# Security Enumerations for Cyber-Physical Systems

Daniel Schlette, Florian Menges, Thomas Baumer, Günther Pernul

▶ **To cite this version:**

Daniel Schlette, Florian Menges, Thomas Baumer, Günther Pernul. Security Enumerations for Cyber-Physical Systems. 34th IFIP Annual Conference on Data and Applications Security and Privacy (DBSec), Jun 2020, Regensburg, Germany. pp.64-76, 10.1007/978-3-030-49669-2_4 . hal-03243630

**HAL Id: hal-03243630**
**https://inria.hal.science/hal-03243630**

Submitted on 31 May 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

# Security Enumerations for Cyber-Physical Systems

Daniel Schlette* , Florian Menges , Thomas Baumer , Günther Pernul

University of Regensburg, 93053 Regensburg, Germany
{firstname.lastname}@ur.de
*corresponding author

**Abstract.** *Enumerations constitute a pivotal element of Cyber Threat Intelligence (CTI). References to enumerated artifacts support a universal understanding and integrate threat information. While traditional IT systems and vulnerabilities are covered by security enumerations, this does not apply to Cyber-Physical Systems (CPS). In particular, complexity and interdependencies of components within these systems demand for an extension of current enumerations. Taking on a CPS security management perspective this work identifies deficiencies within the Common Platform Enumeration (CPE) and the Common Vulnerabilities and Exposures (CVE) enumeration. Models for CPS are thus proposed to cover comprehensiveness and usability. A prototype is used to evaluate the feasibility by demonstrating key features of security enumerations for CPS.*

## 1 Motivation

At present we are experiencing an encompassing transition of our daily life and environment caused by the availability of technology and the efficient processing of information. Formerly separate domains such as physical processes and IT systems become interconnected and can now be remotely controlled. The resulting Cyber-Physical Systems (CPS) allow for exciting new applications. Since this development is accompanied by a continuous increase in complexity, it is also an essential factor for the emergence of many vulnerabilities of CPS. Even for security experts it is a challenging task to keep track of all vulnerabilities that may cause an issue for their organization and require quick countermeasures.

It is evident that a reduction of the given complexity is necessary to solve this issue. Security enumerations are suitable to make complexity manageable as they cover various Cyber Threat Intelligence (CTI) artifacts such as platforms, vulnerabilities or even natural hazards. In general, they are designed to enhance the information flow between organizations by setting up a common and usable reference for considered objects. CTI makes use of security enumerations not only to describe cyber attacks but also to share and collaboratively improve valuable threat information via dedicated platforms and data formats [12,14].

Two of the most notable enumerations are the Common Vulnerabilities and Exposures (CVE) enumeration and the Common Platform Enumeration (CPE). They are, for example, used to describe different properties of the TRITON

malware which we will use in our case study. More specifically, *CVE-2018-7522* provides a standardized identifier, an additional description and further references about the leveraged TRITON vulnerability found in Cyber-Physical Systems. Besides, the firmware component "Schneider Electric - Triconex Tricon MP 3008" affected by the aforementioned CVE entry is encoded as CPE name *cpe:2.3:o:schneider-electric:triconex_tricon_mp_3008_firmware:10.0*. This name covers key characteristics of the platform including vendor, product and version.

While CVE and CPE provide guidance for communicating about vulnerabilities and platforms, the US National Vulnerability Database (NVD) goes one step further. By collecting and linking entries of both security enumerations a connected CPE and CVE search engine is realized. Ultimately, this search engine allows to check whether a vulnerability affects a specific device or vice versa.

However, in appreciation for the NVD and its community there are still improvements targeting complexity as well as the search engine possible. Focusing on CPS, one issue while working with the NVD is the requirement imposed on the user to know the CPE names of her own assets prior to searching for related vulnerabilities. The complexity and heterogeneity of CPS make a comprehensive overview of the deployed components already a challenging task [20,19]. Additionally, CPS introduce novel components for the enumerations such as Supervisory Control And Data Acquisition (SCADA) systems, Programmable Logic Controllers (PLC), actuators and sensors which need to be managed alongside existing components. Security management of CPS also requires highly specific knowledge about CPS as well as cyber security which combined may constitute an obstacle to recognize vulnerabilities and to act quickly according to them.

This culminates in the three following research questions tackled in this paper addressing the reduction of complexity within security management of CPS:

1. How can the overview of numerous and heterogeneous CPS components in a given organization be improved?
2. How can novel classes of CPS components be added to CPE?
3. How can usability of CPE and CVE for users without specific domain knowledge be enhanced?

The remainder of this paper proceeds as follows: First a review of background information on CPS and security enumerations is given in Section 2. We then describe our conceptual approach and perform a detailed analysis of CPS characteristics in Section 3. Deficiencies found in the two security enumerations CPE and CVE lead towards extensions proposed in our concept. Our prototypical implementation is demonstrated based on a use case in section 4. Subsequently, we give an overview on related work in the areas of CPS and enumerations in Section 5 and conclude the paper in Section 6.

## 2 Background

In this section we briefly introduce Cyber-Physical Systems, Common Platform Enumeration (CPE) as well as Common Vulnerabilities and Exposures (CVE).

### 2.1 Cyber-Physical Systems (CPS)

Digital transformation has reached areas from industrial production to medical applications and household sectors. Accordingly, the concept of CPS is applied to describe the deep integration of physical elements into computing and control processes of the cyber domain [10]. The cyber domain categorizes traditional IT, such as servers or workstations, while the physical domain describes physical entities, such as mechanical or chemical processes and components. CPS also cover advanced functionalities and scenarios based on spatial proximity, such as real-time data processing or feedback loops. While these characteristics are desirable from a functionality perspective they introduce complexity as it is often the case for highly connected systems containing multiple components [1].

### 2.2 Common Platform Enumeration (CPE)

In the context of cyber security, enumerations define a naming schema for standardization purposes. They provide unique names to cyber threat intelligence (CTI) artifacts and support, for instance, the identification of IT assets, vulnerabilities, attack patterns as well as quality aspects [21].

The Common Platform Enumeration (CPE) describes IT assets and is maintained by the National Institute of Standards and Technology (NIST). It fulfills two main objectives. First, it allows to assign unique names to classes of applications, operating systems and hardware devices [3]. Secondly, it provides matching mechanisms, including details on how to search and compare CPE names [18].

The naming specification includes three distinct naming methods. A given CPE name is either described as *well-formed CPE name (WFN)*, *formatted string (FS)* or *Uniform Resource Identifier (URI)*, allowing to define product classes [3]. While WFN is an abstract set of attribute-value pairs, both FS and URI names are machine-readable encodings [3]. Listing 1 shows the structure and the individual components of a FS encoding. The values for the listed attributes are implemented as strings. In case values are unspecified (ANY) or there is no meaningful value (NA) these are encoded respectively.

```
cpe:2.3:{part}:{vendor}:{product}:{version}:
    {update}:{edition}:{language}:{sw_edition}:
    {target_sw}:{target_hw}:{other}
```

**Listing 1.** CPE – FS name structure

CPE is in particular useful to link classes of IT assets to vulnerabilities. It is easy to infer that based on CPE entries, context relevant threat information can be retrieved and information security workflows realized. As a result CPE and CVE are oftentimes applied together [23]. Decision making, the creation of information security policies adapted to the prevalent IT infrastructure and the configuration of platforms are additional use case scenarios of CPE.

### 2.3 Common Vulnerabilities and Exposures (CVE)

Enumerations not only focus on platforms found in CPE but also target security artifacts directly. The Common Vulnerabilities and Exposures (CVE) enumeration describes vulnerabilities that may lead to exploitation of systems or violations of security policies[1]. Central element to the CVE enumeration are CVE entries, which serve as unique, common identifiers for publicly known information security vulnerabilities. Essentially, each CVE entry consists of the three main components: **CVE ID**, **description** and **references**.

However, these only show an excerpt of the CVE data model capabilities. The CVE automation working group maintains a repository[2] with the specification of a CVE JavaScript Object Notation (JSON) schema with additional elements. Figure 1 provides a simplified overview of the CVE JSON 4.0 data model. The hierarchy of CVE JSON elements is thereby indicated by different tones of gray.

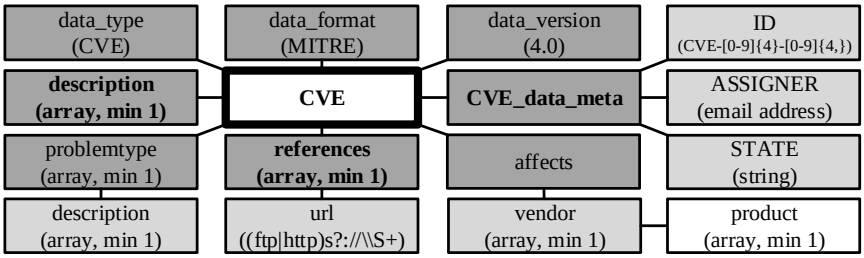


**Figure 1.** CVE simplified JSON data schema

The CVE features are integrated in security products based on the CVE data format. CVE entries are also enhanced by metrics like the Common Vulnerability Scoring System (CVSS). Linking CVE entries to CPE entries can further be of value to organizations trying to protect their IT assets.

## 3 Conceptual Approach

To introduce essential characteristics of CPS into security enumerations our work follows a conceptual approach. First, based on formal CPS specifications we examine common characteristics to derive relevant requirements and deficiencies within existing security enumerations. Then, extensions to the data models of CPE and CVE are proposed. Here, the perspective of a domain expert is incorporated to put focus on complexity and usability aspects. Finally, the approach is implemented in a prototypical search engine to evaluate the previous findings.

### 3.1 Requirements

Our twofold requirements analysis addresses CPS and security issues in CPS as well as the two security enumerations CPE and CVE in the following.

[1] https://cve.mitre.org/about/terminology.html
[2] https://github.com/CVEProject/automation-working-group

**Assumption 1 (Secure CPS)** *The security of CPS is a positive property.*

Common characteristics of CPS go beyond of traditional IT systems. CPS leverage reactive computation and concurrency. Feedback control via designated controllers, real-time computation and utilization in safety-critical scenarios are part of CPS [1]. The characteristics are realized with sensing, actuating and control components and lead to interdependencies [19]. However, this opens various attack vectors. As prior analysis shows, attacks on nuclear facilities and other critical infrastructures do occur and can have far-reaching consequences [13].

Besides that, most CPS include a multitude of different components like sensors and actuators on the field level. Additionally, Programmable Logic Controllers (PLC) are included as direct control elements. SCADA systems provide another control layer. CPS are thus best described as *systems of systems*.

There are also diverse application areas for CPS, such as energy systems, healthcare and transportation [9]. While security experts know about the application scenarios of CPS, they have much less knowledge about procedures inside CPS. In consequence, these CPS are black boxes from a security perspective.

CPS security must also consider different attack vectors due to various interfaces, operating systems and protocols [11]. Since security assessments and measures require a thorough understanding, formal attack detection, security testing and threat modelling [6,2] have received the researchers' attention. Although, guidelines and tools for CPS security management exist [22], usability for component and vulnerability identification can be improved.

**Assumption 2 (Enumerations)** *Security enumerations support security management through identification and searchability of artifacts.*

In an organizational setting, information security workflows are aligned to structured data formats. Security management based on CPE is of great importance in the asset management domain and permits risk analyses. CVE further allows to pinpoint security flaws and vulnerabilities of managed IT assets.

Mapping CPS characteristics to the data models of CPE and CVE reveals a number of deficiencies. While there are security products (e.g. NVD) that combine and link CVE and CPE data there is no properly maintained direct reference. This generic deficiency is further accompanied by deficiencies broadly categorized as *component-based* and *system-based*.

**Component-based deficiencies:** Currently, CPE supports traditional IT assets but CPS specifics are missing. This is mainly because CPS components contain specific programming languages or protocols. With a focus on multiple elements contained within CPS the CPE data model also neglects various technical aspects. The embedded nature of components and their interfaces are aspects left aside. As these properties implicate possible attack vectors and allow the identification of CPS, integration into CPE is deemed necessary.

**System-based deficiencies:** From a system perspective CPS represent a new concept of highly-connected components. Grouping multiple components described by their CPE names and linking related vulnerabilities is not supported

by CPE and CVE data models. System-based deficiencies are thus related to the usability of the enumerations by security analysts with minor CPS knowledge.

Combining the assumptions it can be concluded that there is a need to support a more comprehensive presentation of the CPS, as this is key to enable the search for vulnerabilities. Our extensions to CPE and CVE aim to foster a better understanding of CPS and a reduction of complexity. Integrating enumerations and making security of CPS manageable is a first step towards secure CPS.

### 3.2 Conceptual Meta Model

Our proposed model is built upon the findings of the requirements phase and describes a formal structure and relationships between entities of CPE and CVE. In our enhancements we explicitly take into account compatibility with earlier versions. To achieve this, new attributes are added while the existing ones remain unchanged. Following the identified CPS characteristics as well as CPE and CVE deficiencies we group extensions into four categories. The applied naming convention of these categories documents central features that are addressed by our proposal. Extensions relating to CPS characteristics missing in CPE are specified within *technically exhaustive security enumerations*. Bundling CPS components leads to *recursive security enumerations*. *Application-oriented security enumerations* include extensions with usability focus. Last but not least, *coupled security enumerations* address extensions connecting CPE and CVE directly.

**Technically exhaustive security enumerations** streamline representation of the various components within CPS. We include new elementary attributes and change attribute values as shown in Table 1 to provide a detailed technical description. In this context, the CPS architecture hints at the importance and embedded nature (inseparable software and hardware) of some CPS components [11]. Thus, we introduce *embedded* as a new possible attribute value that covers components within the *part* attribute of CPE names. Interdependencies of CPS are targeted by the new attributes *protocol* and *interface* added to CPE as these allow to express means of communication and connection. Applications used in CPS oftentimes rely on specific programming languages. CPE is extended by a *programming language* attribute to cover this CPS property.

**Recursive security enumerations** address the *system of systems* concept which is inherent to CPS. We therefore propose the extension of CPE with an additional *CPS Bundle* entity type. As a result, multiple connected components of a CPS can be referenced within the model and build a self-contained unit. The attributes of a *CPS Bundle* reflect the recursive nature of CPS and are specified as *ID*, *description* and *references* shown in Table 1. Due to the fact, that CPS are different and contextually dependent we envision a customization option to describe CPS with a *CPS Bundle*. It is thus possible to provide a brief description of a CPS according to a given situation. The purpose of the *description* attribute is to facilitate a first understanding of these systems on a higher level of abstraction. Also, recursive reference to another CPS Bundle in the *references* attribute is possible and supports hierarchical structuring.

| | Attribute | Description | Examples |
|---|---|---|---|
| CPE Extension | part | The part attribute shall have a new value: “e” - embedded component | e |
| | sector | The sector attribute should capture areas where systems are typically deployed | energy; healthcare; transportation |
| | capability | The capability attribute should capture physical functionalities | pressure; viscosity; acceleration |
| | protocol | The protocol attribute should capture means of communication | Profinet; OPC-UA; DNP3; Modbus; IP |
| | programming language | The programming language attribute should capture notations for computer programs | Ladder Diagram; C; Java; Instruction List |
| | interface | The interface attribute should capture means of connection | USB; PCI; SCSI; SATA; RJ-45 |
| | **Attribute** | **Description** | |
| CPS Bundle | ID | The ID attribute should capture unique IDs for a CPS bundle | |
| | description | The description attribute should capture essential CPS information | |
| | references | The reference attribute should capture CPE names of the CPS components and different CPS bundle IDs | |
| CVE Ext. | CVE_ID | The CVE_ID attribute should capture assigned CVE IDs | |
| | description | The description attribute should capture vulnerability information | |
| | references | The references attribute should capture external data sources describing the given vulnerability as well as CPE name representations | |

**Table 1.** Conceptual meta model entity extensions

**Application-oriented security enumerations** include requirements imposed by security experts without detailed knowledge about CPS. Extensions to CPE with focus on application areas of CPS are aimed to make their security manageable regardless of technical background. Our approach captures usability from a security management perspective through the new *sector* and *capability* attributes shown in Table 1. We thereby assume that some knowledge about CPS in the form of capabilities or application area is present at all times.

**Coupled security enumerations** introduce a closer tie between CVE and CPE. Focusing on the JSON data schema for CVE we propose an extension for the attribute values captured with the *references* attribute. Besides references to external data sources documenting the vulnerability, the *references* attribute is able to capture CPE names shown in Table 1. In consequence, CVE and CPE are coupled and vulnerabilities affecting a given CPS can be retrieved more easily.

The meta model for our CPS security enumerations search engine is shown in Figure 2. First, to cover *coupled security enumerations* multiple vulnerabilities (CVE) can be associated with an IT asset (CPE). In addition, Figure 2 describes *recursive security enumerations* as CPE entities can be part of an individual CPS Bundle entity. Any CPS bundle can also contain other CPS bundles.

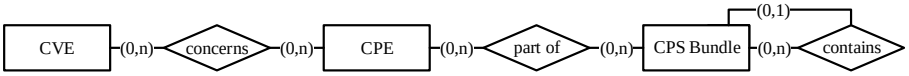


**Figure 2.** Meta model of CPS security enumerations search engine

At last, note that migration of data previously described with either CPE or CVE and integration with our model is feasible. Also, CPE entries do not need to contain values for all (new) attributes. Our model is explicitly designed to capture CPS, as these are currently neglected by security enumerations.

## 4 Use Case

In this section we outline a use case to evaluate our concept. A CPS security enumerations search engine analogous to the generic NVD is central to security management in an organizational setting. Related security processes and common associations are schematically depicted in Figure 3. In general, a security enumerations search engine proves viable by allowing vulnerabilities and IT assets to be identified and eventually patched. The reduction of complexity and improved usability for security management experts without detailed CPS knowledge is the aim of our concept and prototype. Ultimately, if fulfilled this can lead to a better security posture. In the following, the applied technology of our CPS security enumerations search engine and a case study are presented.

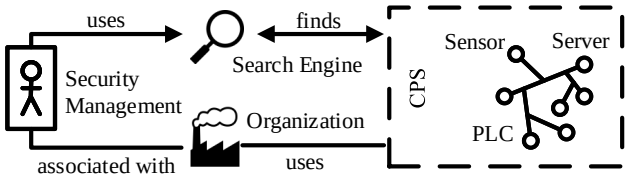


**Figure 3.** Simplified use case for a CPS security enumerations search engine

### 4.1 Case Study

To demonstrate our concept we refer to the TRITON malware used for attacks on oil and gas production facilities in 2017. The malware manipulates *Safety Instrumented Systems (SIS)* that aim to prevent incidents causing severe damage to assets, environment or even humans. Instead of extreme and uncontrollable events, SIS initiate a safe shut down of industrial processes as a last line of automated defence [5]. Since the attackers behind TRITON were able to interact with SIS controllers, there was a risk of unforeseeable disasters, which the SIS was supposed to prevent. Although, TRITON has not yet destroyed physical assets, it has halted production causing financial losses.

Considering the enumerations CPE and CVE, two aspects of the TRITON malware are of relevance. First, it concerns both hardware and operating systems in multiple versions. A standardized description with CPE names is thus an

necessity to avoid miscommunication. Despite of its significance for industrial facilities, relevant elements of the TRITON attacks are not yet properly described by CPE and CVE. While NVD lists related CVE entries and mentions affected components[3], CPE names cannot be found in the dictionary.

In our concept for CPS security enumerations we provide relevant extensions to improve the representation of e.g. cyber attacks using the TRITON malware. We allow the grouping of multiple CPS components within a CPS Bundle as our concept includes *recursive security enumerations*. Despite the fact, that the malware itself mainly targets the "Triconex Tricon MP 3008" controller and its firmware, other CPS components are also affected. To conduct their attack, the attackers leveraged further vulnerabilities of networks, operating systems and workstations prior to infecting the SIS. A comprehensive representation capturing these additional elements is supported by the CPS Bundle and part of the *recursive security enumerations* we designed. Table 2 shows an exemplary **Petrochemical CPS Bundle** related to TRITON with multiple components.

| ID | Description | References (abbreviated) |
|---|---|---|
| 1 | **Petrochemical CPS:** Interconnected components deployed in an industrial setting to refine oil and gas | `cpe:2.3:h:schneider-electric:triconex_tricon_mp_3008 [...];` `cpe:2.3:e:weatherford:maximizer [...];` `[...]` |

**Table 2.** Exemplary CPS composed of multiple components

An extension to CPE names addressing technical details of CPS components is part of our concept. We propose *technical exhaustive security enumerations* that cover the "NCM" network modules of hardware affected by the TRITON malware. Listing 2 shows an exemplary CPE name adhering to the extended model. Furthermore, our model can recognize embedded components like SIS.

```
cpe:2.3:h:schneider-electric:triconex_tricon_mp_3008:*:*:*:*:*:*:*:*
     :Oil_Gas_Production:Safety_Instrument:*:*:NCM
```

**Listing 2.** Exemplary extended CPE name

*Application oriented security enumerations* ensure complexity reduction and usability through CPS properties known to security experts without detailed CPS knowledge. This is achieved by capturing the "oil and gas production" sector as well as "safety instrument" capabilities within a CPE name as shown in Listing 2. These rather generic CPS properties lead to further described information about an entire petrochemical CPS and potential vulnerabilities.

When CPE and CVE are not properly linked it is an impediment for usability and effective security workflows. Integration of both security enumerations is the focal point of *coupled security enumerations*. Within our CPS security enumerations search engine we provide the option to relate entries of CPE and CVE. E.g., the missing link between the "Triconex Tricon MP 3008" firmware and CVE-2018-7522 is established and persisted in the database.

[3] https://nvd.nist.gov/vuln/detail/CVE-2018-7522

### 4.2 Prototypical Implementation

In order to demonstrate the practical applicability of our concept, we have implemented a prototypical CPS search engine for our enumeration concept. The source code of the prototype is available online[4]. It consists of two main components. The conceptual model is implemented using a MySQL database and the application was created with JavaEE 6. The database contains *CPE*, *CVE* and *CPS Bundle* as central entity tables. The references within the database and the functional scope of the application are based on the NVD and extend it to the components presented in this work. As this is a prototype application, additional tools are available for editing the data inventory and creating new CPE, CVE and CPS Bundle objects. The application also offers functionalities to search for CPE and CVE entries and to display the available links between them. The search for CPS bundles also displays the relationships within the bundles.

The data building the basis for our prototypical implementation reflects the state of CPE and CVE from February 2020. In addition, we provide two small sample CPS containing multiple components as exemplary data for CPS bundles.

## 5 Related Work

Information security and CTI [8,12,14] use security enumerations to describe relevant artifacts [15,25]. To the best of our knowledge, there is no academic literature on extending security enumerations although security enumerations evolved and raised their version numbers. It is therefore reasonable to assume, that extensions to security enumerations are driven by dedicated communities.

A multitude of work focuses on CPS due to their prominent role in critical infrastructures [1,10]. Related work on security of CPS approached the topic through the Internet of Things [20]. From there on, the various different areas of security are applied to CPS research. While a number of surveys and overview articles aim to cover CPS security at large [7], attack detection [16], vulnerability analysis [4,24] and formal approaches [26,6] are extensively considered.

Work about both, security enumerations and CPS, is rare. Closest to our research is the work by Upadhyay and Sampalli [24]. It discusses vulnerabilities within SCADA systems, highlighting the necessity of awareness about vulnerabilities within SCADA software and protocols. Here, a strong focus is placed on a review of existing vulnerabilities partially described by CVE. Similar, Nicholson et al. [17] point to unpatched software as a major flaw in SCADA systems.

Maidl et al. [11] provide interesting research results by defining a pattern to structure CPS and classifying the individual components. In addition, the authors outline security considerations about attack vectors for these systems.

McLaughlin et al. [13] present a methodology for security assessment of industrial control systems. They are characterizing parts and features of these systems as a starting point for a more comprehensive description with security enumerations. In a more general perspective Takahashi et al. [23] show the use of security enumerations for security management.

[4] https://github.com/tarnschaf/cyberphysical

## 6 Conclusion and Future Work

With our work we aim to make security of CPS more accessible for security experts. Our analysis showed that CPS and their interdependent components are not yet completely covered by security enumerations. To remediate the identified deficiencies, we propose an extension of CPE and CVE enabling a comprehensive description of CPS. Effective security management also relies on the integration of data from CPE and CVE to attribute vulnerabilities to the affected IT assets.

The meta model we propose extends the security enumerations and provides an overview of the numerous and heterogeneous CPS components. Our search engine realizes the adaptation of CPS to a organization setting and addresses the **1st research question** outlined in Section 1 of this paper. Our work extends the CPE data model with technical features to capture the embedded nature of CPS components. This allows us to address the **2nd research question**. Comprehensiveness and usability aspects relevant for security management are incorporated in our extended CPE. To respond to the **3rd research question** we introduce sector and capability attributes lowering entry knowledge to CPS. The concept is evaluated through a prototype using TRITON as use case.

Although, our work's results are a first step towards security enumerations for CPS several topics demanding further research remain.

First, future work should address the alignment with other standardization efforts and products. While we propose a concept for a CPS security enumerations search engine the usage may be within existing products such as NVD. Other standards for IT asset identification and their integration or conversion to CPE should also be considered. Moreover, management processes related to an IT asset inventory and vulnerabilities will be future points of reference.

Second, further extensions to our proposed meta model might become necessary due to additional user requirements and CPS development. It will be favourable to conduct a user study to determine precise requirements of security management experts beyond the ones described in academic literature. The results can then be used to trigger further improvements and might either culminate in a stand-alone security product or lead towards additional modifications.

A third topic of interest is the collection of data for CPS Bundles. Gathering and maintaining the data can include vendors and operators of CPS. The model can also be complemented by predefined vocabularies for specific attributes to avoid ambiguity and ease usability.

## References

1. Alur, R.: Principles of Cyber-Physical Systems. The MIT Press (2015)
2. Caselli, M., Kargl, F.: A security assessment methodology for critical infrastructures. In: Int. Conf. on Critical Inf. Infrastructures Sec. pp. 332–343 (2014)
3. Cheikes, B.A., Waltermire, D., Scarfone, K.: Common Platform Enumeration: Naming Specification Version 2.3. NIST, Maryland, USA (2011)
4. Coffey, K., Smith, R., Maglaras, L., Janicke, H.: Vulnerability analysis of network scanning on scada systems. Security and Communication Networks 2018 (2018)

5. Di Pinto, A.A., Dragoni, Y., Carcano, A.: Triton: The first ICS cyber attack on safety instrument systems. In: Proc. Black Hat USA. pp. 1–26 (2018)
6. Fernandez, E.B.: Threat Modeling in Cyber-Physical Systems. In: 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing. pp. 448–453 (2016)
7. Humayed, A., Lin, J., Li, F., Luo, B.: Cyber-Physical Systems Security—A Survey. IEEE Internet of Things Journal 4(6), 1802–1831 (2017)
8. Kampanakis, P.: Security Automation and Threat Information-Sharing Options. IEEE Security & Privacy 12(5), 42–51 (2014)
9. Khaitan, S.K., McCalley, J.D.: Design Techniques and Applications of Cyberphysical Systems: A Survey. IEEE Systems Journal 9(2), 350–365 (2014)
10. Lee, E.A.: Cyber-physical systems-are computing foundations adequate. In: Position paper for NSF workshop on cyber-physical systems. vol. 2, pp. 1–9 (2006)
11. Maidl, M., Wirtz, R., Zhao, T., Heisel, M., Wagner, M.: Pattern-based modeling of cyber-physical systems for analyzing security. In: Proceedings of the 24th European Conference on Pattern Languages of Programs. pp. 1–10 (2019)
12. Mavroeidis, V., Bromander, S.: Cyber threat intelligence model: an evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In: Europ. Intelligence and Security Informatics Conference (EISIC). pp. 91–98 (2017)
13. McLaughlin, S., Konstantinou, C., Wang, X., Davi, L., Sadeghi, A.R., Maniatakos, M., Karri, R.: The cybersecurity landscape in industrial control systems. Proceedings of the IEEE 104(5), 1039–1057 (2016)
14. Menges, F., Pernul, G.: A comparative analysis of incident reporting formats. Computers & Security 73, 87–101 (2018)
15. Menges, F., Sperl, C., Pernul, G.: Unifying cyber threat intelligence. In: Trust, Privacy and Security in Digital Business. pp. 161–175. Springer (2019)
16. Mitchell, R., Chen, I.R.: A survey of intrusion detection techniques for cyber-physical systems. ACM Comput. Surv. 46(4) (2014)
17. Nicholson, A., Webber, S., Dyer, S., Patel, T., Janicke, H.: Scada security in the light of cyber-warfare. Computers & Security 31(4), 418–436 (2012)
18. Parmelee, M.C., Booth, H., Waltermire, D., Scarfone, K.: Common Platform Enumeration: Name Matching Specification Version 2.3. NIST, Maryland, USA (2011)
19. Rinaldi, S.M., Peerenboom, J.P., Kelly, T.K.: Identifying, understanding, and analyzing critical infrastructure interdependencies. IEEE Control Systems 21(6) (2001)
20. Roman, R., Zhou, J., Lopez, J.: On the features and challenges of security and privacy in distributed internet of things. Computer Networks 57(10) (2013)
21. Schlette, D., Böhm, F., Caselli, M., Günther, P.: Measuring and visualizing cyber-threat intelligence quality. Int. Journal of Information Security 19(2) (2020)
22. Stouffer, K., Falco, J., Scarfone, K.: Guide to industrial control systems (ICS) security. NIST special publication 800(82) (2014)
23. Takahashi, T., Miyamoto, D., Nakao, K.: Toward Automated Vulnerability Monitoring using Open Information and Standardized Tools. In: 2016 IEEE Int. Conf. on Pervasive Comp. and Comm. Workshops (PerCom Workshops). IEEE (2016)
24. Upadhyay, D., Sampalli, S.: SCADA (Supervisory Control and Data Acquisition) systems: Vulnerability assessment and security recommendations. Computers & Security 89, 101666 (2020)
25. Vielberth, M., Menges, F., Pernul, G.: Human-as-a-security-sensor for harvesting threat intelligence. Cybersecurity 2(23) (2019)
26. Yampolskiy, M., Horváth, P., Koutsoukos, X.D., Xue, Y., Sztipanovits, J.: A language for describing attacks on cyber-physical systems. International Journal of Critical Infrastructure Protection 8, 40–52 (2015)