International Journal of Information Security (2024) 23:187–198
https://doi.org/10.1007/s10207-023-00731-w

REGULAR CONTRIBUTION

Enhancing attack resilience of cyber-physical systems through state dependency graph models

Konstantinos Adamos[1] · George Stergiopoulos[1] · Michalis Karamousadakis[2] · Dimitris Gritzalis[3]

Published online: 22 July 2023

Abstract

This paper presents a method that utilizes graph theory and state modelling algorithms to perform automatic complexity analysis of the architecture of cyber-physical systems (CPS). It describes cyber physical systems risk assessment (CPSRA), a tool to provide automatic decision support for enhancing the overall resilience of CPS architectures often used in critical infrastructures. CPRSA is built to enhance industrial risk assessment and improve the resilience of CPS architecture against malicious attacks on the cyber domain that can affect industrial processes, which is critical in a distributed cyber environment. Such attacks often compromise execution states on physical components and lead to hazards or even disasters through plant malfunction. CPSRA is tested against a real-world testbed model of a large SCADA system that is infused with real-world CVE vulnerabilities in some of its components. The tool creates an isomorphic graph of the CPS process model and uses graph algorithms and network analytics on the model to test cyber-attacks and evaluate attack resilience aspects. The tool's output is then used to pinpoint high-complexity components in terms of influence on the overall CPS architecture and suggest mitigation points for security measure implementation while considering every potential subattack path and subliminal path on the model's attack graph. The paper complements standardized assessment reports and contributes to automatic architecture assessment for critical infrastructure environments and can be used as the basis to model dependencies and threat propagation in larger digital twins, a need outlined in major NIST publications concerning the security of industrial systems that was previously done manually, without automatic insight into state and vulnerability influences.

Keywords Resilience · Cyber-physical systems · Attack graphs · Centrality

1 Introduction

NIST defines Resilience as "the ability of an information system to continue to: (i) operate under adverse conditions or stress, even if in a degraded or debilitated state, while maintaining essential operational capabilities; and (ii) recover to an effective operational posture in a time frame consistent with mission needs" [13]. Resilient systems implement built-in defense mechanisms for minimizing the impact of both accidental and malicious threats [43]. Cyber resiliency refers specifically to the resilience of cyber resources. Cyber resiliency is "intended to enable mission or business objectives that depend on cyber resources to be achieved in a contested cyber environment" [32].

Concerning information technology (IT) and operational technology (OT) systems, a major part of cyber resiliency relates to evaluating a system's architecture that employs technologies and procedures to limit an attacker's ability to compromise an organizational system and achieve a persistent presence in the system [33]. This architecture refers to both the networking of components and the layout of information flow during an organization's execution of its own business processes; whether industrial or not. Concerning industrial control systems, standards and guidelines [14, 40]

✉ Dimitris Gritzalis
dgrit@aueb.gr

Konstantinos Adamos
kadamos@aegean.gr

George Stergiopoulos
g.stergiopoulos@aegean.gr

Michalis Karamousadakis
michalis.karamousadakis@twi.gr

[1] Department of Information and Communication Systems Engineering, University of the Aegean, 832 00 Samos, Greece

[2] TWI-HELLAS, 152 32 Halandri, Greece

[3] Department of Informatics, Athens University of Economics and Business (AUEB), Athens, Greece

typically recommend separating the OT network(s) from the corporate networks when establishing the security architecture for an OT environment for two main reasons: First, (i) the inherent nature of network traffic is different on these two networks, and second (ii) interconnections between IT and OT systems introduce additional attack vectors for adversaries to compromise OT components. Such segregation is often supported by a defence-in-depth strategy that focuses attention and defensive mechanisms on critical functions [40](i.e. implementing managerial and technical security measures across multiple layers and dimensions of a system; from people to equipment).

Still, various controls are not applicable across all Industrial Control Systems (ICS) and Cyber-Physical Systems (CPS) in general [40]. Also, the inherent nature of some industrial processes prohibits the use of typical security controls, e.g. IEC 61850 references that network encryption of data should not be used when control systems require a response time of <= 4ms [12]. As such, critical infrastructure operators are "encouraged to perform a risk-based assessment on their systems to tailor the recommended guidelines and solutions to meet their specific security, business and operational requirements" [40].

Yet, assessing the risk of potential threats to industrial systems is a complex, tedious and error-prone process that requires detailed knowledge of a CPS and its underlying processes. CPS consist of a complex network of interactive components, the size and complexity of which often force OT experts to only focus on critical components or implement out-of-the-box control lists without granular analysis of each CPS's needs. This inadvertently leads to errors or unintentional actions. According to a study by IBM, errors in implementation are the main cause of 95% of cyber security breaches [10]. A survey by SCADAfence found that nearly 80% of respondents said human error presents the greatest risk to OT control systems [34]. Architecture and process complexity in industrial control systems is known to lead to errors and faults, such as cascading failures [31, 37], due to design, implementation and quality control errors. Common errors include mistakes made on the network configuration that can have serious consequences [40].

1.1 Contribution

In this work, we utilize algorithms from graph theory and network analytics to enhance the results of an industrial risk assessment. We build a tool called CPSRA and demonstrate the benefits of automatic complexity analysis of CPS based on their risk assessment results on industrial system resilience. Algorithms behind CPSRA have been previously used for automatically analyzing multi-cloud enterprise networks and a NetFlixOSS microservices Docker architecture and provided insight on component risk, exposure to failures due to complexity and provided network security solutions for risk mitigation on large-scale [39]. We extend this concept to a framework able to automatically analyze CPS architectures and their underlying processes and provide automatic suggestions for enhancing the overall cyber-resilience of a CPS architecture against potential cyber and cyber-physical attacks on a CPS's processes.

CPSRA builds on [39] and uses the DIRE curve as a reference point for providing resilience metrics calculation using the NIST Cybersecurity Framework seven-step process [5]. In summary, this paper contributes the following:

1. We introduce the concept of automatic CPS architecture analysis based on states achieved by components, for decision support CPS environments using graph theory and network analytics.
2. We then build a model testbed of a real-world large SCADA system based on a proposal railway system zone model comprised of multiple stations and processes from CLS/TS 50701:2021. The model is then infused with real-world CVE vulnerabilities [22] in some of its components that realistically depict potential CPS implementations. CVEs are taken from the U.S. National Vulnerability Database (NVD) [27] concerning specific equipment (e.g. Rockwell ControlLogix 1756-5585 PLC).
3. CPSRA then creates a model that is an isomorphic graph to the original testbed and inputs assessment data to indicative processes (either from CVEs or indicative threat and impact values based on NIST's assessment scales).
4. The tool then executes graph algorithms and network analytics on the model to test theoretical resilience aspects within nonlinear cyber-physical processes implemented in the SCADA. We use CPSRA's output to propose which system states, vulnerabilities and configurations have the most significant overall risk to the CPS architecture, pinpoint high-complexity components per industrial process and suggest mitigation points for security measure implementation while considering every potential subattack path and subliminal path on the model's attack graph.

Such graph models are also often used to create Digital Twins of processes regardless of scale: from IT to smart city levels. This work will act as a baseline in designing and developing state-model graphs that can be enlarged to create a Digital Ecosystem for Resilience utilising Smart City Digital Twin (SCDT) technology that analyzing attacks or disasters in multiple ICS from multiple Critical Infrastructures in a given society, leveraging new and emerging technologies and innovations to improve risk assessment, reduce vulnerability, and building community disaster resilience. The aim is to enhance the operational capabilities of Disaster

Risk Management from Industrial Control systems in critical infrastructures with components used for simulations, training and evaluation of the behaviour of sub-systems, threats and human factor; an early Detection and Situational awareness environment and tool for assessment of risks, vulnerability and capacity assessment.

1.2 Structure

Sect. 2 briefly presents relevant research publications and compares our contributions to existing literature. Section 3 introduces CPSRA's main building blocks and algorithms, along with methods to model CPS processes using the tool. Section 4 presents the methodology, along with the input and output for each step. In Sect. 5 we build the aforementioned railway model and evaluate CPSRA for decision support on security and resilience objectives. Section 6 concludes our work and focuses on current limitations and potential future challenges.

2 Related work

2.1 CPS resilience

Several techniques have been introduced to enhance the resilience of CPS processes and industrial components. Researchers studied various aspects of cyber resilience in these systems and have proposed different solutions to enhance their security and resilience against cyber-attacks. Their focus has been on developing frameworks, methodologies and metrics to enhance the security and functionality of these systems.

Bodeau et al. [3] proposed a cyber resiliency engineering framework with four (4) goals, i.e. anticipate, withstand, recover, and evolve, and eight (8) objectives to meet these goals, i.e. understand, prepare, prevent, continue, constrain, reconstitute, transform and re-architect. Sterbenz et al. [36] developed a resilience evaluation framework that evaluates network resilience with the use of a resilient strategy. The two (2) phases of the resilient strategy consist of six (6) steps, i.e. defend, detect, remediate, recover, diagnose, and refine. Francis et al. [8] introduced a resilience analysis framework that includes system identification, objective setting, and vulnerability analysis and focuses on resilience capacities to provide quantitive resilience metrics and assess resilience in infrastructure systems (i.e. in a power grid).

Authors in [18] defined a new model for testing the resilience of ICS and CPS through Chaos Engineering, a technique first introduced by Netflix to test decentralized content management networks directly in the production environment [2]. Although novel, this approach mostly utilizes state transitions for fault analysis, rather than aiming at architectural design errors that affect cyberattacks.

The authors of [20], with the aim of improving resilience when converging OT and IT systems in industrial environments, introduced component diversity as a solution. They utilize the NVD database and a similarity metric to evaluate the vulnerability similarity and infection rate between IT-OT components in industrial environments. Through the application of the discrete Markov Random Field (MRF), the authors claim that can determine the optimal level of diversification of industrial components in large-scale industrial networks to avoid exploits e.g. malware propagation, in components with similar vulnerabilities.

Haque et al. [9] extended the R4 resilience framework [41] to develop ICS-CRAT, an ICS cyber resilience assessment tool that utilizes resilient metrics based on a qualitive ICS data approach to enable security experts to make informed decisions regarding mitigation measures. In [35], Schenato et al. consider the problem of control and estimation in a networked system when the communication links are subject to disturbances (corresponding to packet losses), resulting from a denial-of-service (DoS) attack, for instance. The estimation and control of linear systems, when some of the sensors or actuators are corrupted by an attacker, is studied in [6]. In that work, they propose an efficient algorithm, inspired by techniques in compressed sensing, to estimate the state of the plant despite attacks. Paridari et al. [29] introduce an ICS cyber-resilient framework that can detect anomalies (e.g. cyber-attacks) with the use of virtual sensors and ML algorithms and apply fault-tolerant control techniques to reduce cyber-attack effectiveness and enhance the resilience of industrial networks.

Previous work in CPS resilience either focuses on generic frameworks that relate or extend notions around theoretic assessment and NIST CyberSecurity Framework (CSF) [3, 8, 36] or provides technical approaches to measuring state disturbances for the detection of cyberattacks (e.g. DoS) [35]. Our work endeavours to address limitations during experts' assessment of industrial infrastructures, by taking as input risk scenarios and outcomes along with formal CVEs detected during expert assessment and automatically evaluate the cyber-resilience of the underlying CPS architecture based on the actual business processes implemented with the industrial components.

2.2 Attack graphs for CPS resilience

A variety of work has utilized attack graphs to enhance the resilience of CPS processes and industrial components [1, 46]. In [46], the authors present SOCCA, a CPS contingency analysis framework that generates a directed graph of CPS components and then uses the Markov Decision Process (MDP) to create a state-based model. The framework assesses contingencies (e.g. vulnerabilities) by using the MDP state-based model and the already established detection systems to discover and rank these contingencies based on their impact and attack complexity. Authors of [44] propose a vulnerability assessment model to quantify the risk of CPS zero-day vulnerabilities with the use of attack graphs. In their work, they use the CVSS metric for known vulnerabilities and then apply their zero-day attack graph algorithm to show that new vulnerable nodes are added to the same attack graph. Al Ghazo and Ratnesh in [1] present an approximate solution to the min label-cut (MLC) problem. They compute an abstracted attack graph and then iteratively step-by-step identify the cut-edges of set nodes that lead to the terminal node. This way they identify critical attack paths in attack graphs of industrial environments.

Our work is similar to [46] but we use states to model the entire architecture of a CPS instead of Markov processes for modelling state transitions. This allows us to evaluate the overall architecture and analyze attack paths without the need for full state transitions for each component. Contrary to the other aforementioned publications, we use states as graph nodes and implement Centrality metrics and depth-first analysis in the graph's subtrees to analyze the underlying infrastructure. Thus, our work can analyze potential attack paths and the influence of components inside an architecture, rather than evaluate states themselves.

3 An isomorphic process model

3.1 CPS process modeling

A generic CPS operation model can be seen as a nonlinear system often described by the following equation set:

$$x(t) = g(x(t), u(t), \delta(t)) \tag{1}$$

$$\Delta y(t) = h(x(t), u(t), \delta(t)) \tag{2}$$

where the state $x(t) \in R^n$ with $x(0) = x_0$, the measurement output deviations $\Delta y(t) \epsilon R^m$, the control input $u(t) \in R^m$ representing set-points in processes, and $N = 1, ..., n$ representing the index set of the CPS components used in the overall CPS architecture. $\delta(t) \in R^n_\delta$ denotes the vector of potential disturbances and/or measurement noise, while $f(\cdot)$ and $g(\cdot)$ are the generic nonlinear functions to be estimated based on the problem statement.

An industrial process is essentially a feedback control loop that uses a function on reference input to control an industrial process. The function contributes towards maintaining a particular system property in a specific pattern or within a specific range. The development of individual process models and their information flow requires the model to be able to describe the steady states according to the anticipated conditions of the system metrics [18].

CPS automated functions assign tasks to system components that operate within component setpoints. The sensor network of a CPS can be viewed as a directed graph $G = (V, E, A)$ with nodes $V = 1, ..., m$, edges $E \subseteq V \times V$, and $A = [a_{ij}] \in R^{m \times m}$ which is a weighted adjacency matrix with non-negative adjacency elements. The adjacency value a_{ij} is greater than 0 if and only if the edge $(i, j) \in E$, where $|V| = m$ is the system node set. Each node $\upsilon_i \in V$ of the graph model is considered globally reachable because a path exists from υ_i to every other node [45].

CPSRA's modelling approach utilizes the graph depiction method proposed in [11]. Each node in graph G represents a potential state of a CPS component; i.e. the result of an action that produces a specific system state proven able to happen on a specific component (e.g. a PLC), since previous logical dependencies leading up to that derived fact are true. Nodes are the result of applying component interactions iteratively (represented by edges).

A directed edge illustrates the dependency of a system state (node) Vj on another Vi, i.e. $Vi \rightarrow Vj$. Edge dependencies effectively construct logical dependency paths of components and their states that may be used as potential attack paths [23]. Each edge depicts a different derivation, so the number of edges is equal to all possible states' derivations from observed system configurations [21, 45]. Edges represent logical dependencies between potential system states and contain logical requirements as attributes. These attributes reflect the preconditions for an attacker to realize a step/achieve a system state. Attributes can either be configuration primitives (an implemented system configuration state) or derived facts detected during the analysis of primitives (e.g. vulnerability CVE-2019 exists on a web server). Primitives are generally configuration information of systems, as reported by the host and network scanners (e.g. "access control list granted" that indicates that a firewall permits access to a server).

This modelling approach makes certain principal assumptions in order to be able to properly describe the steady states according to the anticipated conditions of a system. Specifically:

- The states of the modelled system are all considered stabilizable and protected from measurement noise.
- The linear closed-loop system (1) (2) is stable, which requires matrix A to be stable (i.e. have reproducible values) and that control input $u(t)$ to be observable for all modelled processes.
- The dependency graph model assumes a single initiating event (disruption) at a single component that results in cascading disruptions of states in other components inside a process flow.

3.2 Dependency paths and cumulative risks

A graph edge denotes a derivation, i.e. $Vi \rightarrow Vj$; thus, it inherits a risk relation that is derived from a dependence of state Vj on an accessible/available vulnerability provided by state Vi). Based on risk assessment standards [1, 3, 4, 5], the methodology quantifies the risk of each graph edge using the impact Ii, j, and the likelihood Li, j of a vulnerability being exploited. The product of these two values is defined as the dependency risk Ri, j of system state Vj due to its dependence on state Vi. The numerical value of each edge is the level of the cascade risk between the receiver and the sender node. This risk is depicted using a risk scale [1 to 10] where 10 is the most severe risk.

The algorithm assesses the nth-order cascading risks or attack paths using a recursive algorithm based on [19, 37, 38]. If $S1 \rightarrow S2 \rightarrow ... \rightarrow Sn$ is an nth-order dependency between n system states S, with weights $L_{i,i+1}$, $I_{i,i+1}$ corresponding to each first-order dependency of the attack path, then the cascading risk $R1, ..., n$ exhibited by Sn for this state dependency path is computed as shown in Eq. 3. The presented method calculates the Cascading risk of a system state dependency path as follows:

$$R_{1,...,n} = L_{1,...,n} I_{n-1,n} = (\Pi_{i=1}^{n-1} L_{i,i+1}) I_{n-1,n} \tag{3}$$

The cumulative dependency risk (as defined in Eq. 3) represents the overall risk of the system states that participate within a specific attack path of an nth-order dependency. For a chain of system states dependencies, $S1 \rightarrow S2 \rightarrow ... \rightarrow Sn$, the cumulative dependency risk (denoted as $CR1, ..., n$) is the total risk resulting from the nth-order dependency:

$$CR_{1,...,n} = \Sigma_{i=1}^{n} - R_{1,...,i} = \Sigma_{i=1}^{n} (\Pi_{j=1}^{i-1} L_{j,j+1}) I_{i-1,I} \tag{4}$$

The overall dependency risk is calculated using Eq. 4, which involves adding up the dependency risks of the nodes affected in the chain due to a system state in the source node. The methodology can calculate the overall risk of the graph (G_r, eq. 5) by summing the cumulative dependency risk for each nth-order dependency in the graph. This is done by using the total number (n) of all system state sub-chains (i.e., possible attack paths) and their cumulative dependency risks.:

$$G_r = \Sigma_{i=1}^{n} CR_{1,...,n} \tag{5}$$

When considering interdependent assets, it is important to follow the relationship between assets and their dependent nature, or else to quantify each assets influence to other assets. We follow extended work in the area [19, 37, 38] that utilize the sum of the likelihood of an asset influencing another, dependent asset times the impact of this influence. The sum mathematically models the recursive algorithm of asset influence to each other, by examining each asset as the root of a dependency, construct the chain of its n-order dependencies, and assess the dependency risk of each chain through the summary of all interconnected, common-threat incidents [19].

Low-risk edges can be omitted given a threshold on the maximum amount of nodes that participate inside a chain. The threshold is usually considered to be less than 6 nodes [19, 38] after carefully considering the scale reduction of dependent probabilities for a threat to actually impact an asset after a given amount of propagation. In any case, practitioners of the methodology can define their own threshold for removing low-risk edges from chains, based on the risk appetite and tolerance of the organisation that utilizes the presented method. Each disaster, impact or otherwise hazard can be recalculated according to the overall output, needs and budget of the organisation that aims to reduce risk across complex ICS environments.

3.3 Risk assessment input

The common reference of risk as a cybersecurity assessment metric is the following Eq. 6:

$$\begin{aligned} &Risk = Likelihood \times Impact = \\ &Threat\ probability \times Component\ Vulnerability \times \\ &Impact\ of\ attack \end{aligned} \tag{6}$$

As mentioned in Sect. 3, a graph edge is a derivation of two (2) interconnected nodes, i.e. $Vi \rightarrow Vj$, where Vi (e.g. a sensor) and Vj (e.g. a PLC). A vulnerability in the sensor will affect the decisions taken by the PLC. In this case, the graph edge risk is calculated as: $R_{i,j} = L_{i,j} \times I_{i,j}$, where Li, j is the combination of the threat probability and component vulnerability and $I_{i,j}$ is the impact of the attack. In the case that a security mechanism already exists in a CPSs, this would influence the inherent risks of the asset protected (node) and this would be captured using the existing Likelihood input given to the equations,

To calculate $L_{i,j}$ and $I_{i,j}$, we use the NVD Database, which is a recommendation of international standards [40] to organisations that want to assess the risks of their CPS processes and industrial components. The NVD database utilizes the CVSS 3.1 Severity and Metrics scoring system [7, 17] as the current industry standard to quantitatively capture the essential features of vulnerabilities found in the Common Vulnerabilities and Exposures (CVE) database [22]. The CVE database contains information on publicly known cybersecurity vulnerabilities on products and services, including CPS and industrial components (e.g. CVE-2016-8562 vulnerability on a Siemens SIMATIC CP 1543-1

PLC). The CVSS 3.1 produces numerical scores that reflect each CVE vulnerability's Impact score and Exploitability score, based on three metric groups (i.e. Base, Temporal, and Environmental) and classifies the scores into a 5-level vulnerability severity scale (i.e. Critical, High, Medium, Low, Info). These metrics align with international and industry standards [15, 26] for measuring the vulnerability of a cybersecurity attack. The Exploitability score reflects the ease and technical means by which the vulnerability of a node can be exploited, while the Impact score reflects the direct consequence of a successful exploit to a node. We assign the CVE Exploitability score as the $L_{i,j}$ and the CVE Impact score as the $I_{i,j}$ to calculate the risk for each graph edge.

We utilize the NVD database along with CISA's ICS-CERT Advisories [4] to capture CVE vulnerabilities that affect CPS processes and industrial components. Then we infuse the components of our SCADA model testbed with these CVE vulnerabilities and measure the exploitability and the impact. We measure the exploitability and the impact metrics of common IT nodes by applying the CVSS 3.1 Severity and Metrics scoring system and the exploitability and impact metrics of CPS nodes and their interconnections that are used in railway systems by leveraging the environmental metrics with the algorithm used in [42]. In this way, we provide a more comprehensive assessment by taking into consideration the threats that are posed in a CPS railway environment.

3.4 Centrality graph metrics

Centrality metrics are often used in network models to determine the relative importance of nodes and the influence they have on the overall graph. These metrics provide a useful means of identifying important nodes that could be used for risk mitigation control implementation [28, 37]. In this study, the isomorphic state model graph is analyzed by CPSRA using two centrality metrics to determine the importance of each system state (i.e. a node) within the CPS architecture, particularly in the context of an attack path.

Closeness centrality and Betweenness centrality are two types of centrality metrics that are particularly useful for identifying high-influence nodes in a dependency risk graph. Nodes with high closeness centrality have short average distances from most nodes in the graph, which means they are part of many dependency chains, and sometimes, they may even initiate dependency chains. On the other hand, nodes with high betweenness centrality lie on a high proportion of dependency risk paths, which means that they contribute to multiple risk paths, even if they do not initiate them.

Closeness centrality: Closeness is a measure that captures the relative position of a node within a two-dimensional space using geodesic distances. It quantifies how centrally or peripherally located a node is in comparison to others. This metric can be defined as follows: $C(V_i) = \sum_{\forall v \in V(G)} \delta(v, u)$ where $\delta(v, u)$ is the average shortest path between node v and any other node in the graph.

Betweenness centrality: This metric measures the number of paths in which a node participates. It is defined as $B(V_i) = \sum_{v \neq u \neq i \in V} \delta_{u,i}(v)$ where $\delta_{u,i}(v) = \sigma_{i,j}(v)/\sigma_{i,j}$. $\sigma_{i,j}(v)$ denotes the number of geodesic distances from i to j in which node v is present and $\sigma_{i,j}$ is the number of geodesic distances from i to j in general.

The nodes with high centrality values can be used to pinpoint vulnerabilities with the highest impact in a CPS architecture. They are also useful as cluster generators, which can be used to divide the population of system states into groups with similarities. This is particularly useful for risk assessment and mitigation

4 Methodology

4.1 The three steps of our methodology

1. *Step 1 - Attack graph modelling:* All potential attack paths that may exist in a railway system are plotted on a graph. To automate this process, we leverage established assessment reports on a target CPS, blueprints and enterprise documents. A reduced attack graph is produced by removing low-risk edges.
2. *Step 2 - Graph risk analysis:* All possible n-order attack paths are computed by calculating the reduced attack graph of step 1. Then, the cumulative dependency risk of each attack path is produced and the overall risk of all potential attack scenarios that exist (i.e. the entire network risk) is calculated. Additionally, the attack paths are sorted based on their risk levels and prioritized based on their potential impact on the entire network.
3. *Step 3 - Centrality group formation:* Finally, the algorithm pre-computes the Betweenness and Closeness centrality metric values for each node and produces clusters and rankings of system states.

Figure 1 shows the graphical representation of the methodology flow.

4.2 CPRSA tool

We developed the CPSRA tool to dynamically calculate CPS critical attack paths and analyse CPS state dependencies. The tool utilizes an isomorphic graph representation of a CPS system. This input includes the likelihood (exploitability) and impact values for each node and edge in the graph. For each edge V_i to V_j, the estimated likelihood $L_{i,j}$ and the maximum expected impact $I_{i,j}$ are necessary for static analysis. Given

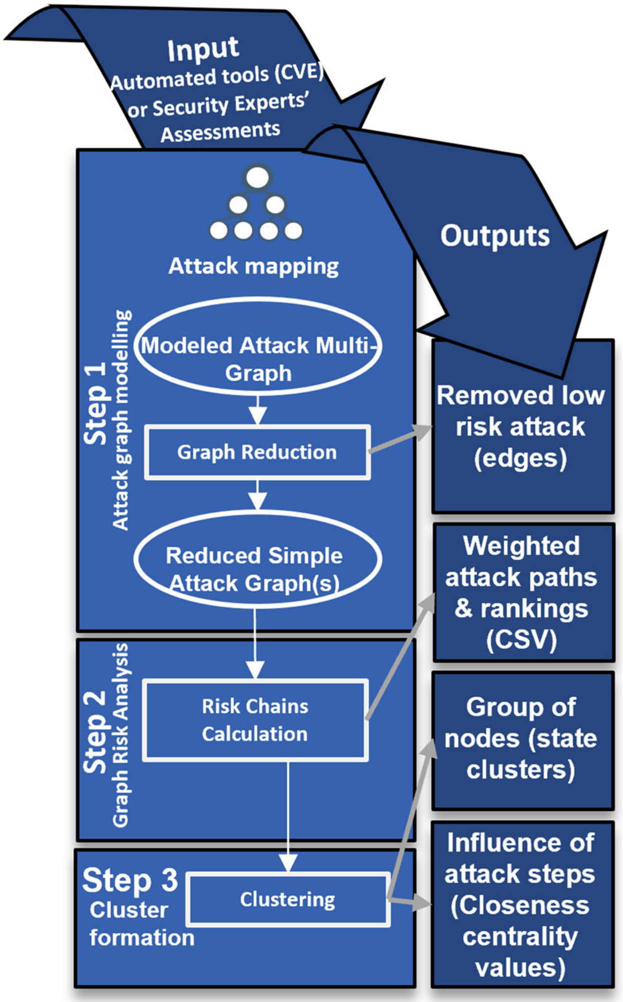
Input
Automated tools (CVE) or Security Experts' Assessments
Outputs
Attack mapping
Step 1
Attack graph modelling
Modeled Attack Multi-Graph
Graph Reduction
Reduced Simple Attack Graph(s)
Step 2
Graph Risk Analysis
Risk Chains Calculation
Step 3
Cluster formation
Clustering
Removed low risk attack (edges)
Weighted attack paths & rankings (CSV)
Group of nodes (state clusters)
Influence of attack steps (Closeness centrality values)

Fig. 1 Methodology flow

the input dependency graph, CPSRA generates the following output:

1. A list of current state dependency routes, limited to a maximum dependency order of 6 (by default).
2. The cumulative Dependency Risk is calculated for each dependency route using Eq. 4 and the expected cumulative Dependency Risk for a specific component is computed using Eq. 5. Dependency paths can be ranked according to their cumulative Risk values.
3. Centrality metrics for each node are produced to measure the influence of attack steps to measure node/state impact.
4. If the risk assessment experts have set a maximum risk limit, CPSRA can also identify all the paths that exceed the risk limit for further mitigation measures.

5 Use case experimentation

5.1 Reference architecture

To test CPSRA we build a model testbed of a real-world large SCADA system based on a proposal railway system zone model comprised of multiple zones and processes from CLS/TS 50701:2021.

The model testbed comprises different areas and different IT and CPS components. Namely, the Enterprise IT area consists of a Database Server, an Application Server, a File Server, a router (Firewall and VPN) and multiple workstations. The SCADA Control Center area consists of a Database Server, a Data Historian, a local switch, an Application Server and control room workstations. The Automation System Head area consists of a central PLC controller and a router that interconnects the three (3) grid zone fields. Each grid zone consists of a router, one (1) or two (2) switches, a Local Server (only in Zone 2), multiple PLCs and sensors. Figure 2 shows a graphical representation of the railway model testbed. Each component is an actual vendor-specific system, and it may be vulnerable to real-world CVE vulnerabilities. Each component's exploitability and impact metrics are retrieved from the NVD Database. An attacker's goal could be to exploit an IT Business workstation in order to gain a foothold and move laterally to gain access to the SCADA Control Center and eventually tamper the PLCs. For example, an attacker could exploit an IT Business workstation via a phishing attack, to gain access to the Enterprise IT network and move laterally to the SCADA Control Center to discover critical PLCs. Then the attacker could gain access to the Central PLC controller and tamper with a PLC that is used in Zone 1 field and is responsible for the signalling and barriers control in several railroad crossings. The potential consequences of this action are significant, as it could result in multiple accidents and the loss of human life. As can be seen, the attacker can use different penetration modes and pathways in this attack graph.

5.2 Input assessment data and tool output

We used several CVE vulnerabilities and respective components to build an indicative scenario, as shown in Table 3. To calculate the risk of each node and edge of our graph we utilize the exploitability (likelihood) and impact metrics of the NVD database.

Tables 1 and 2 present CPSRA's output as calculated over the conceptual model graph created using standard asset

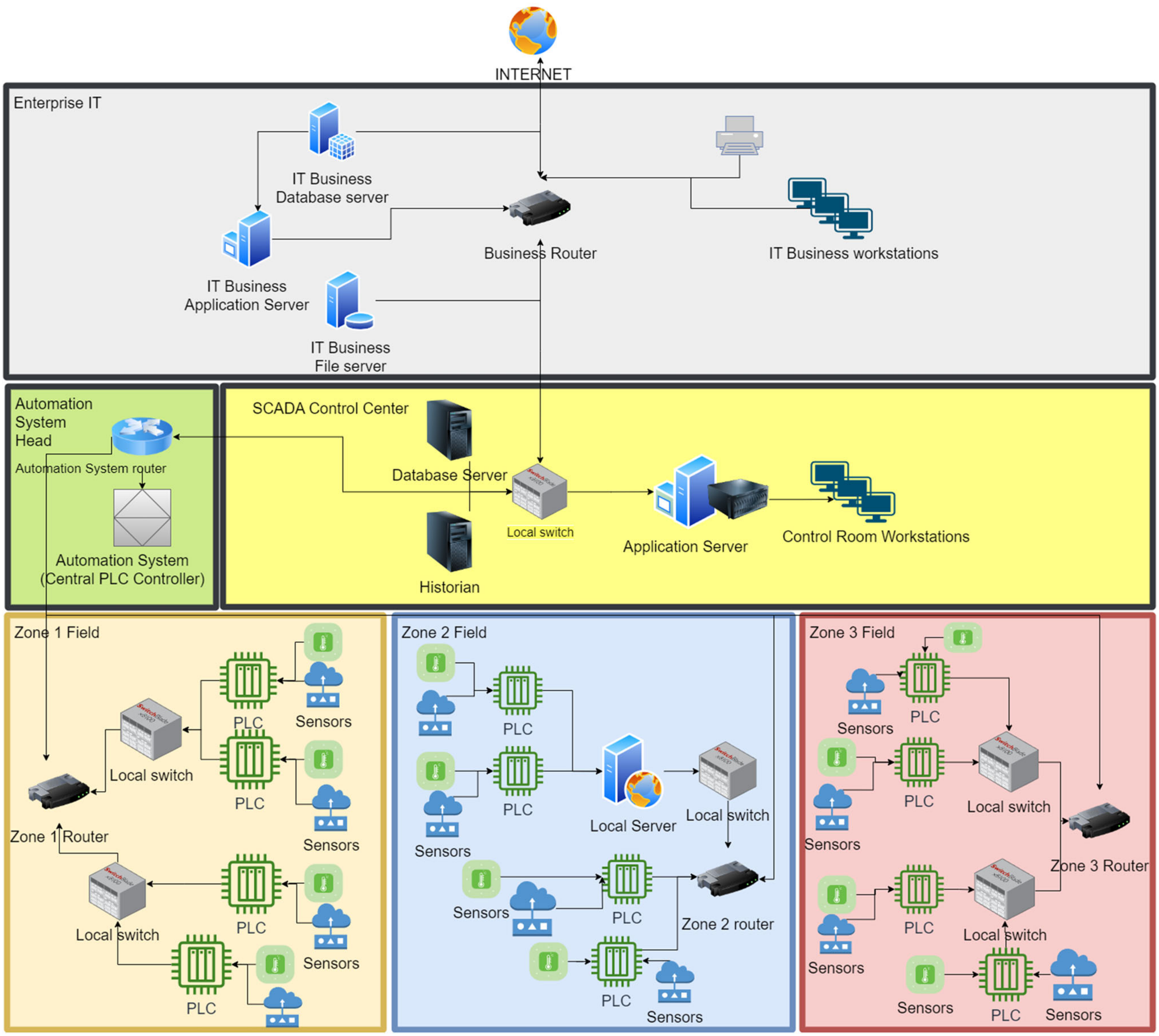

Fig. 2 Model testbed of a railway system

states (see Fig. 3). Table 1 presents the worst-case reduced attack paths based on the risk of each path's final edge and the cumulative dependency risk of each path. Table 2 presents the components that are most influential on the overall CPS architecture and underlying processes.

The graph is created by deriving the states of components under attack and creating an attack-based State graph, where each node is an asset's state after being compromised by an existing vulnerability. We gather requests-responses and information exchange using TCPdump [16] and component vulnerabilities using Nessus vulnerability scanner [30]. Then, we use the outputs' common vulnerability enumeration entries to understand at the states of a compromised component and what would this allow in terms of cascading to other components (i.e. how can the compromised states from a vulnerability exploitation allow to further attack/influence other assets). E.g. by being an RCE, a compromised asset A can allow for scanning and attack on another component B that receives Requests from the given compromised component A.

5.3 Analysis results

The model graph and its edges are not presented in detail due to space limitations. Attack paths that exist on the graph have an order of equal or less than 6 components/states. By analyzing both the attack path risk and centrality metrics for each node, it is apparent that certain edges pose a high risk.

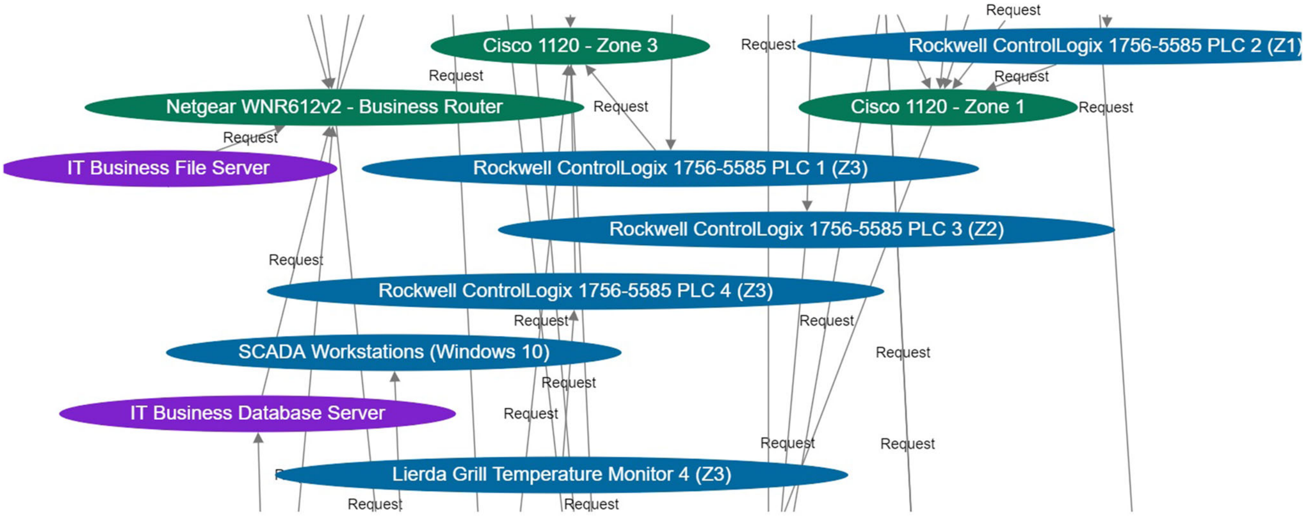
Request
Cisco 1120 - Zone 3
Request
Rockwell ControlLogix 1756-5585 PLC 2 (Z1)
Request
Request
Netgear WNR612v2 - Business Router
Request
Cisco 1120 - Zone 1
Request
Request
IT Business File Server
Rockwell ControlLogix 1756-5585 PLC 1 (Z3)
Rockwell ControlLogix 1756-5585 PLC 3 (Z2)
Request
Rockwell ControlLogix 1756-5585 PLC 4 (Z3)
Request
Request
SCADA Workstations (Windows 10)
Request
Request
IT Business Database Server
Request
Request
Request
Lierda Grill Temperature Monitor 4 (Z3)
Request
Request

Fig. 3 Partial view of the created model graph

Table 1 Highest cumulative dependency-risk and node-risk attack paths

ID	Paths	Node risk	Cum. dependency risk
P1	Lierda Grill Temperature Monitor 1 (Z1) -> Rockwell ControlLogix 1756-5585 PLC 1 (Z1) -> Cisco 1120 - Zone 1 -> Rockwell ControlLogix 1756-5585 PLC (Central Controller)	4.6	29.11
P2	Lierda Grill Temperature Monitor 1 (Z2) -> Rockwell ControlLogix 1756-5585 PLC 1 (Z2) -> Local Data server -> Cisco 1120 - Zone 2 -> Rockwell ControlLogix 1756-5585 PLC (Central Controller)	2.0	26.61
P3	Rockwell ControlLogix 1756-5585 PLC 1 (Z1) -> Cisco 1120 - Zone 1 -> Rockwell ControlLogix 1756-5585 PLC (Central Controller)	4.8	21.15
P4	IT Business Database Server -> Netgear WNR612v2 - Business Router -> SCADA App Server -> SCADA Workstations (Windows 10)	3.8	20.9

Table 2 Highest centrality metrics per component

Node	Betweeness	Closseness
Netgear WNR612v2 - Business Router	**12.50**	0.25
Local Data server	**6.00**	0.17
Cisco 1120 routers	**5.25**	0.11
SCADA App Server	**4.00**	0.08
Rockwell ControlLogix 1756-5585	2.50	**0.50**
Netgear WNR612v2—Business Router	2.07	**0.33**
Cisco 1120 - Zone 2	1.25	**0.28**

Bold values indicate the top values indicating strong node influence in the graph's paths

Table 3 Used CVE vulnerabilities per component for CPSRA scenario tests

Component name	CVE	Base score (Likelihood–Impact)
GE Proficy Historian	CVE-2022-46732	10.0 (1.0–10.0)*
Netgear WNR612v2 Wireless Router	CVE-2023-23110	6.0 (0.6–8.8)
IT Business DB Server	CVE-2008-5416	8.5 (0.85–10)
IT App Server	CVE-2022-34918	4 (0.4–10)
Business Workstations	CVE-2022-21922	7.5 (0.8–9.5)
IT Printer	CVE-2022-23284	3 (0.3–9.5)
Rockwell ControlLogix 1756-5585 PLC (Central Controller)	CVE-2020-12001	6.0 (1.0–6.0)*
Cisco 1120 Connected Grid Router	CVE-2020-3426	9 (0.9–8.8)
Lierda Grill Temperature	CVE-2019-15304	9.5 (0.95–9.5)*

The highest cumulative risk path, (see Table 1, P1) has a cumulative risk score of 26.01. This shows that the overall risk to the entire architecture, should this attack take place, is higher than any other attack due to the inherent impact of an attack on each of these components (based on their actual usefulness in the CPS, and their influence in the represented architecture). The temperature sensor appears to be the best attack vector for the entire CPS since current vulnerabilities allow for attackers to maximize impact on CPS processes; e.g. SCADA workstations could receive false indications (while measurements appear to be accurate) or the attackers could take control of the controller logic, either by tampering with sensory input or reaching and attacking multiple PLCs through CVE-2020-12001.

The betweenness and closeness metrics for each node can affect the risk of each attack path. It is important to notice that the highest metrics relate to different components (see Table 2). If an adversary were to exploit the node with the highest betweenness and closeness metrics, such as the:

1. Rockwell ControlLogix 1756-5585 PLC 4 (highest Closeness metric) or
2. Netgear WNR612v2—Business Router (highest Betweenness metric)

this would respectively allow him to:

1. Access multiple attack paths (maximum available choices for advanced persistent threats), or
2. Reach critical systems relatively easily (few hops).

Another interesting tool output highlights the overall impact of various attack scenarios on the CPS and identifies the "Lierda" field sensor, the GE Proficy Historian Server and "SCADA App Server" components as having the highest overall influence in possible worst-case attack scenarios on the entire CPS (see Table 1 entry vectors in paths). These two, along with the Rockwell ControlLogix 1756-5585 PLC should be prioritized for vulnerability remediation and risk mitigation. While the Cisco 1120 routers are frequently involved in high-risk attacks, they are not key attack steps in all possible attacks on the CPS, which is intuitively true for non-man-in-the-middle attacks (Table 3).

Reaching the SCADA server and workstations through the IT Business Database also was detected as a high-risk scenario, due to chain vulnerabilities found in relevant components. This was made possible due to the lack of isolation between the IT and OT environments. Although intuitively true to any expert, the tool could pinpoint the best attack vector and the easiest path to achieve this.

Tool analysis also concludes that some vulnerabilities are more important in securing specific end services, such as the Historian RCE (CVE-2022-46732) and the Lierda Sensor vulnerability (Admin privilege) (CVE-2019-15304), while others are more important in securing the overall network from as many attacks as possible (e.g. CVE-2020-12001 found in PLCs). Prioritizing vulnerabilities that affect the highest influential states based on Centrality (i.e. Rockwell ControlLogix 1756 5585 PLC 4 (Central Controller), Cisco 1120 routers and the GE Proficy Historian Server) will have the greatest cumulative effect on all possible attack paths.

5.4 Mitigating highest risk chains

The initial step in mitigating risks identified within a chain involves classifying each asset in the chain based on its potential impact in case of a security breach. For each asset type and asset inside a given risk chain, the three cybersecurity objectives—confidentiality, integrity, and availability—are associated with one of three levels of potential impact should there be a breach of security. It is important to remember that for an ICS, availability is generally the greatest concern.

FIPS 200 [24] documents a set of minimum security requirements covering 18 security-related areas with regard to protecting the confidentiality, integrity, and availability of federal information systems and the information processed, stored, and transmitted by those systems. Also, NIST 800-82 [25] includes controls that provide tailored baselines for low-impact, moderate-impact, and high-impact ICS. These tailored baselines can be utilized as starting specifications and recommendations that can be applied to any risk chain and its underlying ICS assets, directly by responsible personnel.

Further tailoring to add or remove controls and control enhancements based on each asset type to reflect organization-specific needs, assumptions, or constraints is recommended and most controls should be implemented at the root of each chain and recalculate its likelihood, along with the overall risk chain to consider further reductions to nodes as progressing from the root node to the tail.

6 Conclusions

In this study, we proposed CPRSA to improve the overall resilience of CPS architectures. The tool accepts as input a CPS model testbed along with real-world vulnerabilities that affect the model's components and produces an isomorphic graph of the model. The tool automatically performs graph algorithms and network analytics on the model and outputs high-risk components and potential sub-attack and subliminal attack paths on the model's attack graph. Based on this analysis, the tool suggests mitigation points for security measure implementation.

This study also addresses the need for automatic cybersecurity architecture assessment in CPS environments, which has been outlined in major NIST publications. Prior to the development of CPSRA, such assessments were done manually without automatic insights into state and vulnerability influences.

In conclusion, the study proves that different components affect the overall CPS differently based on the realized attacks. Sometimes, components that do not include high-impact vulnerabilities (e.g. Rockwell ControlLogix 1756-5585 PLC (Central Controller) and Netgear WNR612v2—Business Router) might have a greater impact due to their positioning inside the CPS. Thus, mitigating the risks on these components will have the greatest impact on the system.

Acknowledgements This work has received funding from the European Union's Horizon Europe programme under Grant Agreement No 101074008.

Funding Open access funding provided by HEAL-Link Greece.

Data Availability All data generated or analyzed during this study are included in this published article.

Declarations

Conflict of interest None of the authors have received a speaker honorarium from any company. All authors declare that none of them has any conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Al Ghazo, A.T., Kumar, R.: Identification of critical-attacks set in an attack-graph. In: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 0716–0722. IEEE (2019)
2. Basiri, A., Behnam, N., De Rooij, R., Hochstein, L., Kosewski, L., Reynolds, J., Rosenthal, C.: Chaos engineering. IEEE Softw. **33**(3), 35–41 (2016)
3. Bodeau, D.J., Graubart, R., Picciotto, J., McQuaid, R.: Cyber resiliency engineering framework. Tech. rep, MITRE CORP BEDFORD MA (2011) https://nvlpubs.nist.gov/nistpubs/cswp/nist.cswp.04162018.pdf
4. CISA (2022) Ics cert advisories. https://www.cisa.gov/uscert/ics/advisories. Accessed Apr 2023
5. Cybersecurity CI. Framework for improving critical infrastructure cybersecurity. https://www.nvlpubsnistgov/nistpubs/CSWP/NISTCSWP (2018)
6. Fawzi, H., Tabuada, P., Diggavi, S.: Secure estimation and control for cyber-physical systems under adversarial attacks. IEEE Trans. Autom. Control **59**(6), 1454–1467 (2014)
7. FIRST. Common vulnerability scoring system v3.1. Available: https://www.first.org/cvss/user-guide (2019). Accessed Apr 2023
8. Francis, R., Bekera, B.: A metric and frameworks for resilience analysis of engineered and infrastructure systems. Reliab. Eng. Syst. Saf. **121**, 90–103 (2014)
9. Haque, M.A., Shetty, S., Krishnappa, B.: Ics-crat: a cyber resilience assessment tool for industrial control systems. In: 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), pp. 273–281. IEEE (2019)
10. IBM. IBM security services 2014 cyber security intelligence index (2014)
11. Ibrahim, A., Bozhinoski, S., Pretschner, A.: Attack graph generation for microservice architecture. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 1235–1242 (2019)
12. IEC. "IEC 62351—Cyber Security Series for the Smart Grid. International Electrotechnical Commission (IEC) (2023)

13. Initiative JTFT. SP 800-39. Managing information security risk: Organization, mission, and information system view. National Institute of Standards & Technology (2011)
14. ISA, IEC. ISA/IEC 62443—Security for industrial automation and control systems. International Society of Automation (ISA) & International Electrotechnical Commission (IEC) (2009)
15. ISO, IEC. ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection—Information security management systems—Requirements. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) (2022)
16. Jacobson, V., McCanne, S., Schwan, K.: Tcpdump: a lightweight packet analyzer. In: Proceedings of the Winter USENIX Conference, USENIX Association (1989)
17. Johnson, P., Lagerström, R., Ekstedt, M., Franke, U.: Can the common vulnerability scoring system be trusted? A Bayesian analysis. IEEE Trans. Dependable Secure Comput. **15**(6), 1002–1015 (2016)
18. Konstantinou, C., Stergiopoulos, G., Parvania, M., Esteves-Verissimo, P.: Chaos engineering for enhanced resilience of cyber-physical systems. In: 2021 Resilience Week (RWS), pp. 1–10. IEEE (2021)
19. Kotzanikolaou, P., Theoharidou, M., Gritzalis, D.: Assessing n-order dependencies between critical infrastructures. Int. J. Crit. Infrastruct. **9**, 93–110 (2013)
20. Li, T., Feng, C., Hankin, C.: Scalable approach to enhancing ICS resilience by network diversity. In: 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp 398–410. IEEE (2020)
21. Luo, Z., Zuo, F., Jiang, Y., Gao, J., Jiao, X., Sun, J.: Polar: function code aware fuzz testing of ICS protocol. ACM Trans. Embed. Comput. Syst. (TECS) **18**(5s), 1–22 (2019)
22. Mitre. Common vulnerabilities and exposures. https://cve.mitre.org/ (2022). Accessed Apr 2023
23. Nateghi, R.: Multi-dimensional infrastructure resilience modeling: an application to hurricane-prone electric power distribution systems. IEEE Access **6**, 13478–13489 (2018)
24. National Institute of Standards and Technology (NIST). Minimum security requirements for federal information and information systems. Federal Information Processing Standards Publication 200. https://csrc.nist.gov/publications/detail/fips/200/final (2006)
25. National Institute of Standards and Technology (NIST).Guide to industrial control systems (ics) security. NIST Special Publication 800-82. https://csrc.nist.gov/publications/detail/sp/800-82/rev-3/final (2021)
26. NIST.Nist special publication 800-30 revision 1-guide for conducting risk assessments (2012)
27. NIST. National vulnerability database. Available: https://nvd.nist.gov/ (2022). Accessed Apr 2023
28. Oldham, S., Fulcher, B., Parkes, L., Arnatkevici, A., Suo, C., Fornito, A.: Consistency and differences between centrality measures across distinct classes of networks. PloS One **14**(7), e0220061 (2019)
29. Paridari, K., O'Mahony, N., Mady, A.E.D., Chabukswar, R., Boubekeur, M., Sandberg, H.: A framework for attack-resilient industrial control systems: attack detection and controller reconfiguration. Proc. IEEE **106**(1), 113–128 (2017)
30. Renaud, T., Fillion, F., Dufresne, L., Bélanger, S.: Nessus: a comprehensive vulnerability scanning tool. J. Netw. Syst. Manag. **13**(2), 193–212 (2005)
31. Rinaldi, S.M., Peerenboom, J.P., Kelly, T.K.: Identifying, understanding, and analyzing critical infrastructure interdependencies. IEEE Control Syst. Mag. **21**(6), 11–25 (2001)
32. Ross, R., Pillitteri, V., Graubart, R., Bodeau, D., McQuaid, R.: Developing cyber resilient systems: a systems security engineering approach. Tech. rep, National Institute of Standards and Technology (2019)
33. Ross, R., Pillitteri, V., Guissanie, G., Wagner, R., Graubart, R., Bodeau, D.: Enhanced security requirements for protecting controlled unclassified information: A supplement to nist special publication 800–171 (final public draft). Tech. rep, National Institute of Standards and Technology (2020)
34. SCADAfence. The 2022 state of operational technology survey results (2022)
35. Schenato, L., Sinopoli, B., Franceschetti, M., Poolla, K., Sastry, S.S.: Foundations of control and estimation over lossy networks. Proc. IEEE **95**(1), 163–187 (2007)
36. Sterbenz, J.P., Cetinkaya, E.K., Hameed, M.A., Jabbar, A., Rohrer, J.P.: Modelling and analysis of network resilience. In: 2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011), pp 1–10. IEEE (2011)
37. Stergiopoulos, G., Kotzanikolaou, P., Theocharidou, M., Gritzalis, D.: Risk mitigation strategies for critical infrastructures based on graph centrality analysis. Int. J. Crit. Infrastruct. Prot. **10**, 34–44 (2015)
38. Stergiopoulos, G., Kotzanikolaou, P., Theocharidou, M., Lykou, G., Gritzalis, D.: Time-based critical infrastructure dependency analysis for large-scale and cross-sectoral failures. Int. J. Crit. Infrastruct. Prot. **12**, 46–60 (2016)
39. Stergiopoulos, G., Dedousis, P., Gritzalis, D.: Automatic analysis of attack graphs for risk mitigation and prioritization on large-scale and complex networks in industry 4.0. Int. J. Inf. Secur. (2022). https://doi.org/10.1007/s10207-020-00533-4
40. Stouffer, K., Pease, M., Tang, C., Zimmerman, T., Pillitteri, V., Lightman, S.: Guide to operational technology (ot) security. Tech. rep, National Institute of Standards and Technology (2022)
41. Tierney, K., Bruneau, M.: Conceptualizing and measuring resilience: a key to disaster loss reduction. TR news (250) (2007)
42. Ur-Rehman, A., Gondal, I., Kamruzzaman, J., Jolfaei, A.: Vulnerability modelling for hybrid industrial control system networks. J. Grid Comput. **18**, 863–878 (2020)
43. Verissimo, P., Correia, M., Neves, N.F., Sousa, P.: Intrusion-resilient middleware design and validation. Inf. Assur. Secur. Priv. Serv. **4**, 615–678 (2009)
44. Wang, W., Chen, L., Han, L., Zhou, Z., Xia, Z., Chen, X.: Vulnerability assessment for ICS system based on zero-day attack graph. In: 2020 International Conference on Intelligent Computing, pp. 1–5. Automation and Systems (ICICAS), IEEE (2020)
45. Zhang, M., Chen, C.Y., Kao, B.C., Qamsane, Y., Shao, Y., Lin, Y., Shi, E., Mohan, S., Barton, K., Moyne, J., et al.: Towards automated safety vetting of plc code in real-world plants. In: 2019 IEEE Symposium on Security and Privacy (SP), pp 522–538. IEEE (2019)
46. Zonouz, S., Davis, C.M., Davis, K.R., Berthier, R., Bobba, R.B., Sanders, W.H.: Socca: a security-oriented cyber-physical contingency analysis in power infrastructures. IEEE Trans. Smart Grid **5**(1), 3–13 (2013)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.