as unauthorized database access leading to information tampering. This inconsistency underscores the need for a unified approach that enables both manufacturing and cyber security professionals to collaboratively address and mitigate cyber security challenges within manufacturing systems.

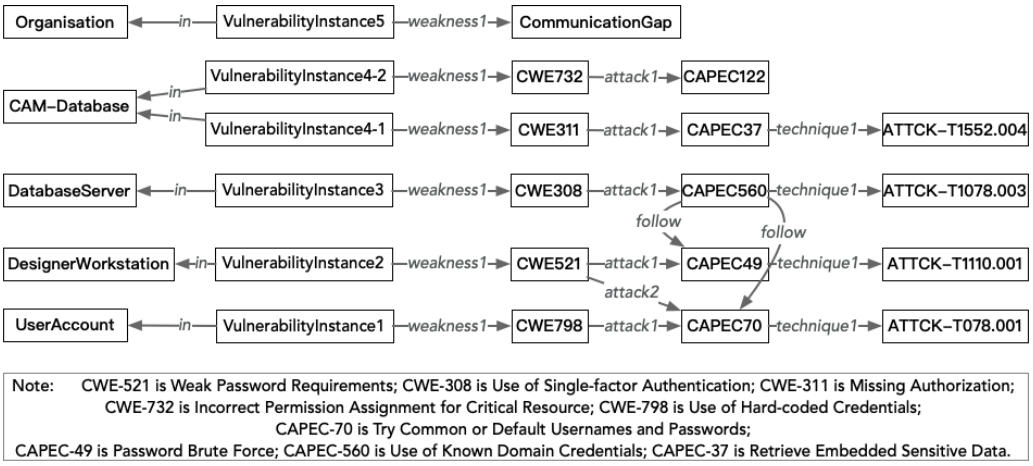

Figure 2. Example of vulnerability chains in CPPS.

Assessing the risk within manufacturing systems necessitates a comprehensive analysis of vulnerabilities that span the intricately linked IT and OT components. The mere identification of isolated vulnerabilities and threats falls short of addressing the multifaceted nature of contemporary systems [35]. Efforts to model these systems are aimed at identifying both discrete vulnerabilities, like outdated software, and systemic weaknesses, such as inadequate network segmentation. Techniques like tree structures, directed graphs, and logic diagrams have become prevalent for conducting overarching cybersecurity assessments or modeling potential exploits [36,37]. Nonetheless, many existing models, often tailored to specific system architectures or network setups, focus on assessing the probability or impact of particular vulnerabilities, including denial-of-service (DoS) attacks. These models typically lack flexibility and scalability and do not prioritize these aspects during their design phase [36]. Consequently, adapting the current cybersecurity assessment frameworks to new types of vulnerabilities often requires significant modifications, rendering them neither cost-effective nor efficient [38].

2.4. Digital Twin Applications in Cybersecurity

While DTs have yet to be extensively applied to cybersecurity challenges, existing research underscores their potential as an effective method [21,39]. Originating from the manufacturing sector, DTs offer a familiar framework for manufacturing engineers to address cybersecurity issues, while the insights generated by DT models provide actionable intelligence for cybersecurity engineers [40].

Eckhart and Ekelhart [41] pioneered the integration of DTs into information security, proposing a CPS twinning model that leverages standardized data formats like Automation ML for efficient simulation environment construction. This approach facilitates automatic acquisition of the data necessary for generating DT models, while also incorporating safety and security rules to detect potential intrusions by comparing commands between senders and receivers. Building on this foundation, subsequent work by Eckhart and Ekelhart [42] enhanced the model to include real-time data from physical systems, enabling accurate virtual mirroring and state transition monitoring. This extended model proved effective in detecting intrusions, including man-in-the-middle and insider attacks, demonstrating DT's potential as a robust platform for intrusion detection.

Lou et al. [43] further applied AML to model cyber–physical systems and conducted functional safety and cybersecurity analyses using DTs.

In the 'CyberFactory#1' project, Bécue et al. [44] explored DT's application in assessing production system responses to cyber-attacks and predicting potential damages, although detailed outcomes were not disclosed.

Bitton et al. [45] investigated the development of cost-effective, reliable, and security-oriented DT models, suggesting the value of creating purpose-specific multi-view DTs.

Suhail et al. [46] introduced the concept of gamification for DT security, adopting an offensive security stance. This innovative approach transforms DT into a versatile platform that not only facilitates a learning environment geared toward enhancing security awareness but also supports automated security evaluations and offers transparent DT assessments for security analysts. This is achieved by seamlessly incorporating machine learning technologies.

Additionally, DTs have been utilized beyond production systems, such as in safeguarding user privacy in smart automotive systems. Damjanovic-Behrendt [47] developed a DT model for smart cars to analyze operational, safety, and privacy data, employing data anonymization to mitigate privacy risks.

Although interactions across the physical, digital, and human domains are increasing, research exploring the application of DT technology across diverse architectural archetypes, particularly for managing numerous unique assets, remains limited [20]. Thus, developing a comprehensive reference architecture for leveraging DT to support smart manufacturing is necessary. Toward this direction, Sellitto et al. [48] redefined their enterprise architecture approach to depict a cooperative intelligent transport system scenario, evolving it into a threat-focused DT. This innovative shift was guided by the reference architecture model for Industry 4.0 (RAMI 4.0), facilitating a comprehensive depiction of the system's lifecycle.

Lu et al. [49] introduced a DT-based reference model that incorporates an information framework to depict the physical specifications and a data processing module to generate real-time representations of physical objects.

Balta et al. [16] proposed a framework-oriented DT architecture to support cyber-attack detection in CPPS. Additionally, an experimental case study is conducted on off-the-shelf 3D printers to illustrate the effectiveness of the proposed DT framework in detecting cyber-attacks.

Nevertheless, there is a limited effort in integrating DTs into cybersecurity assessments, especially in the area of mitigation prioritization [15]. In this paper, we focus on using DT in vulnerability analysis and virtual patches to support the prioritization of potential mitigation strategies.

3. Digital Twin-Based Security Assessment for CPPS

This section introduces the proposed methodology that integrates DT technology to support comprehensive security assessment through enhanced component visibility and vulnerability analysis. We introduce the framework and the reference architecture proposed for CPPS, followed by a detailed discussion on the defined dependence rules and vulnerability assessment methods for vulnerable component prioritization.

3.1. Framework Architecture

The proposed framework illustrated in Figure 3 consists of three key modules: the *CPPS data* layer, which collects both static system configuration and real-time network data such as streaming and machining data; a *security database*, which collects security instances from online sources such as the National Vulnerability Database (NVD) and Microsoft Security Database, correlating with standard enumerations, such as *CWE* and *CAPEC*; and a DT layer, which processes data from the above two modules to support vulnerability retrieval, risk calculation, and virtual patch, enabling patch prioritization. In this paper, the DT model integrates real-time data from the security database to reflect current vulnerability instances and existing exploits, but it has not yet been integrated

with real-time network data. Full integration is planned for future work, as discussed in Section 6. Next, we introduce each module in detail.

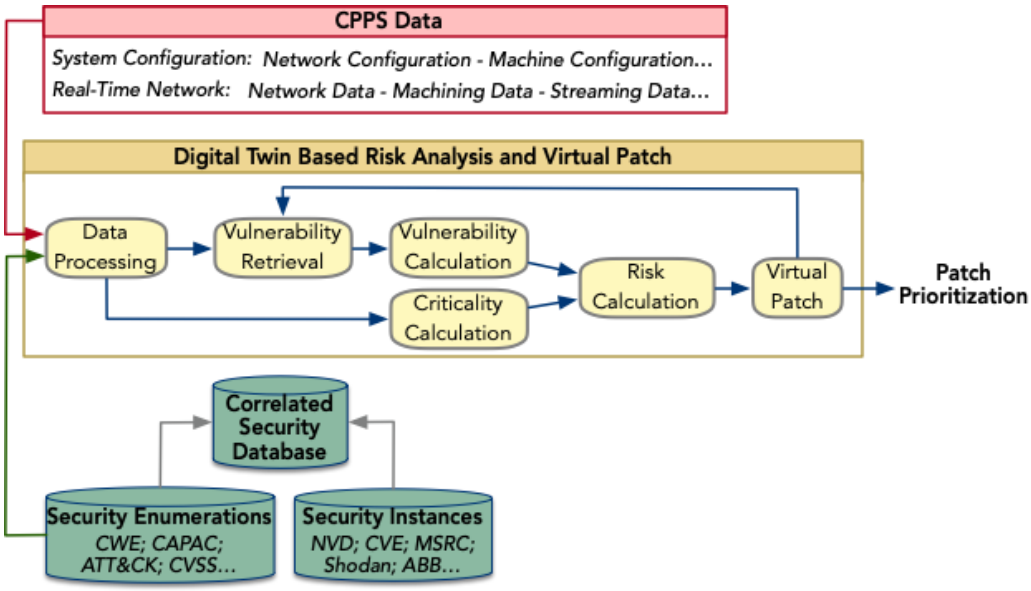

Figure 3. Leveraging Digital twin in security assessment and virtual patching.

3.2. Reference Architecture for CPPS

A DT reference model was developed to simulate the structure of a standard CPPS [50]. This foundational knowledge facilitated the construction of accurate and representative models. We engaged in collaboration with two industrial production experts and two operators from a manufacturing firm, conducting interviews to gather in-depth knowledge about the structure of manufacturing networks. Initially, we established a reference model based on the Purdue model [51] and prevailing industrial standards. Subsequently, we refined and expanded this model in an iterative manner, incorporating feedback from the interview participants to ensure a comprehensive and accurate representation.

Lee Edward A. characterizes cyber–physical systems (CPSs) as the nexus between the physical and cyber realms [52]. However, this work adopts a more expansive view of CPPSs, conceptualizing them as the amalgamation of physical elements, cyber components, and the control mechanisms that bridge these two domains.

3.2.1. Physical Layer of CPPS

The physical layer includes critical components such as the *PLC Gripper* system for controlling grippers, a robotic system for automation, a *Worker Operation* system for operational verification by local operators via mobile devices, and a workstation set up with cameras for tracking worker activities, as illustrated in Figure 4. Physical components are responsible for executing tangible processes, such as production and machining operations. The purpose of including *Worker Operation* and *Worker Identification* is to align our reference model with a focus on Industry 5.0 [53], particularly on human-central dynamics and human factors in manufacturing.

3.2.2. Control Layer of CPPS

The control layer introduces a critical distinction between IT and OT components, enriching the CPPS framework [54]. OT components directly impact physical processes and include devices such as HMI, IED, PLC, and RTU. HMIs serve as control panels, enabling human operators to interact with PLCs and IEDs, which are integral to automating and monitoring physical tasks. PLCs, which are specialized computers within the OT spectrum, execute programs to automate tasks based on sensor inputs, while IEDs, connected to sensors and actuators, facilitate automatic actuation, showcasing the intricate interplay

between cyber and physical components. For example, the control layer empowers human operators to oversee assembly operations through the *PLCController*, with dedicated computers in the control center collecting and displaying production data, as presented in Figure 5. Other important components of this layer include a historian server for historical data retrieval, an application server for data analysis and software support, and supervisory control and data acquisition (SCADA) server and timer for CPPS monitoring and control.

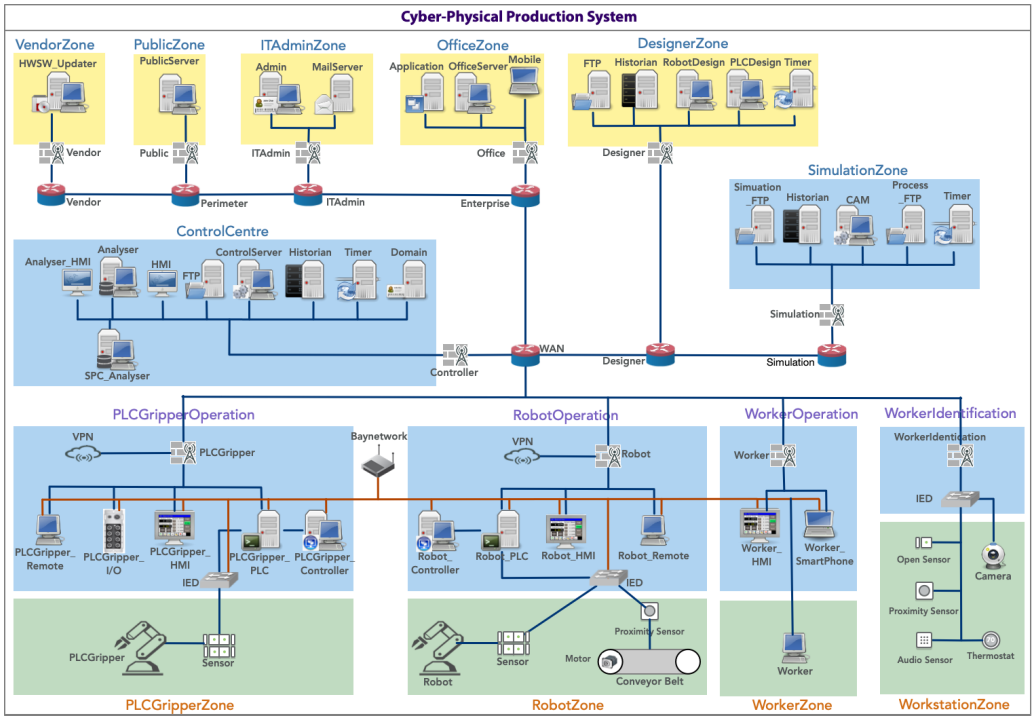

Figure 4. Instantiated cyber–physical production system.

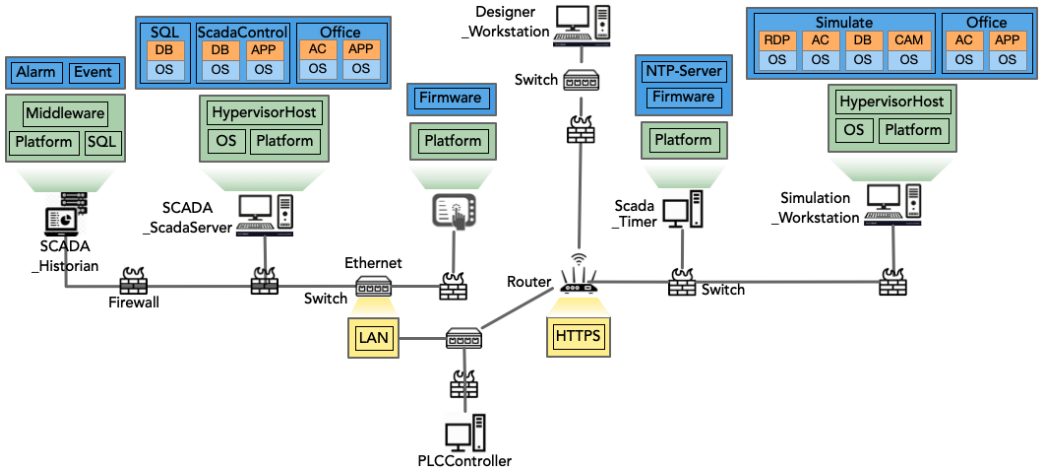

Figure 5. Control layer of the CPPS reference model.

3.2.3. Cyber Layer of CPPS

Cyber components encompass not only software, operating systems, and data storage and transfer but also the networks that facilitate visibility among these elements, as shown in Figure 6. IT components, including devices like routers and switches, are pivotal for information-processing tasks. Specifically, the cyber layer facilitates an enterprise network for internal data sharing and financial transactions, interconnected with the external internet through secure routers and firewalls. This configuration encompasses servers for web and email services, with communication protocols such as HTTPS ensuring the secure and

efficient transfer of data. Additionally, designers and engineers may access the system either onsite or remotely through remote desktop login. Other important components of this layer include a domain controller to implement security measures as access control, and vendor workstation accessed from public internet to maintain or upgrade CPPS.

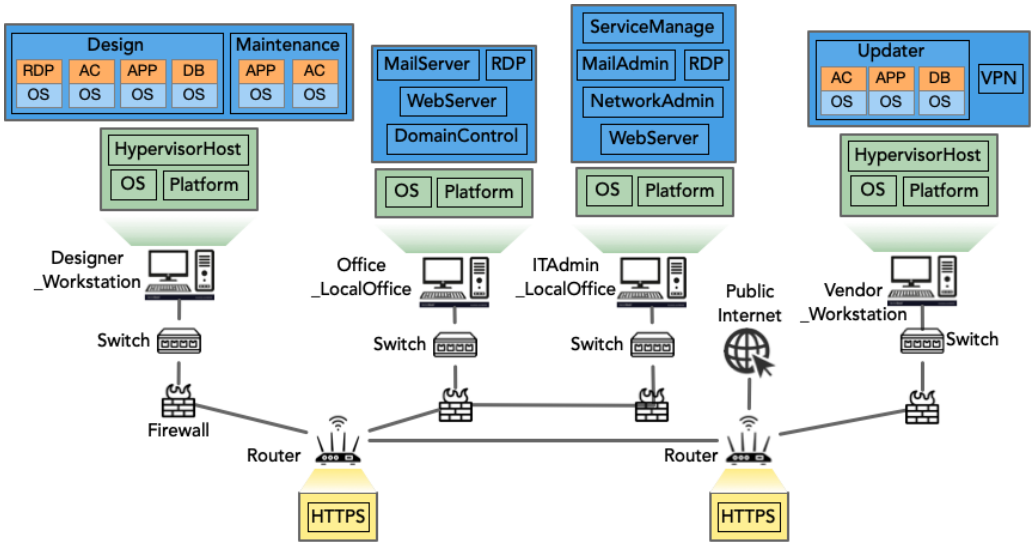

Figure 6. Cyber layer of the CPPS reference model.

An instantiated reference model for the manufacturing system is illustrated in Figure 4 through a layered network diagram, distinguished by color-coded boxes.

3.3. Dependency Analysis and Criticality Calculation

In the proposed CPPS architecture, we pinpoint critical components integral to the organization's operational continuity and mission fulfillment. Our focus lies on data and information components involved in production processes. This includes data stored on memory and hardware disks, as well as data in transit between computing nodes. For instance, programming instructions (e.g., G-code or M-code files) are transmitted from the CAM server to the control server within the control center. Machining data are relayed to the controllers such as the PLC controller for production execution. Simultaneously, processed data and a copy of the machining data are stored in the historian server, which maintains a time-tagged database of the production system's data points. The PLC controller is directly linked to the PLC gripper. Additionally, datasets such as product and manufacturing information, tool condition data, and product inspection data are both critical and confidential, essential for ensuring the system's functionality.

To refine the criticality analysis process, we elaborated on the concept of functional dependencies (FD), as Definition 1, builds upon our previous works [55,56].

Definition 1 (Function Dependence)**.** *If component C_i requires component C_j for its functional operations, then C_i has a functional dependence on C_j, denoted as $FD_{(i,j)}$.*

We introduce seven FD rules to elucidate the complexity of software component interactions, utilizing these for system dependency mapping based on static configuration data. Here, the embedding rule describes dependency arising from one component embedded within another, which is a vertical relationship in the system's architecture. The interaction rule indicates that the dependency is based on the interaction or data exchange between components, reflecting a horizontal relationship.

Consider a cyber component, such as C_i, an IT or OT component, such as C_j, a hypervisor or operating system component, such as C_i, and a physical component, such as C_j. The functional dependency of one component on another is represented by $FD_{(i,j)}$,

indicating that component C_i is functionally dependent on component C_j. The rules can then be formalized as follows:

1. **FD Embedding Rule (ER):**
   - **ER-1:** $FD_{(i,j)} \leftarrow C_i \subseteq C_j$, for C_i embedded in C_j.
   - **ER-2:** $FD_{(j,i)} \leftarrow C_j \subseteq C_i$, for C_j contained in C_i.
2. **FD Interaction Rule (IR):**
   - **IR-1:** $FD_{(k,j)} \leftarrow C_k \xleftarrow{\text{data}} C_j$, for C_k receiving process data from C_j.
   - **IR-2:** $FD_{(j,i)} \leftarrow C_j \xleftarrow{\text{control}} C_i$, for C_j receiving control data from C_i.
3. **FD Data Rule (DR):**
   - **DR-1:** $FD_{(i,j)} \leftarrow C_i \xleftarrow{\text{stream}} C_j$, for C_i as data stream recipient from C_j.
   - **DR-2:** $FD_{(i,j)} \leftarrow C_i \xleftarrow{\text{listen}} C_j$, for C_i listening to the data stream from C_j.
4. **FD Network Rule (NR):**
   - **NR-1:** $FD_{(i,j)} \leftarrow C_i \hookrightarrow [\text{network}]C_j$, for C_i connected to the network via C_j.

$\subseteq$ denotes an embedding relationship, $\xleftarrow{\text{data}}$ and $\xleftarrow{\text{control}}$ represent data and control flow dependencies, respectively, $\xleftarrow{\text{stream}}$ and $\xleftarrow{\text{listen}}$ indicate data stream relationships, and $\hookrightarrow [\text{network}]$ symbolizes network connectivity. These refined FD rules provide a structured framework for identifying and analyzing functional dependencies within a system, thereby enhancing the accuracy and comprehensiveness of vulnerability assessments. In doing so, we establish dependency matrices $FD_{(i,j)}$ between component nodes C_i and C_j. These matrices enable the analysis of centrality and influence levels of nodes.

We further define the criticality of components, and component criticality score (CCS), considering their dependencies. Let $\mathcal{C}$ be the set of components in a system, and M be the total number of components. For each component $C_i \in \mathcal{C}$, we apply Equation (1) to calculate the criticality scores, N_i^{FD}, of these components. A component C_i with a higher value of N_i^{FD} is considered a critical function point, indicating its higher criticality in the system.

$$CCS_i = N_i^{FD}, (0 < i < M) \tag{1}$$

3.4. Vulnerability Virtual Patch and Risk Analysis

To further refine our vulnerability assessment, we gather detailed system configuration and component information, enabling us to query a localized vulnerability database introduced in our previous work [57]. Specifically, we integrated cybersecurity data from diverse open-source repositories, such as *NVD* and *Shodan* into a localized database using *MongoDB*. This integration process also includes the correlation of vulnerability instances to standard enumerations and categorizations such as *CWE* and *CAPEC*.

The vulnerabilities are documented up to the investigation date and are analyzed with the average severity scores associated with each component calculated to reflect the vulnerable levels. The idea of a vulnerability score calculation considering different severity scales is inspired by [58]. We calculate the average score of vulnerabilities under different severity scales according to the Common Vulnerability Scoring System (CVSS) [59], including *none* ([0]), *low* ([0.1–3.9]), *medium* ([4.0–6.9]), *high* ([7.0–8.9]), and *critical* ([9.0–10.0]).

We define the component risk score as CRS, considering multiple contributing factors, including CCS, and the weighted average score of vulnerabilities across different severity scales for each component, such as CVS, as shown in Equations (2) and (3).

$$CRS = CCS \times CVS \tag{2}$$

$$CVS = \left(\sum_{i \in \{C,H,M,L\}} w_i \cdot \text{Sum}(S_i) \right) \cdot \frac{1}{\sum_{i \in \{C,H,M,L\}} N_i} \quad (3)$$

where

- w_i: Weighing factor for each severity level i.
- $\text{Sum}(S_i)$: Sum of vulnerability scores across different severity scales i.
- N_i: The number of vulnerabilities under each scale.

Using CRS, we define the patch prioritization rule in Definition 2.

Definition 2 (Patch Prioritization Rule)**.** *Let C_i denote a component with a set of vulnerability instances, V, existing within it. For each vulnerability instance, $v_j \in V$, the application of a patch influences the component risk score, CRS_j. The prioritization of patches is determined by ranking the component risk scores, CRS_j, in ascending order, from the lowest to the highest.*

Our DT-centered framework also includes pre-defined rules such as the cascading failures rule in Definition 3 to support the attack simulation. Such capability will enable us to further integrate attack simulation-based virtual patching in future works.

Definition 3 (Propagation Rule for Cascading Failure)**.** *If there exists a failure or a component, C_i, where C_j is functionally dependent on ($FD_{(i,j)} = 1$), then the failure is likely to propagate to C_j with a probability, p_{ij}. The propagation probability, p_{ij}, is influenced by system configurations, network structures, and security compliance measures.*

4. Case Study

We evaluated the proposed theoretical framework within a practical setting by implementing an HRC assembly system. This application was instrumental in validating our proposed approach, providing empirical evidence of its efficacy and relevance within the context of contemporary manufacturing practices.

4.1. Human–Robot Collaborative Assembly System

In this study, we employed an HRC assembly system, which involves humans and robots performing concurrent tasks within a shared space. This setup underscores the imperative of robust cybersecurity measures to safeguard human workers from potential harm resulting from compromised robot operations [60].

The HRC assembly system comprises three workstations: a tool-changing station, an *ABB IRB 2600-20(12)/1.65* robot with a PLC gripper, and a conveyor. The system's layout is depicted in Figure 7, while Figure 8 illustrates the physical arrangement of these components. Human operators interact with the robot across these workstations, monitored by cameras (*Microsoft Kinect*) for planning and scheduling the assembly process. Various systems facilitate data flow and command transmission between workstations and the robot: a *UnitController* for assembly data analysis and command issuance, a *Cockpit* for process planning, a *CollisionAvoidance* system for analyzing human–robot movement, and a *WorkerIdentification* system for tracking worker movements.

We refined our reference model by gathering data from three primary sources: outputs from the *SYMBIO-TIC* project, a field study at the ASSAR venue, and interviews with former *SYMBIO-TIC* project participants. More detailed insights into the HRC system can be found in the [61] project, especially the third demonstration at the ASSAR Industrial Innovation Arena in Sweden. The digital-twin model for HRC is presented in Figure 9.

The *WorkerIdentification* and *UnitController* systems independently evaluate the positions and availability of workers and the robot. Utilizing these data, the *Cockpit* system orchestrates the assembly process planning and scheduling for product batches, subsequently relaying these plans to the *UnitController*. The *UnitController* then gathers detailed assembly operation instructions, such as robot movements from graphical robot programming software (e.g., [62]) and gripper commands from robot simulation software (e.g., [63]).

These instructions are converted into executable codes and I/O signals by the *UnitController*, directing both the robot's actions via controllers (e.g., IRC [64]) and the gripper's operations through PLC.

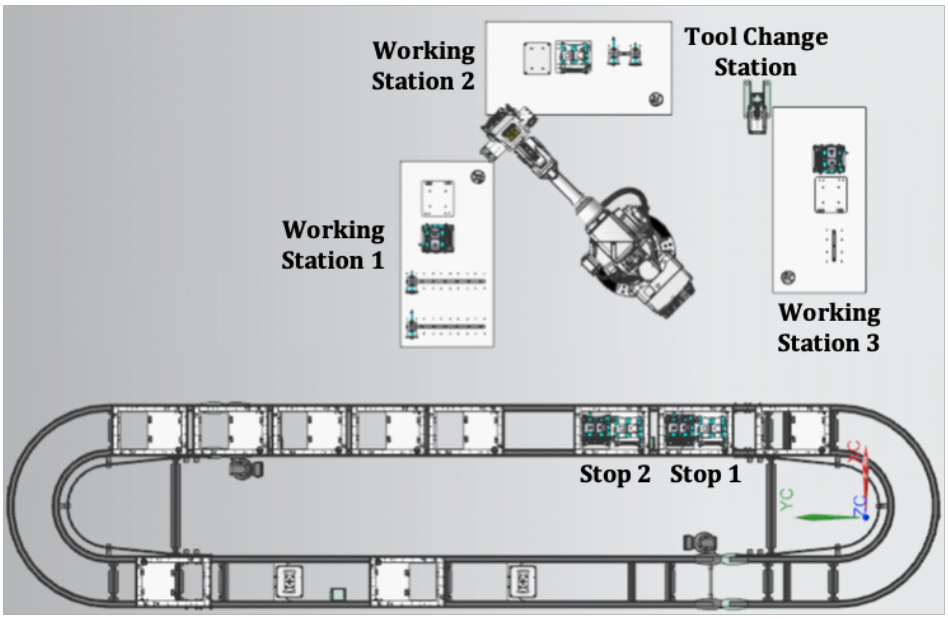

Figure 7. Layout of HRC assembly workstations.

Figure 8. Physical setup of HRC assembly workstations.

Furthermore, the *UnitController* communicates task instructions to workers and workstations via the *HMIC*, typically accessed through mobile devices. To enhance the HRC system's resilience, assembly process data are duplicated and synchronized across the *UnitController*, *CollisionAvoidance*, and *Cockpit* systems. The *CollisionAvoidance* system, upon detecting potential human–robot collisions, adjusts the robot's trajectory and communicates updated instructions to the *UnitController* to prevent accidents.

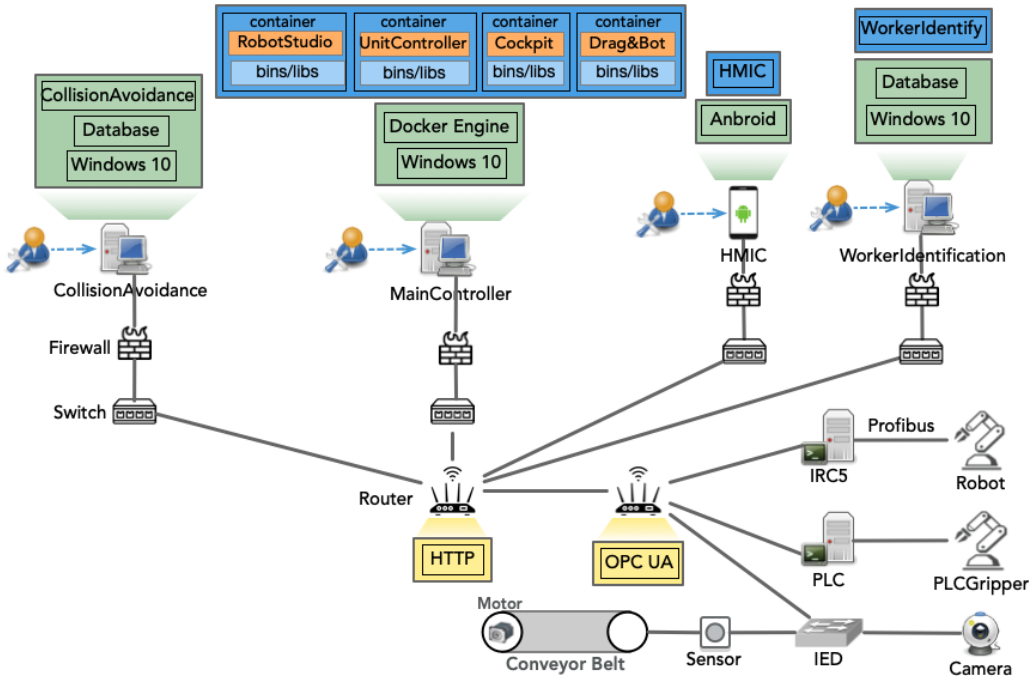

Figure 9. Human–robot collaborative system that integrates *SYMBIO-TIC*.

Simulation and programming tools such as *RobotStudio*, *Drag&Bot*, and others are consolidated within a workstation running *Windows 10* for ×64 systems, as shown in Figure 9. The *CollisionAvoidance* operates on a separate *Windows 10* workstation, while *WorkerIdentification* runs on another *Windows 10* computer. *HMIC* applications are hosted on Android devices. Among these software components, only *Drag&Bot* supports direct remote access. Others, like *Cockpit*, are encapsulated in software containers (e.g., doc [65]) to enhance security and minimize data exposure risks, with strict process communication, memory allocation controls, and role-based access policies.

The network, protected by a password, employs an *ASUS* router with SSH encryption and software firewalls for IP-specific connections, ensuring secure communication and service connectivity.

4.2. Model-Based Vulnerability Assessment for the Human–Robot Collaborative Assembly System

Our criticality study consists of two steps: (i) a criticality calculation using defined dependence rules and Equation (1), and (ii) a conversation with stakeholders to determine the weighting of criticality.

In alignment with the US-CERT [66] asset management guidelines, We first identified assets and then evaluated the critical components within the HRC assembly system. We rank these components by their criticality, determined by how functionally dependent other components are on them. The top five critical components identified include *HRC_MainController_UnitController* with a functional dependency (FD) score of 6, *HRC_Router* and *HRC_MainController_OperatingSystem*, each with an FD score of 5, *HRC_MainController_DockerEngine* with an FD score of 4, and *HRC_MainController_Drag&Bot* with an FD score of 2. To validate and further refine the criticality of these components, we consulted with project members from *SYMBIO-TIC*. In addition to the initially identified components, they emphasized the significance of the physical PLC gripper and robot, as well as the data components exchanged among *RobotStudio*, *Drag&Bot*, *UnitController*, and *Cockpit*, underscoring a comprehensive view of system criticality.

We collected configuration and component information for the HRC assembly system and subsequently formulated queries for our localized vulnerability database, focusing on 14 essential components as illustrated in Figure 9. This process yielded 41 documented vulnerability instances up to 24 February 2024, categorized into 5 critical, 30 high, and

6 medium severity vulnerabilities. Note that, here, we count vulnerabilities by their unified *(CVE)-IDs*.

Table 1 presents the criticality levels, incorporating factors such as functional dependencies, the total number of identified vulnerabilities, and their average severity scores.

In the APP layer, Docker containers bundle program codes and dependencies. A container is reasonably separated from other containers and its host system. Therefore, databases are not shared between computers. Nonetheless, several known Docker vulnerabilities, such as the container breakout vulnerability, allow an attacker to further exploit confined software through a backdoor. Table 1 suggests that the operating system of the *MainController* can be given the highest prioritization.

Table 1. Vulnerability patch decision-making considering criticality and severity.

Component	Criticality	Number of Vulnerability	Average Severity
HRC_MainController_UnitController	6	N/A	N/A
HRC_MainController_RobotStudio	1	1	7.4
HRC_MainController_Cockpit	2	N/A	N/A
HRC_MainController_Drag&Bot	2	N/A	N/A
HRC_MainController_DockerEngine	4	8	7.23
HRC_MainController_OperatingSystem	5	19	7.74
HRC_WorkerIdentification_OperatingSystem	1	19	7.74
HRC_WorkerIdentification_WorkerIdentification	1	N/A	N/A
HRC_CollisionAvoidance_OperatingSystem	2	19	7.74
HRC_CollisionAvoidance_CollisionAvoidance	2	N/A	N/A
HRC_HMIC_OperatingSystem	1	2	7.3
HRC_Router	5	4	8.08
HRC_IRC5	2	2	9.8
HRC_PLC	2	5	7.2

4.3. Attack Simulation Using Digital-Twin Model

Utilizing our DT model and particularly Definition 3, we can effectively simulate a range of vulnerabilities and attack scenarios, assessing their overall impact on the system. This allows us to virtually patch the system and evaluate the impact of such a patch on component risk scores using Definition 2.

We seamlessly integrated our DT security model with a combination of open-source and commercialized tools to facilitate comprehensive attack simulations. One tool utilized is *securiCAD*. Through this integration, the HRC model encompasses an average of 630 components and approximately 885 dependencies, spanning both physical and cyber aspects. Physical dependencies are organized by zones, while cyber dependencies rely on configuration settings and data connections. This model enables detailed simulation of attack scenarios by establishing specific entry points for attackers, which are assumed to occur with certainty. For instance, a phishing attack might involve tricking an internal user into initiating unauthorized data flow to a malicious host, identified as the primary entry point for the attack.

The model introduces vulnerabilities through configurations of deficient defense mechanisms, assigning probabilities to evaluate the risk level of each vulnerability. For example, the minimal likelihood of a firewall's presence indicates a significant risk of access control vulnerabilities. Subsequent analysis examines how attacks spread and their ripple effects, comparing the severity of different scenarios. Key metrics for evaluation encompass the probability of an attack's success, its ramifications on production and safety, and the financial implications of defense measures, as determined by expert assessments.

In this analysis, we evaluate the cascading effects of two models subjected to the same phishing attack, using attack graphs to trace the paths and vulnerability chains, as shown in Figures 10 and 11.

In these graphs, lines with arrows denote attack trajectories, with red lines highlighting the primary path and orange lines indicating secondary or alternative paths. The model depicted in Figure 10, which employs more robust access control measures, demonstrates an average time-to-compromise (TTC) of 110 days for an attacker targeting the robot network. Conversely, the model illustrated in Figure 11, compromised by vulnerabilities such as default password settings, presents an easier target for attackers, reducing the average TTC to 68 days and thereby indicating a significantly higher risk level. Clearly, the configuration represented in Figure 10 emerges as the more secure and preferable option.

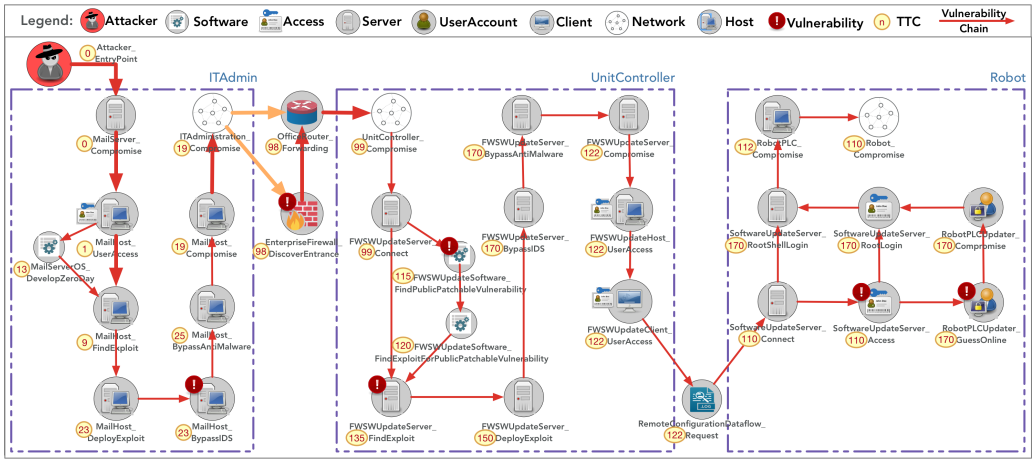

Figure 10. Phishing attack scenario with more secure access control. (Note that the numbers with circles indicate time-to-compromise for each attack path).

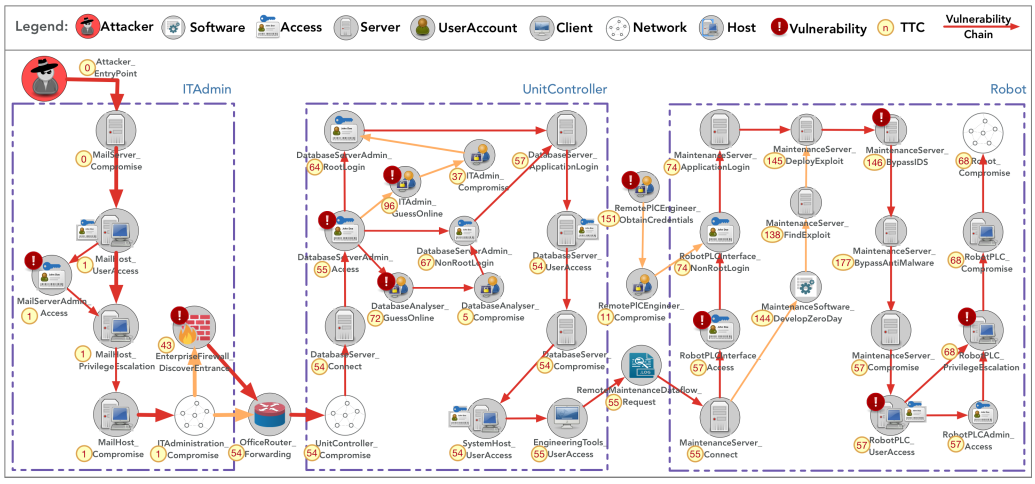

Figure 11. Phishing attack scenario with less secure access control. (Note that the numbers with circles indicate time-to-compromise for each attack path).

5. Discussion

During our model-based evaluation, we identified a structural weakness in the HRC system. Even though all databases (such as *Cockpit*, *HMIC*, and *Drag&Bot* databases) are password-protected, passwords are saved in plain text in configuration files. For instance, the configuration file stores the *RobotStudio* password to enable data connection with *Drag&Bot*. This vulnerability is classified as *CWE-260* and may allow an attacker to obtain privileges or assume identity. Once adversaries obtain access to the *RobotStudio* system, they may alter robot production procedures and damage the entire HRC system.

Additionally, we acknowledge that this static analysis only covers a subset of the system's components, potentially leaving some vulnerabilities unaddressed. Nevertheless, the proposed taxonomy and instantiated models lay the groundwork for further development with more complex systems and elucidated rules for query-based vulnerability analysis.

Through our modeling process and iterative interviews to refine the reference models, we derived insights that highlight the limitations of the Purdue model as a foundational framework. While it served as a starting point, its origins predate the Industry 4.0 era, posing challenges in aligning with the dynamic and interconnected nature of contemporary industrial settings. Its focus on conventional systems may not fully address the security challenges posed by emerging technologies, potentially leaving gaps in cyber resilience. Additionally, we incorporated insights from the challenges posed by Industry 5.0 [53] regarding human-centric dynamics [67,68]. Consequently, we carefully considered worker factors in our reference model and included a human–machine collaborative system in our case study to account for the intricate interplay between humans and automated systems in modern industrial environments.

The instantiated reference models for manufacturing can also function as a knowledge base for IT/OT convergent CI models, which are analyzed by external tools for risk analysis or attack simulations, as illustrated in the previous section.

6. Conclusions

This paper introduces a framework based on DT technology for comprehensive system dependence analysis and support for vulnerability assessment within CPPS. The proposed approach offers a collaborative platform for manufacturing and cybersecurity engineers to collaboratively address cybersecurity issues from a unified standpoint. Through the utilization of DT architecture, the framework enables systematic identification and prioritization of critical components, subsequently subjecting them to vulnerability analysis, attack simulation, and virtual patching. The outcomes of this assessment are quantitatively presented, providing a structured approach for evaluating and ranking vulnerable component mitigation prioritization in CPPS. To exemplify the applicability of the proposed method, an HRC assembly system is scrutinized as a practical case study. Through this case study, we illustrate the effectiveness of our digital twin architecture in identifying critical components and assessing vulnerabilities in an operational context. The instantiated HRC assembly model not only facilitates model-based vulnerability assessment but also aids in the identification of structural vulnerabilities within the system. A significant vulnerability identified was the unencrypted storage of passwords in configuration files, posing a substantial risk if exploited.

Looking ahead, we plan to incorporate simulation-based optimization techniques to explore more efficient configurations across diverse objectives. Another direction of future studies includes employing explainable artificial intelligence techniques [69] for conducting multi-level vulnerability assessments. The goal is to produce fine-grained vulnerability indicators that incorporate environmental and temporal factors, tailoring the granularity of information for stakeholders at various hierarchical levels to ensure optimal situational awareness [70]. We also plan to build on the current virtual patch and mitigation prioritization method from the component level to the asset and system levels.

Author Contributions: Conceptualization, Y.J. (Yuning Jiang) and W.W.; methodology, Y.J. (Yuning Jiang); validation, Y.J. (Yuning Jiang) and W.W.; formal analysis, Y.J. (Yuning Jiang); resources, W.W.; data curation, Y.J. (Yuning Jiang) and W.W.; writing—original draft preparation, Y.J. (Yuning Jiang) and W.W.; writing—review and editing, J.D., X.L. and Y.J. (Yanguo Jing); visualization, Y.J. (Yuning Jiang). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to security concerns.

Acknowledgments: The work is supported by the Knowledge Foundation (KKS), Sweden, through the VF-KDO project and the EU H2020 SYMBIO-TIC project. The authors used *Grammarly* to check the grammar and for English language enhancement. After using this tool, the authors reviewed and edited the content as needed. The authors take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

Abbreviations	Definitions
ATT&CK	adversarial tactics, techniques, and common knowledge
APT	advanced persistent threat
DT	digital twin
CAM	computer-aided manufacturing
CVE	common vulnerability exposures
CWE	common weakness enumeration
CAPEC	common attack pattern enumeration and classification
CPS	cyber–physical system
CPPS	cyber–physical production system
CRS	component risk score
CCS	component criticality score
CVS	component vulnerability score
FD	functional dependence
HRC	human–robot collaborative
HMI	human–machine interface
IED	intelligent electronic device
IT	information technology
NC	numerical control
OT	operational technology
PLC	programmable logic controller
TTC	time to compromise

References

1. Monostori, L.; Kádár, B.; Bauernhansl, T.; Kondoh, S.; Kumara, S.; Reinhart, G.; Sauer, O.; Schuh, G.; Sihn, W.; Ueda, K. Cyber-physical systems in manufacturing. *Cirp Ann.* **2016**, *65*, 621–641. [CrossRef]
2. Mohurle, S.; Patil, M. A brief study of wannacry threat: Ransomware attack 2017. *Int. J. Adv. Res. Comput. Sci.* **2017**, *8*, 1938–1940.
3. Tuptuk, N.; Hailes, S. Security of smart manufacturing systems. *J. Manuf. Syst.* **2018**, *47*, 93–106. [CrossRef]
4. Wu, D.; Ren, A.; Zhang, W.; Fan, F.; Liu, P.; Fu, X.; Terpenny, J. Cybersecurity for digital manufacturing. *J. Manuf. Syst.* **2018**, *48*, 3–12. [CrossRef]
5. Bhamare, D.; Zolanvari, M.; Erbad, A.; Jain, R.; Khan, K.; Meskin, N. Cybersecurity for Industrial Control Systems: A Survey. *Comput. Secur.* **2020**, *89*, 101677. [CrossRef]
6. Asghar, M.R.; Hu, Q.; Zeadally, S. Cybersecurity in Industrial Control Systems: Issues, Technologies, and Challenges. *Comput. Netw.* **2019**, *165*, 106946. [CrossRef]
7. Anton, S.D.D.; Fraunholz, D.; Krohmer, D.; Reti, D.; Schneider, D.; Schotten, H.D. The global state of security in industrial control systems: An empirical analysis of vulnerabilities around the world. *IEEE Int. Things J.* **2021**, *8*, 17525–17540. [CrossRef]
8. Rotibi, A.O.; Saxena, N.; Burnap, P.; Tarter, A. Extended dependency modeling technique for cyber risk identification in ICS. *IEEE Access* **2023**, *11*, 37229–37242. [CrossRef]
9. Samanis, E.; Gardiner, J.; Rashid, A. SoK: A Taxonomy for Contrasting Industrial Control Systems Asset Discovery Tools. In Proceedings of the 17th International Conference on Availability, Reliability and Security, Vienna, Austria, 23–26 August 2022; pp. 1–12. [CrossRef]
10. Staves, A.; Gouglidis, A.; Hutchison, D. An Analysis of Adversary-Centric Security Testing within Information and Operational Technology Environments. *Digit. Threat. Res. Pract.* **2023**, *4*, 1–29. [CrossRef]
11. Elhabashy, A.E.; Wells, L.J.; Camelio, J.A. Cyber-physical security research efforts in manufacturing–a literature review. *Procedia Manuf.* **2019**, *34*, 921–931. [CrossRef]
12. Yampolskiy, M.; King, W.E.; Gatlin, J.; Belikovetsky, S.; Brown, A.; Skjellum, A.; Elovici, Y. Security of additive manufacturing: Attack taxonomy and survey. *Addit. Manuf.* **2018**, *21*, 431–457. [CrossRef]
13. Humayed, A.; Lin, J.; Li, F.; Luo, B. Cyber-Physical Systems Security—A Survey. *IEEE Int. Things J.* **2017**, *4*, 1802–1831. [CrossRef]
14. Qian, C.; Liu, X.; Ripley, C.; Qian, M.; Liang, F.; Yu, W. Digital twin—Cyber replica of physical things: Architecture, applications and future research directions. *Future Int.* **2022**, *14*, 64. [CrossRef]

15. Baiardi, F.; Tonelli, F. Twin based continuous patching to minimize cyber risk. *Eur. J. Secur. Res.* **2021**, *6*, 211–227. [CrossRef]
16. Balta, E.C.; Pease, M.; Moyne, J.; Barton, K.; Tilbury, D.M. Digital twin-based cyber-attack detection framework for cyber-physical manufacturing systems. *IEEE Trans. Autom. Sci. Eng.* **2023**, *21*, 1695–1712. [CrossRef]
17. Tao, F.; Qi, Q.; Wang, L.; Nee, A. Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison. *Engineering* **2019**, *5*, 653–661. [CrossRef]
18. Alshammari, K.; Beach, T.; Rezgui, Y. Cybersecurity for digital twins in the built environment: Current research and future directions. *J. Inf. Technol. Constr.* **2021**, *26*, 159–173. [CrossRef]
19. Pokhrel, A.; Katta, V.; Colomo-Palacios, R. Digital twin for cybersecurity incident prediction: A multivocal literature review. In Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, Seoul, Republic of Korea, 27 June–19 July 2020; pp. 671–678.
20. Aheleroff, S.; Xu, X.; Zhong, R.Y.; Lu, Y. Digital twin as a service (DTaaS) in industry 4.0: An architecture reference model. *Adv. Eng. Inform.* **2021**, *47*, 101225. [CrossRef]
21. Böhm, F.; Dietz, M.; Preindl, T.; Pernul, G. Augmented Reality and the Digital Twin: State-of-the-Art and Perspectives for Cybersecurity. *J. Cybersecur. Priv.* **2021**, *1*, 519–538. [CrossRef]
22. Vielberth, M.; Glas, M.; Dietz, M.; Karagiannis, S.; Magkos, E.; Pernul, G. A digital twin-based cyber range for SOC analysts. In Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy, Calgary, AB, Canada, 19–20 July 2021; Springer: Cham, Switzerland, 2021; pp. 293–311.
23. Moyne, J.; Qamsane, Y.; Balta, E.C.; Kovalenko, I.; Faris, J.; Barton, K.; Tilbury, D.M. A requirements driven digital twin framework: Specification and opportunities. *IEEE Access* **2020**, *8*, 107781–107801. [CrossRef]
24. Qamsane, Y.; Moyne, J.; Toothman, M.; Kovalenko, I.; Balta, E.C.; Faris, J.; Tilbury, D.M.; Barton, K. A methodology to develop and implement digital twin solutions for manufacturing systems. *IEEE Access* **2021**, *9*, 44247–44265. [CrossRef]
25. Mahoney, T.C.; Davis, J. *Cybersecurity for Manufacturers: Securing the Digitized and Connected Factory*; Technical Report; University of Michigan Library: Ann Arbor, MI, USA, 2017.
26. Zio, E. Challenges in the vulnerability and risk analysis of critical infrastructures. *Reliab. Eng. Syst. Saf.* **2016**, *152*, 137–150. [CrossRef]
27. Common Weakness Enumeration (CWE). Available online: https://cwe.mitre.org/index.html (accessed on 23 February 2024).
28. Tatam, M.; Shanmugam, B.; Azam, S.; Kannoorpatti, K. A review of threat modelling approaches for APT-style attacks. *Heliyon* **2021**, *7*, e05969. [CrossRef] [PubMed]
29. Common Attack Pattern Enumeration and Classification (CAPEC). Available online: https://capec.mitre.org/index.html (accessed on 23 February 2024).
30. Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK). Available online: https://attack.mitre.org/ (accessed on 23 February 2024).
31. Wells, L.J.; Camelio, J.A.; Williams, C.B.; White, J. Cyber-physical security challenges in manufacturing systems. *Manuf. Lett.* **2014**, *2*, 74–77. [CrossRef]
32. Sturm, L.D.; Williams, C.B.; Camelio, J.A.; White, J.; Parker, R. Cyber-physical vulnerabilities in additive manufacturing systems: A case study attack on the. STL file with human subjects. *J. Manuf. Syst.* **2017**, *44*, 154–164. [CrossRef]
33. DeSmit, Z.; Elhabashy, A.E.; Wells, L.J.; Camelio, J.A. An approach to cyber-physical vulnerability assessment for intelligent manufacturing systems. *J. Manuf. Syst.* **2017**, *43*, 339–351. [CrossRef]
34. Elhabashy, A.E.; Wells, L.J.; Camelio, J.A.; Woodall, W.H. A cyber-physical attack taxonomy for production systems: A quality control perspective. *J. Intell. Manuf.* **2019**, *30*, 2489–2504. [CrossRef]
35. Kure, H.; Islam, S.; Razzaque, M. An Integrated Cyber Security Risk Management Approach for a Cyber-Physical System. *Appl. Sci.* **2018**, *8*, 898. [CrossRef]
36. Noel, S.; Harley, E.; Tam, K.; Limiero, M.; Share, M. CyGraph: Graph-Based Analytics and Visualization for Cybersecurity. In *Handbook of Statistics*; Elsevier: Amsterdam, The Netherlands, 2016; Volume 35, pp. 117–167.
37. Lallie, H.S.; Debattista, K.; Bal, J. An Empirical Evaluation of the Effectiveness of Attack Graphs and Fault Trees in Cyber-Attack Perception. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1110–1122. [CrossRef]
38. Tayouri, D.; Baum, N.; Shabtai, A.; Puzis, R. A survey of MulVAL extensions and their attack scenarios coverage. *IEEE Access* **2023**, *11*, 27974–27991. [CrossRef]
39. Alcaraz, C.; Lopez, J. Digital twin: A comprehensive survey of security threats. *IEEE Commun. Surv. Tutorials* **2022**, *24*, 1475–1503. [CrossRef]
40. Eckhart, M.; Ekelhart, A. Digital Twins for CYBER-Physical Systems Security: State of the Art and Outlook. In *Security and Quality in Cyber-Physical Systems Engineering: With Forewords by Robert M. Lee and Tom Gilb*; Springer: Cham, Switzerland, 2019; pp. 383–412.
41. Eckhart, M.; Ekelhart, A. Towards security-aware virtual environments for digital twins. In Proceedings of the 4th ACM Workshop on Cyber-Physical System Security, Incheon, Republic of Korea, 4–8 June 2018; pp. 61–72.
42. Eckhart, M.; Ekelhart, A. A specification-based state replication approach for digital twins. In Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and Privacy, Toronto, ON, Canada, 15–19 October 2018; pp. 36–47.
43. Lou, X.; Guo, Y.; Gao, Y.; Waedt, K.; Parekh, M. An idea of using Digital Twin to perform the functional safety and cybersecurity analysis. In Proceedings of the INFORMATIK 2019: 50 Jahre Gesellschaft für Informatik–Informatik für Gesellschaft (Workshop-Beiträge), Kassel, Germany, 23–26 September 2019.

44. Bécue, A.; Fourastier, Y.; Praça, I.; Savarit, A.; Baron, C.; Gradussofs, B.; Pouille, E.; Thomas, C. CyberFactory# 1—Securing the industry 4.0 with cyber-ranges and digital twins. In Proceedings of the 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), Imperia, Italy, 13–15 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–4.
45. Bitton, R.; Gluck, T.; Stan, O.; Inokuchi, M.; Ohta, Y.; Yamada, Y.; Yagyu, T.; Elovici, Y.; Shabtai, A. Deriving a cost-effective digital twin of an ICS to facilitate security evaluation. In Proceedings of the Computer Security: 23rd European Symposium on Research in Computer Security, ESORICS 2018, Barcelona, Spain, 3–7 September 2018; Proceedings, Part I 23; Springer: Cham, Switzerland, 2018; pp. 533–554.
46. Suhail, S.; Iqbal, M.; Hussain, R.; Jurdak, R. ENIGMA: An explainable digital twin security solution for cyber–physical systems. *Comput. Ind.* **2023**, *151*, 103961. [CrossRef]
47. Damjanovic-Behrendt, V. A digital twin-based privacy enhancement mechanism for the automotive industry. In Proceedings of the 2018 International Conference on Intelligent Systems (IS), Funchal, Portugal, 25–27 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 272–279.
48. Sellitto, G.P.; Masi, M.; Pavleska, T.; Aranha, H. A Cyber security digital twin for critical infrastructure protection: The intelligent transport system use case. In Proceedings of the IFIP Working Conference on the Practice of Enterprise Modeling, Riga, Latvia, 24–26 November 2021; Springer: Cham, Switzerland, 2021; pp. 230–244. [CrossRef]
49. Lu, Y.; Liu, C.; Kevin, I.; Wang, K.; Huang, H.; Xu, X. Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robot. Comput.-Integr. Manuf.* **2020**, *61*, 101837. [CrossRef]
50. Liu, S.; Zheng, P.; Bao, J. Digital Twin-based manufacturing system: A survey based on a novel reference model. *J. Intell. Manuf.* **2023**, 1–30. . [CrossRef]
51. Williams, T.J. The Purdue enterprise reference architecture. *Comput. Ind.* **1994**, *24*, 141–158. [CrossRef]
52. Lee, E.A. The past, present and future of cyber-physical systems: A focus on models. *Sensors* **2015**, *15*, 4837–4869. [CrossRef]
53. Aheleroff, S.; Huang, H.; Xu, X.; Zhong, R.Y. Toward sustainability and resilience with Industry 4.0 and Industry 5.0. *Front. Manuf. Technol.* **2022**, *2*, 951643. [CrossRef]
54. Tao, F.; Qi, Q. New IT driven service-oriented smart manufacturing: Framework and characteristics. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *49*, 81–91. [CrossRef]
55. Jiang, Y.; Jeusfeld, M.A.; Ding, J.; Sandahl, E. Model-Based Cybersecurity Analysis: Extending Enterprise Modeling to Critical Infrastructure Cybersecurity. *Bus. Inf. Syst. Eng.* **2023**, *65*, 643–676. [CrossRef]
56. Jiang, Y. Vulnerability Analysis for Critical Infrastructures. Ph.D. Thesis, University of Skövde, Skövde, Sweden, 2022.
57. Jiang, Y.; Atif, Y.; Ding, J. Cyber-physical systems security based on a cross-linked and correlated vulnerability database. In Proceedings of the International Conference on Critical Information Infrastructures Security, Copenhagen, Denmark, 24–26 August 2019; Springer: Cham, Switzerland, 2019; pp. 71–82.
58. Jacobs, J.; Romanosky, S.; Adjerid, I.; Baker, W. Improving vulnerability remediation through better exploit prediction. *J. Cybersecur.* **2020**, *6*, tyaa015. [CrossRef]
59. Common Vulnerability Scoring System (CVSS). Available online: https://www.first.org/cvss/ (accessed on 23 February 2024).
60. Wang, L.; Gao, R.; Váncza, J.; Krüger, J.; Wang, X.V.; Makris, S.; Chryssolouris, G. Symbiotic human-robot collaborative assembly. *Cirp Ann.* **2019**, *68*, 701–726. [CrossRef]
61. Symbiotic Human-Robot Collaborative Assembly: Technologies, Innovations and Competitiveness. Available online: https://cordis.europa.eu/project/id/637107 (accessed on 23 February 2024).
62. Drag&Bot. Available online: https://www.dragandbot.com/ (accessed on 23 February 2024).
63. RobotStudio. Available online: https://new.abb.com/products/robotics/robotstudio (accessed on 23 February 2024).
64. IRC5. Available online: https://new.abb.com/products/robotics/controllers/irc5 (accessed on 23 February 2024).
65. Docker. Available online: https://www.docker.com/products (accessed on 23 February 2024).
66. US-CERT Asset. Available online: https://www.cisa.gov/protect-assets (accessed on 23 February 2024).
67. Jiang, Y.; Atif, Y.; Ding, J.; Wang, W. A Semantic Framework with Humans in the Loop for Vulnerability-Assessment in Cyber-Physical Production Systems. In Proceedings of the International Conference on Risks and Security of Internet and Systems, Hammamet, Tunisia, 29–31 October 2019; Springer: Cham, Switzerland, 2019; pp. 128–143.
68. Siyaev, A.; Valiev, D.; Jo, G.S. Interaction with industrial digital twin using neuro-symbolic reasoning. *Sensors* **2023**, *23*, 1729. [CrossRef] [PubMed]
69. Liao, Q.V.; Gruen, D.; Miller, S. Questioning the AI: Informing Design Practices for Explainable AI User Experiences. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, New York, NY, USA, 25–30 April 2020; pp. 1–15. [CrossRef]
70. Elder, S.; Rahman, R.; Fringer, G.; Kapoor, K.; Williams, L. A Survey on Software Vulnerability Exploitability Assessment. *ACM Comput. Surv.* **2024**. [CrossRef]