PeerJ Computer Science

HVA_CPS proposal: a process for hazardous vulnerability analysis in distributed cyber-physical systems

Alan Jamieson[1], Chris Few[2], Kenny Awuson-David[1] and Tawfik Al-Hadhrami[3]

[1] The Office of Gas and Electricity Markets (Ofgem), London, United Kingdom
[2] National Grid, London, United Kingdom
[3] School of Science and Technology, Nottingham Trent University, Nottingham, United Kingdom

ABSTRACT

Society is increasingly dependent upon the use of distributed cyber-physical systems (CPSs), such as energy networks, chemical processing plants and transport systems. Such CPSs typically have multiple layers of protection to prevent harm to people or the CPS. However, if both the control and protection systems are vulnerable to cyber-attacks, an attack may cause CPS damage or breaches of safety. Such weaknesses in the combined control and protection system are described here as hazardous vulnerabilities (HVs). Providing assurance that a complex CPS has no HVs requires a rigorous process that first identifies potential hazard scenarios and then searches for possible ways that a cyber-attacker could cause them. This article identifies the attributes that a rigorous hazardous vulnerability analysis (HVA) process would require and compares them against related works. None fully meet the requirements for rigour. A solution is proposed, HVA_CPS, which does have the required attributes. HVA_CPS applies a novel combination of two existing analysis techniques: control signal analysis and attack path analysis. The former identifies control actions that lead to hazards, known as hazardous control actions (HCAs); the latter models the system and searches the model for sequences of attack steps that can cause the HCAs. Both analysis techniques have previously been applied alone on different CPSs. The two techniques are integrated by extending the formalism for attack path analysis to capture HCAs. This converts the automated search for attack paths to a selected asset into an exhaustive search for HVs. The integration of the two techniques has been applied using HCAs from an actual CPS. To preserve confidentiality, the application of HVA_CPS is described on a notional electricity generator and its connection to the grid. The value of HVA_CPS is that it delivers rigorous analysis of HVs at system design stage, enabling assurance of their absence throughout the remaining system lifecycle.

Subjects Algorithms and Analysis of Algorithms, Autonomous Systems, Computer Networks and Communications, Embedded Computing, Security and Privacy
Keywords Attack graph, Control actions, Cyber-security, HVA_CPS, Hazard, Vulnerability, Operational technology

Submitted 17 November 2022
Accepted 19 January 2023
Published 21 April 2023

Corresponding author
Kenny Awuson-David,
kenny.awuson-david@ofgem.gov.uk

Academic editor
Muhammad Aleem

Additional Information and Declarations can be found on page 25

DOI 10.7717/peerj-cs.1249

How to cite this article Jamieson A, Few C, Awuson-David K, Al-Hadhrami T. 2023. HVA_CPS proposal: a process for hazardous vulnerability analysis in distributed cyber-physical systems. *PeerJ Comput. Sci.* **9**:e1249 http://doi.org/10.7717/peerj-cs.1249

Table 1 **Definition of terms.**

Term	Definition
Loss	Events which the CPS protection system is intended to prevent: *e.g.*, damage or destruction of the physical system; injury or loss of life; release of toxic chemicals into the environment.
Hazard	A system state that can lead to a loss in the worst-case environmental conditions, such as a cyber-attack.
Component vulnerability	A weakness in a system component that a cyber-attacker could exploit to gain additional system privileges.
Hazardous vulnerability (HV)	A type of hazard in which a combination of component vulnerabilities enables a cyber-attacker to manipulate both the control and protection systems such that a demand on the protection system is created but not met and causes a loss.
Hazardous vulnerability analysis (HVA)	Analysis of whether a CPS has hazardous vulnerabilities.

INTRODUCTION

CPSs are integrations of computation and physical processes (*Lee, 2008*). Their uses include autonomous vehicles, chemical processing plants, power generators and manufacturing plants. In each case, the physical process is controlled by the computational process. The computational or cyber process may be vulnerable to cyber-attacks which can disrupt, damage or destroy the physical process. Examples of such attacks are the Stuxnet attack on the Iranian uranium enrichment process, (*Awuson-David, 2022*; *Langner, 2011*), the Shamoon attack on Saudi Aramco (*Bronk & Tikk-Ringas, 2013*) and the Triton attack on a Saudi Arabian oil refinery (*DiPinto, Dragoni & Caracano, 2018*). The Stuxnet and Triton attacks both exploited what this article defines as 'hazardous vulnerabilities': *i.e.,* a combination of component vulnerabilities within a system, which, if exploited, can cause a loss. In this context, losses are events which cause harm to people, equipment or the environment. Hazards are system states which, if exploited by a cyber-attacker or other environmental conditions, will lead to losses. Hazardous vulnerabilities typically occur in a CPS when an attacker can manipulate both the control and protection or safety systems from a single point of access. The impact from exploitation of hazardous vulnerabilities in critical national infrastructure (CNI) can be considerable and it is therefore highly desirable to minimise them during the system design phase.

The gap that this article addresses is that between the high level of assurance needed that CNI does not have hazardous vulnerabilities, and the lower level of assurance provided by existing methods. A key challenge is analysing CPSs as both holistic systems of systems with emergent security properties, and as a large collection of components each with their own security properties. The need to consider both holistic and reductionist viewpoints makes hazardous vulnerability analysis of CPS highly complex. This introduction describes some background concepts and practices which are relevant to the challenge and to the related works reviewed in the next section. Key concepts are threat modelling and attack path analysis.

For a safety critical CPS, process hazard analysis (PHA) is commonly undertaken during the system design phase to identify potentially hazardous states (*Baybutt, 2015*; *Lyu, Ding & Yang, 2019*). It is with this information, at this point in the system lifecycle, that security can be implemented most cost effectively, and the benefits can be realised throughout the system lifetime. However, in order to decide what security measures are sufficient, risk managers need to consider what threats need to be countered. In general, simply following generic security standards or good design practices is not sufficient to determine what threats the system will be secure from, because these measures do not analyse system security from the attackers' perspective. As recognised by the UK National Cyber Security Centre, the concept of attacker sophistication levels can help risk managers specify the threats the system is to be resilient to *M. P (2018)*. Defining attacker sophistication and techniques is often referred to as 'threat modelling'. System security can then be tracked in terms of the minimum attacker sophistication level needed to cause a given level of impact. In principle, the more precisely the threat models are defined, the more precisely system security can be defined. The evolution of threat models to current capabilities is outlined below.

A simple early threat model was defined in 1983 for attacks on public key protocols (*Dolev & Yao, 1983*). More recently, a set of seven attacker sophistication levels has been defined by the Oasis open standards body in its standard for Structured Threat Information eXpression (STIX) (*STIX, 2021*). Further material to help develop attacker sophistication levels is provided in a study of 'Attacker Models and Profiles for Cyber-Physical Systems' (*Rocchetto & Tippenhauer, 2016*). This study defines six common attacker profiles in terms of 29 separate attributes, such as knowledge, aims and resources. Threat models can be further refined to reflect weaknesses of a specific system by drawing from a knowledge base of attacker techniques applied to similar systems; *e.g* (*Ahmed et al., 2022*). A still further level of granularity in defining attacker profiles is provided by the meta attack language (MAL) (*Johnson, Lagerström & Ekstedt, 2018*). This defines a machine-readable syntax for describing attack steps in terms of the system privileges accessible to an attacker at the start and end of a step within an attack path, and the likely time for an attacker to complete them. There is a growing ecosystem of MAL programmes which already defines attack steps for IT systems (coreLang), industrial control systems (icsLang), vehicles (vehicleLang) and power systems (powerLang) (*Katsikeas et al., 2020*; *Katsikeas et al., 2019*; *Andrew, Katsikeas & Hacks, 2022*; *Hacks et al., 2020*; *Hacks & Katsikeas, 2021*). Collectively, these approaches provide a rich repository of material for defining the threat scenarios that a CPS is to be defended against (*Awuson-David et al., 2021*).

Assessing which types of attack a CPS will be resilient to requires iterative analysis of whether an attacker with access to one component in the CPS has the capabilities to overcome its defences and access other components to progress along the cyber kill chain (*Yadav & Mallari, 2015*). This is known as attack path analysis. Providing security assurance of this type at system design stage requires detailed analysis before the system can be physically tested. Attack graphs provide one means of capturing the information needed to support this analysis (*Shandilya, Simmons & Shiva, 2014*).

As the capabilities of attackers grow in sophistication and CPSs become ever more complex, analysing system security at the design stage becomes ever more difficult. In *Rocchetto, Ferrari & Senni (2019)*, the authors claim that "the manual extraction of threat scenarios is extremely complex and highly error prone when considering a CPS, and may be unfeasible for large-scale CPS, where there can be thousands of complex attacks." In these circumstances, some automation of the vulnerability analysis is highly desirable.

This article proposes a novel process for determining the existence of hazardous vulnerabilities in distributed cyber-physical systems (CPS). The contribution of this article is the HVA_CPS process, a novel refinement and combination of existing processes for analysing the security of CPS. In the HVA_CPS process, much of the vulnerability analysis is automated. The process provides detailed traceability from threat scenarios to potential exploitation of hazardous vulnerabilities. Traceability is beneficial because it helps risk managers to make targeted and evidenced business cases for security improvements (*Awuson-David et al., 2021*).

The article is structured as follows: Section 2 reviews related works; Section 3 illustrates a hazardous vulnerability in a CPS; Section 4 derives attributes needed for a rigorous HVA process; Section 5 presents the new HVA_CPS process, Section 6 evaluates HVA_CPS against the attributes needed for rigour; finally, Section 7 draws conclusions on the value of HVA_CPS to risk managers and Section 8 outlines some limitations creating opportunities for further work.

RELATED WORKS

Many studies have contributed to the development of processes and tools for analysing the cyber security of information and control systems. Surveys of related methods are presented in *Cherdantseva et al. (2016)*; *Nguyen et al. (2015)*; *Kriaa et al. (2015)*; *Geismann & Bodden (2020)*; *Mohamed, Kardas & Challenger (2021)* and *Mohamed, Challenger & Kardasa (2020)*. This section reviews selected methods in terms of their suitability for hazardous vulnerability analysis (HVA).

An early stage in a rigorous HVA process is to identify possible hazards. There are long-standing safety processes for hazard analysis, *e.g.*, (*Baybutt, 2015*; *Dunjó et al., 2010*). In the context of cyber-security and cyber-physical systems (CPSs), the hazards of most interest are those which a cyber-attacker can exploit by manipulating information or control signals. A methodology which outputs unsafe control actions and scenarios which can cause them is systems theoretic process analysis (STPA) (*Young & Leveson, 2014*; *Ishimatsu et al., 2010*), as illustrated in Fig. 1. The methodology has since been extended and generalised to include any form of hazardous control actions (HCAs). This methodology, known as Security Enhanced-STPA (SE-STPA), is also combined with formal methods to prove that the control model does not permit hazardous states to arise (*Butler et al., 2019*). However, on its own, SE-STPA does not provide the means to analyse the security of a specific implementation of the control model against detailed threat models.

Another approach which uses a simple threat model and formal methods to prove security properties of complex systems is described in *Li (2018)*. Its inputs are diagrams

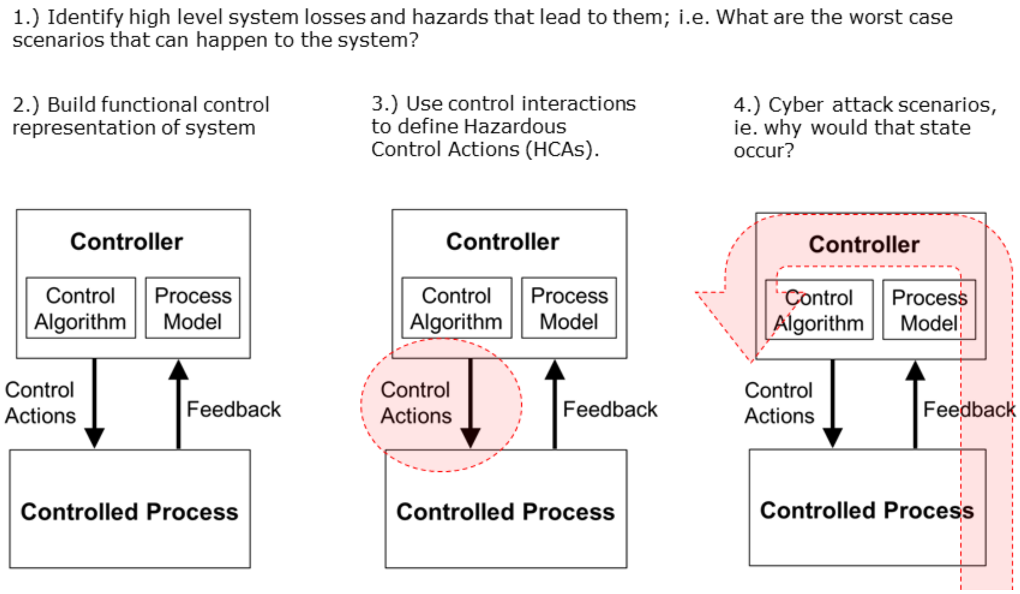

1.) Identify high level system losses and hazards that lead to them; i.e. What are the worst case scenarios that can happen to the system?
2.) Build functional control representation of system
3.) Use control interactions to define Hazardous Control Actions (HCAs).
4.) Cyber attack scenarios, ie. why would that state occur?
Controller
Control Algorithm
Process Model
Control Actions
Feedback
Controlled Process
Controller
Control Algorithm
Process Model
Control Actions
Feedback
Controlled Process
Controller
Control Algorithm
Process Model
Control Actions
Feedback
Controlled Process

Figure 1 System theoretic process analysis (STPA) steps.
Full-size DOI: 10.7717/peerjcs.1249/fig-1

using Systems Modelling Language—Security (SysML-Sec). However, the threat model is the early model developed for public key protocols (*Dolev & Yao, 1983*). This limits the threat scenarios to a small subset of those applicable to CPSs.

A SysML-based security analysis process with enhanced threat modelling is described in *Bakirtzis et al. (2020)*. SysML internal block diagrams are manually enriched with descriptive keywords and are then, through an automated process, converted to graphs in GraphML format. The nodes in the graph represent system components; the links between them represent communication paths. Another automated process searches public vulnerability databases for vulnerabilities associated with the system components. The nodes are then associated with relevant vulnerabilities. The output is a set of 'exploit chains' for which there is an associated vulnerability or weakness for each link in the chain. Whilst this information is useful to designers at the design stage, it requires substantial manual analysis to derive from this the minimum attacker sophistication needed to cause harm. An analyst would need to consider the defences in place for each step and whether there were attack vectors, such as social engineering of privileged users, which avoided the need to find an exploitable vulnerability in each link of the identified exploit chains. The method does not specifically look for hazardous vulnerabilities nor does it explicitly use a threat model.

An approach which, in principle, reduces the burden upon the human analyst is described in *Deloglos, Elks & Tantawy (2020)*. It models the likely sequence of attack steps as an attacker enters and moves through a CPS. Attackers can be defined through an extensible set of attributes and, at each stage of the attack path, the most likely next step is calculated from the attacker's attributes, the assets accessible to them and their vulnerabilities. The method includes a threat model, a knowledge base and a repeatable process but the authors do not claim that it has been applied on a complex CPS and it does not appear to be supported by tools capable of making it cost effective at scale.

A model-based approach which is supported by a mature tool is 'An Actor-Based Approach for Security Analysis of Cyber-Physical Systems' (*Moradi et al., 2020*). It uses an actor-based modelling language, Timed Rebeca (*Khamespanah et al., 2015*), to model the state of a simple CPS and a model checker to search for breaches of security requirements. The method identifies combinations of simple attacks which breach security. However, the method is only applied to a high-level logical model, so would be unable to predict the level of attacker sophistication needed to compromise a specific implementation of the system design.

Another tool-supported approach which does have some capability to model the implementation of a control system is described in *Kriaa (2016)*. 'Joint safety and security modeling for risk assessment in cyber physical systems' uses the Figaro object-oriented probabilistic programming language (*Pfeffer, 2009*) as the basis for building models of CPSs. It builds upon the use of Boolean driven Markov processes (BDMP) for analysing attack paths, as described in *Kriaa, Bouissou & Laarouchi (2015)* and *Pietre-Cambacedes, Deflesselle & Bouissou (2011)*. Objects are defined to represent generic components of CPSs, including their potential failure modes and attack steps. These object types can be instantiated and connected to represent the CPS of interest. Figaro converts this diagrammatic representation of the CPS into a text-based representation enabling automated processing. Using BDMP or Monte Carlo simulations, associated tools can predict the most likely sequences of events to cause breaches of safety or security. A limitation of the method is that it does not appear to include defensive measures at the object level. Consequently, as described, the method cannot predict the attacker sophistication needed to overcome them. The article does not claim that the method has been demonstrated on a complex CPS.

The problem of modelling defensive measures is addressed in a modelling tool, cyber security modelling language (CySeMoL) described in *Holm et al. (2013)*. CySeMoL defines a meta-model through which models of computer systems can be built. The meta-model specifies 22 different types of objects which include operating system, software product, firewall and dataflow. Each object type has associated attributes of attack steps and defences. The meta-model also defines associations between object types; *e.g.*, a dataflow may be permitted to pass through a firewall. Models built using CySeMoL can be converted into attack graphs through an automated process. An attack graph is a form of Bayesian network in which the network nodes represent levels of privilege that an attacker has gained in the system of interest (*Ou, Boyer & McQueen, 2006*). The links between the nodes represent attack steps to increase attacker privileges. Each attack step is assigned a level of difficulty according to the value of the defensive attributes that must be overcome. Automated analysis of the attack graph identifies the most exploitable attack paths through Dijkstra's shortest path search algorithm (*Dijkstra, 1959*).

In 2014, CySeMoL was refined into the Predictive Probabilistic Cyber Security Modelling Language (P2CySeMoL) with a more detailed meta-model. P2CySeMoL was used to model six systems used in a cyber defence exercise and calculate their expected times to compromise. These times had a statistically significant correlation with the actual times to compromise taken by the exercise participants (*Holm et al., 2014*). P2CySeMoL was further developed in a commercial product described in *Ekstedt et al. (2015)* which was trialled on an industrial control system in *Few et al. (2021)*. The predicted shortest attack paths provided useful insights to the system control engineer and cyber specialist. The same method and tool have also been used in the analysis of load balancing in a renewable energy grid (*Vernotte et al., 2018*). A limitation of the approach is that the modelling tool was not customised to modelling control systems as distinct from IT systems. In consequence, it did not model combinations of control actions which would damage the generator

A process which does combine the benefits of systems theoretic process analysis (STPA) with those of attack graphs is presented in *Castiglione & Lupu (2020)*. It uses STPA to derive hazardous control actions from behavioural and functional system models. The system is then modelled using architecture analysis design language, from which an attack graph can be generated and analysed to determine whether an attacker can cause hazards (*Carnegie-Mellon University & Software Engineering Institute, 2006*). Finally, the physical system is modelled using differential equations to show the impact of the hazard. The method is applied to a simple communication-based train control system. The most significant limitations are the fidelity of models that can readily be built, the maturity of the supporting tool chain and its limited application to real systems.

Each of the processes described above has its merits but also at least one limitation that limits the level of rigour that it can deliver. Thus, there is a need for the derivation of the set of attributes needed for a rigorous hazardous vulnerability analysis (HVA) process and development of a compliant solution.

The following section presents an example of a hazardous vulnerability which sets the scene for deriving the attributes needed for a rigorous process to find such vulnerabilities or evidence their absence.

AN ILLUSTRATIVE HAZARDOUS VULNERABILITY

A hazardous vulnerability (HV) is illustrated in Fig. 2, the HV diagram. It shows a generator connected to the electricity grid and the cyber systems which control it. The control system is aligned with the Purdue reference architecture comprising levels 1 to 4 (*Williams, 1994*). This layered architecture aims to create multiple barriers to an internet-based attacker, whilst enabling ultimate control of the generator to be retained by business managers *via* workstations in level 4, the enterprise zone. Level 3, the operations management zone, supports decision making on efficient operation of the generator and its associated assets. Level 2, the supervisory zone, provides functionality to configure settings in control and protection functions such as the relays which connect and disconnect the generator from the grid. Level 1 includes the devices which send commands to the physical system and receive data from its sensors.

The generator control system is designed to synchronise the generator's voltage frequency and phase with that of the distribution system before the two are connected. If the generator is connected to the grid when not synchronised (malsynchronisation), the differences in voltage between the grid and generator cause large currents to flow in the

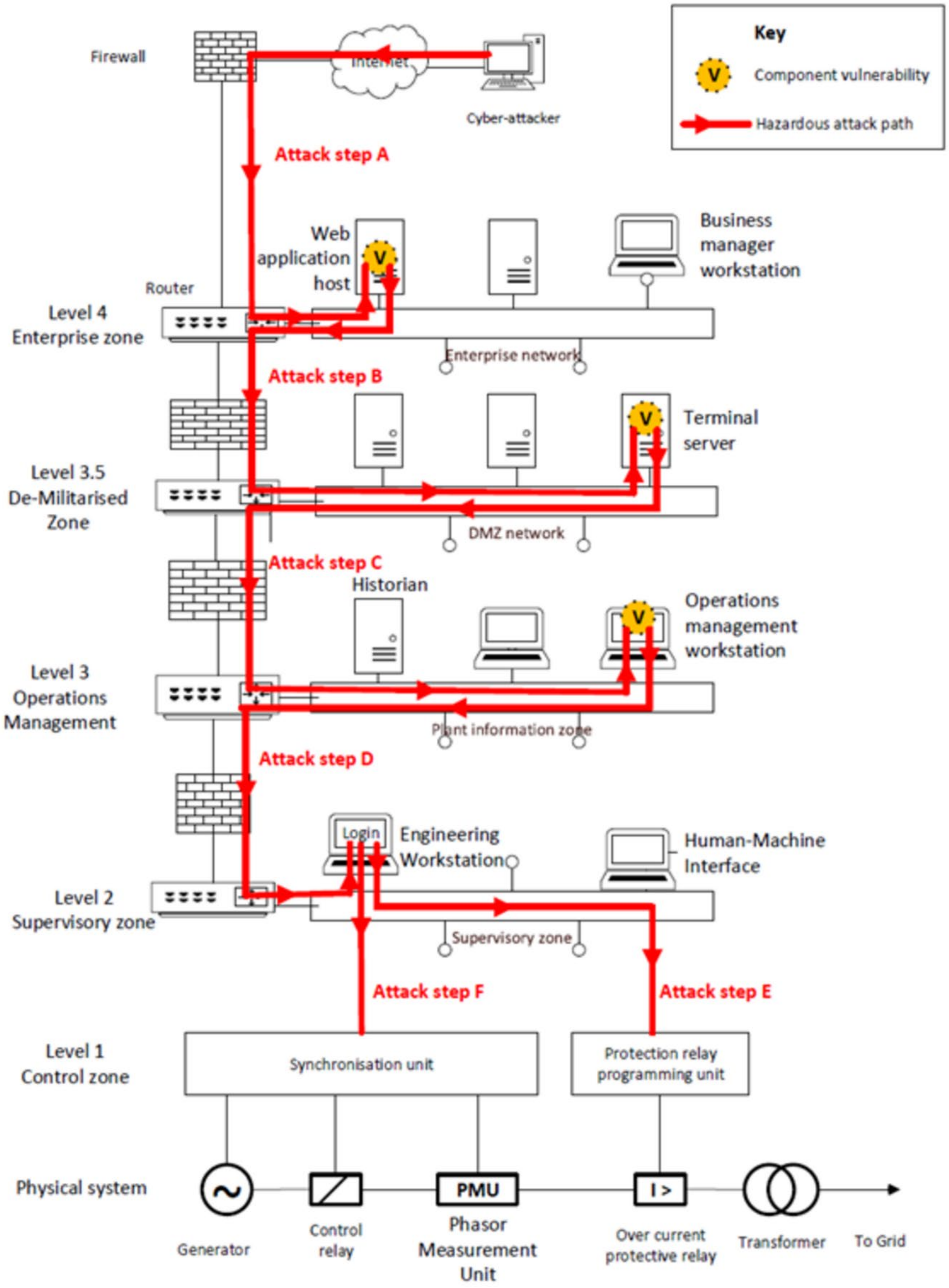


Figure 2 Hazardous vulnerability (HV) diagram.

Full-size DOI: 10.7717/peerjcs.1249/fig-2 generator windings, which, in turn, generate large torques in the generator shaft. Both can damage the generator. The destructive consequences of such a failure in both the control and protection systems was shown in the Aurora generator test of 2007 (*Potvin, 2019*; *Greenberg, 2020*; *Greenberg, 2019*). In this test, a 28-tonne generator was repeatedly connected to the grid when unsynchronised, as shown in the video at *CNN (2007)*. The generator was destroyed within a minute.

This scenario can be readily prevented by the use of a protection relay which monitors current flow. If a pre-set current threshold is exceeded, the relay will disconnect the generator from the grid within the safety response time (SRT), preventing any damage. However, if an attacker can manipulate both the control and protection systems, the generator can still be destroyed.

The red arrows in the HV diagram show a sequence of possible attack steps which combine to form an attack path from the internet to both control and protection relays. In attack step A, the attacker makes a connection to an internet-facing web application and exploits a vulnerability to gain user privileges on the hosting operating system. In step B, the attacker uses this privilege to connect to a terminal server in the De-Militarised Zone (DMZ). The purpose of the terminal server is to give the business manager in the enterprise zone controlled access to operational data collected in the historian. In our example, this is poorly configured, and in attack step C, the attacker is able to reach the Operations Management workstation. From here the attacker is assumed to be able to obtain credentials that enable access to the engineering workstation in the supervisory zone. Attack step D uses these credentials to log in to the engineering workstation and gain access to the applications used to configure the synchroniser and protection relay. In attack step E, the threshold current at which the protection relay will disconnect the generator from the grid is reset to a level in excess of that caused by a synchronisation failure. In attack step F, the synchroniser is reprogrammed such that the generator connects to the grid when not synchronised. This creates a demand on the protection relay, which fails to respond within the SRT, causing generator damage or destruction.

A rigorous HVA process would reliably show whether this or other hazardous vulnerabilities are exploitable for a given threat scenario.

ATTRIBUTES OF A RIGOROUS HVA PROCESS FOR DISTRIBUTED CPSS

There are many frameworks for assessing system security, but to the authors' knowledge, none that are specifically designed to find hazardous vulnerabilities (HVs) in distributed cyber-physical systems (CPSs). This section derives from first principles the attributes needed for a process that reliably finds HVs in distributed CPSs. The purpose of developing these attributes is to form the basis for comparing new and existing HVA processes.

To meet the needs of a risk manager, the rigorous process would predict whether the CPS is resilient to specified threat scenarios, including those exploiting hazardous vulnerabilities. This necessitates a detailed threat model, and an understanding of the system level hazards that are possible for a cyber-attacker to cause.

Table 2 **Attributes of a rigorous HVA process for distributed CPS.**

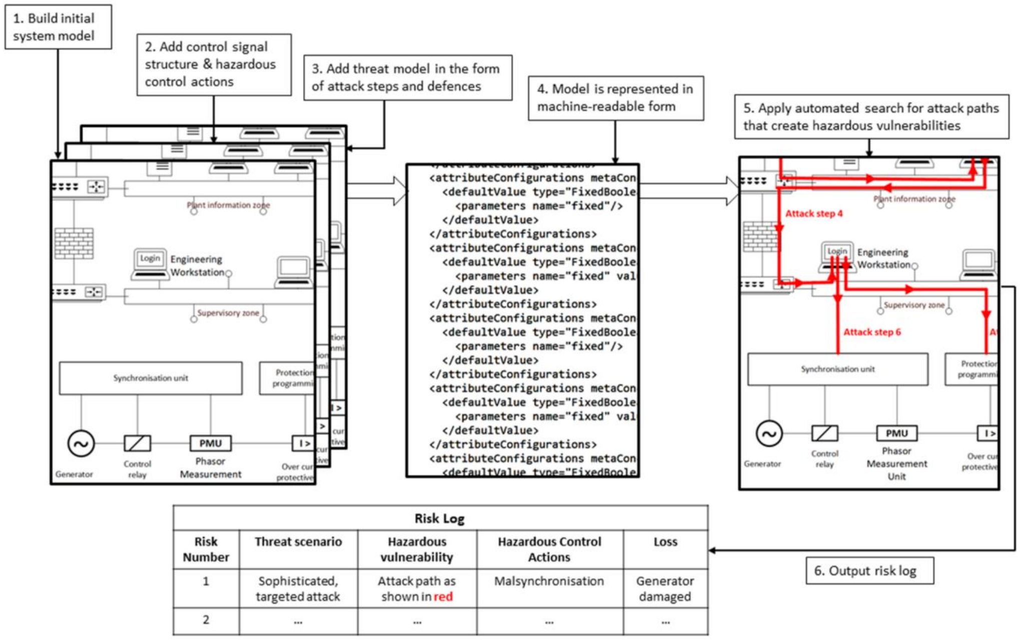
Attribute category	Attribute name	Attribute	Justification
Process inputs	Threat model	A threat model for defining threat scenarios in terms of attacker capabilities and system access.	Enables assessment of system security in terms of the threat scenarios the system is resilient to.
	Knowledge base	Includes necessary information of: Common attacker tactics and techniques Common vulnerabilities, exposures and exploits Attacks and vulnerabilities that are specific to CPSs Hazardous control actions specific to the system of interest.	Necessary for accurate assessment of system security.
Process	Full lifecycle support	Can be applied throughout the system lifecycle.	Enables security analysis at design stage onwards.
	Reproducible	Is recorded in sufficient detail to be repeatable & reproducible.	To give risk owners confidence in the security assessment.
	Tool support	Tool(s) have been developed to minimise the human effort and expertise needed to apply the methodology.	Increases analyst productivity.
	Maturity	Sufficiently mature to have been demonstrated on a complex distributed CPS.	To show that the methodology is viable and scalable.
	High fidelity model	A high-fidelity system model capable of capturing component vulnerabilities.	To enable accurate system security analysis.
Process outputs	Traceability of threat to loss	A clear chain of causality from threat scenarios to possible attack steps and attack paths to exploitation of HVs that cause a loss.	Provides assurance that the analysis is correct.
	Predictive power	A prediction of whether the CPS is resilient to specified threat scenarios including those targeting hazardous vulnerabilities.	Helps risk managers judge whether security is adequate for their needs.

The analysis should be presented in sufficient detail that risk managers would be able to identify the security improvements which would increase the capabilities required for an attacker to cause a given level of impact. It would identify the points in the system which the most damaging attack paths pass through. These are known as choke points, and the best options for improving system security often include securing them because they can block the most attack paths. An example in Fig. 2 of a choke point is the Level 2 network switch as the attack paths to both the control and protection systems pass through it.

Achieving this level of analysis requires an understanding of each device and software application with details of their defences and connections. It also needs to capture threat scenarios, attack vectors and vulnerabilities applicable to the system of interest. The HVA process therefore needs to be able to draw from knowledge bases containing such information.

A rigorous HVA process would reliably show whether this, or other hazardous vulnerabilities, are exploitable for a given threat scenario. This enables security to be designed into the system and maintained as the system and its environment evolve.

For risk managers to have confidence in the analysis, it needs to be repeatable and reproducible so that different assessors would reach the same conclusions. It should also be supported by tools to minimise the necessary human effort and expertise. In combination,


1. Build initial system model
2. Add control signal structure & hazardous control actions
3. Add threat model in the form of attack steps and defences
4. Model is represented in machine-readable form
5. Apply automated search for attack paths that create hazardous vulnerabilities
6. Output risk log
Plant information zone
Login
Engineering Workstation
Supervisory zone
Synchronisation unit
Protection programming
Generator
Control relay
PMU
Phasor Measurement
Phasor Measurement Unit
Over current protective
Attack step 4
Attack step 6

```
<attributeConfigurations metaCon
  <defaultValue type="FixedBoole
    <parameters name="fixed"/>
  </defaultValue>
</attributeConfigurations>
<attributeConfigurations metaCon
  <defaultValue type="FixedBoole
    <parameters name="fixed" val
  </defaultValue>
</attributeConfigurations>
<attributeConfigurations metaCon
  <defaultValue type="FixedBoole
    <parameters name="fixed"/>
  </defaultValue>
</attributeConfigurations>
<attributeConfigurations metaCon
  <defaultValue type="FixedBoole
    <parameters name="fixed" val
  </defaultValue>
</attributeConfigurations>
<attributeConfigurations metaCon
  <defaultValue type="FixedBoole
```

Risk Log

Risk Number	Threat scenario	Hazardous vulnerability	Hazardous Control Actions	Loss
1	Sophisticated, targeted attack	Attack path as shown in red	Malsynchronisation	Generator damaged
2	...	...	...	...

Figure 3 **HVA_CPS process—conceptual diagram.**

Full-size DOI: 10.7717/peerjcs.1249/fig-3

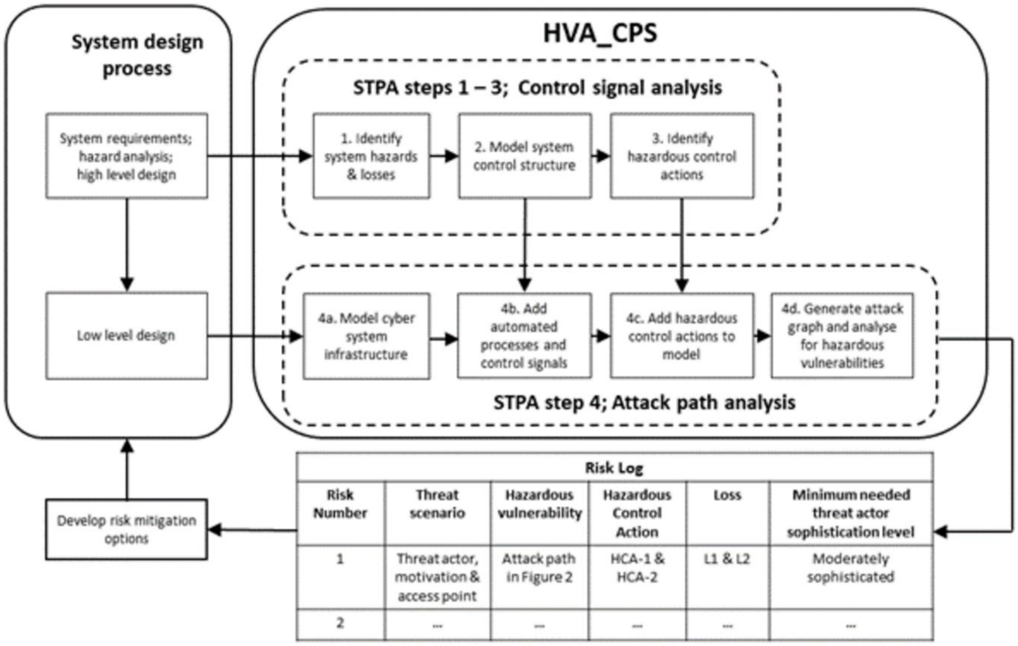
these attributes should provide a high level of traceability from threat scenarios to hazards: *i.e.,* there is a clear chain of attacker actions leading from the threat scenario to the hazard. These attributes are summarised in Table 2.

There is a need for an HVA process with the attributes discussed in this section. In the following section a new HVA process for CPSs, defined as HVA_CPS, will be introduced.

THE PROPOSED PROCESS: HVA_CPS

This section describes the HVA_CPS process for distributed CPSs. It draws from methods described in the related works and aims to meet all the attributes for a rigorous process. The process is illustrated conceptually in Fig. 3. Using a tool designed for the purpose, a model is built of the CPS. The model has the structure and information to largely automate HVA. The novelty of HVA_CPS is the combination of techniques and tools to give detailed traceability from threat scenarios to possible attacker actions to system losses. If the CPS has no hazardous vulnerabilities (HVs), evidence of their absence is provided through automated, exhaustive searches of the model for them.

The HVA_CPS process is shown in more detail in Fig. 4. It includes two previously developed sub-processes which can largely be performed concurrently. The contribution of this article is the integration of the sub-processes, which enables the detailed traceability of threats to hazards. The first sub-process, control signal analysis, is used to identify hazardous control actions (HCAs); *i.e.,* combinations of control actions which cause hazards (*Young & Leveson, 2014*). Hazards are system states which can lead to losses in the worst-case environmental conditions, such as cyber-attacks. In this context, control


System design process
System requirements; hazard analysis; high level design
Low level design
Develop risk mitigation options
HVA_CPS
STPA steps 1 – 3; Control signal analysis
1. Identify system hazards & losses
2. Model system control structure
3. Identify hazardous control actions
4a. Model cyber system infrastructure
4b. Add automated processes and control signals
4c. Add hazardous control actions to model
4d. Generate attack graph and analyse for hazardous vulnerabilities
STPA step 4; Attack path analysis
Risk Log

Risk Number	Threat scenario	Hazardous vulnerability	Hazardous Control Action	Loss	Minimum needed threat actor sophistication level
1	Threat actor, motivation & access point	Attack path in Figure 2	HCA-1 & HCA-2	L1 & L2	Moderately sophisticated
2	...	...	...	...	...

Figure 4 The HVA_CPS proposed process.

Full-size DOI: 10.7717/peerjcs.1249/fig-4

signals include commands from a controlling process to elements that drive the physical system, and feedback from sensors on the physical system to the controlling process. The second sub-process, attack path analysis, identifies sequences of attacker actions which can cause HCAs (*Johnson, Lagerström & Ekstedt, 2018*; *Ekstedt et al., 2015*). The HVA_CPS process steps are described below. Steps 1–3 follow those in STPA, as shown in Fig. 1. These steps output the control system structure and the HCAs. Step 4 in STPA is represented by steps 4a –4d in HVA_CPS. These steps identify cyber-attack scenarios which can cause the HCAs.

An output of HVA_CPS is a risk log, for which each entry details the threat scenario in terms of the threat actor, their motivation and access point; the exploited HV; the HCAs it initiates; the loss that is caused, and the minimum threat actor sophistication level needed to carry out the attack.

The HVA_CPS process is now explained in detail through the use case of an electricity generator connecting to a grid, as described in Section 3.

Control signal analysis

System hazards and losses are identified in Step 1 of HVA_CPS. They are derived from high-level system requirements and the associated system design. System losses can be in many forms, including financial, safety and service delivery. In the context of the generator connected to an electricity grid described in Section 3, losses (L) include damage to the generator (labelled as L1) and disruption of power delivery to the grid (L2). Hazards (H)

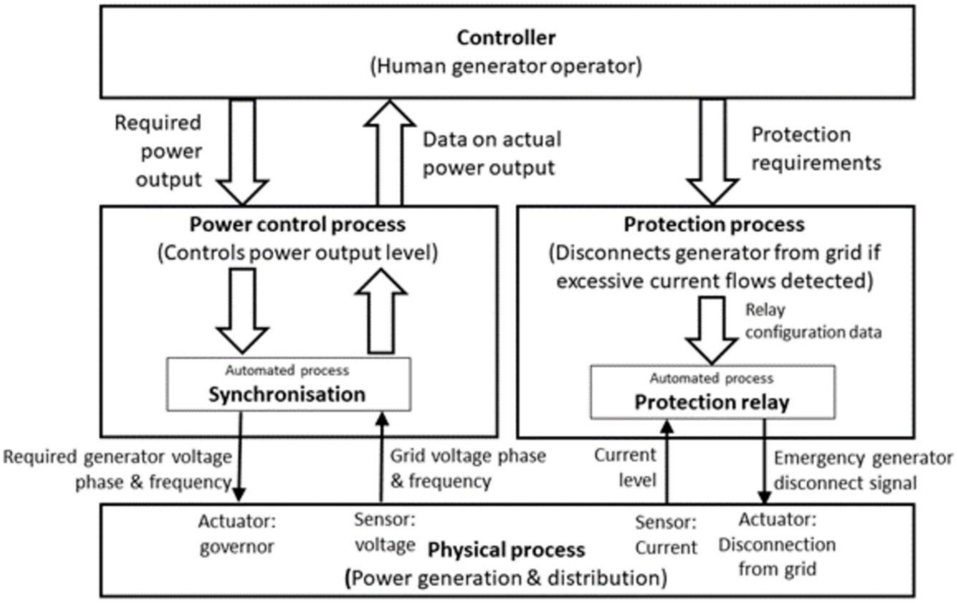

Controller
(Human generator operator)
Required power output
Data on actual power output
Protection requirements
Power control process
(Controls power output level)
Protection process
(Disconnects generator from grid if excessive current flows detected)
Relay configuration data
Automated process
Synchronisation
Automated process
Protection relay
Required generator voltage phase & frequency
Grid voltage phase & frequency
Current level
Emergency generator disconnect signal
Actuator: governor
Sensor: voltage
Physical process
(Power generation & distribution)
Sensor: Current
Actuator: Disconnection from grid

Figure 5 **Control signal diagram for generator synchronisation.**
Full-size DOI: 10.7717/peerjcs.1249/fig-5

are system states which can lead to losses in the worst-case environmental conditions, such as cyber-attacks.

In order to identify scenarios in which losses arise, the control structure is modelled in Step 2. The model shows the control signals that synchronise the generator to the grid and connect or disconnect them. The control signals can be modelled as passing between the system controller, controlling processes and the physical process of energy generation and transmission. A simplified version of the control signal diagram is shown in Fig. 5. In this model, the power control process is separated from the protection process. The conditions under which the protection relay should disconnect the generator from the grid are set by the operator and used to configure the relay. The automated processes and control signals captured in this step feed into the system model developed in step 4.

Possible HCAs are identified in step 3. For each possible control action, it is considered whether a hazardous state would arise in each of the following scenarios:

- Not providing the control action.
- Providing the control action.
- Providing a potentially safe control action but too early, too late, or in the wrong order.
- The control action lasts too long or is stopped too soon (for continuous control actions, not discrete ones).

HCAs are described in terms of the controller, control action and the conditions in which the control action, or absence of it, becomes hazardous. This formulation enables traceability from specific actions by a cyber-attacker to the HCA and to a system loss. Table 3 shows example HCAs for the synchronisation unit and protection relay, and their

Table 3 **Illustrative HCAs for the synchronisation unit and protection relay.**

Controller	Control action	Hazardous control action (HCA)	Losses that result from a combination of HCA-1 & HCA-2
Synchronisation unit	Required generator voltage phase and frequency values	HCA-1: The synchronisation unit does not set the required voltage phase and frequency before the generator is connected to the grid.	L1 & L2
Protection relay	Emergency generator disconnect signal	HCA-2: The protection relay fails to open the circuit breaker between the generator and grid within the safety response time after an excessive current flow is detected.	

connection to losses. If required, HCAs can be prioritised according to the size of loss that they can cause.

The HCAs provide input to Step 4 in the STPA process, which is to identify scenarios that could lead to the HCAs occurring. In the usual application of STPA, the scenarios would include human error, component failures and physical attacks. However, the HVA_CPS process is applied to identifying scenarios that can be caused by cyber-attackers. This is the purpose of attack path analysis, which is described in steps 4a –4d.

Attack path analysis

Whereas the control signal analysis considers which failures in control can lead to losses, the attack path analysis considers whether a cyber-attacker can cause those control failures to occur. In order to understand whether a cyber attacker can cause any of the HCAs identified above, the system needs to be modelled in detail from the attacker's perspective.

A novel aspect of HVA_CPS is the level of detail that can be captured in the modelling formalism in a machine-readable format such that automated algorithms can search for attack paths that cause HCAs. The cyber aspects of the low-level system design are captured in step 4a in a model with the machine-readable format. The model is built from a class library of asset types, including host devices, applications, networks, dataflows, identities, vulnerabilities and firewalls. A modelling tool is used to instantiate asset types to represent system components. Connections between assets are represented in the model by relationships between the instantiated versions. For example, an application is hosted on a processor and operating system; it makes or receives connections to other applications, and it can have vulnerabilities.

Security aspects of the cyber system are captured through attributes associated with each asset type. These attributes represent attack steps and/or defences. Attack steps are actions that attackers can take to increase their level of access and privilege in the CPS. Attack steps can be assigned a measure of difficulty based upon relevant vulnerabilities, exploits and defences. Attack steps have associations with parent and child attack steps; *i.e.,* those attack steps which enable it and those which it enables.

Table 4 **Structure of the meta-model.**

Element type	Example
Asset type	Host device, application, connection, ICS application, synchronisation unit
Defensive attribute	Authentication, encryption, patched, locked down
Attack step	Exploit vulnerability, intercept dataflow, guess password
Attack step difficulty	Expressed in terms of expected time or threat actor sophistication level required to complete it.
Relationships	Application is hosted by device; user has an identity
Impact	Expressed by user-defined scale

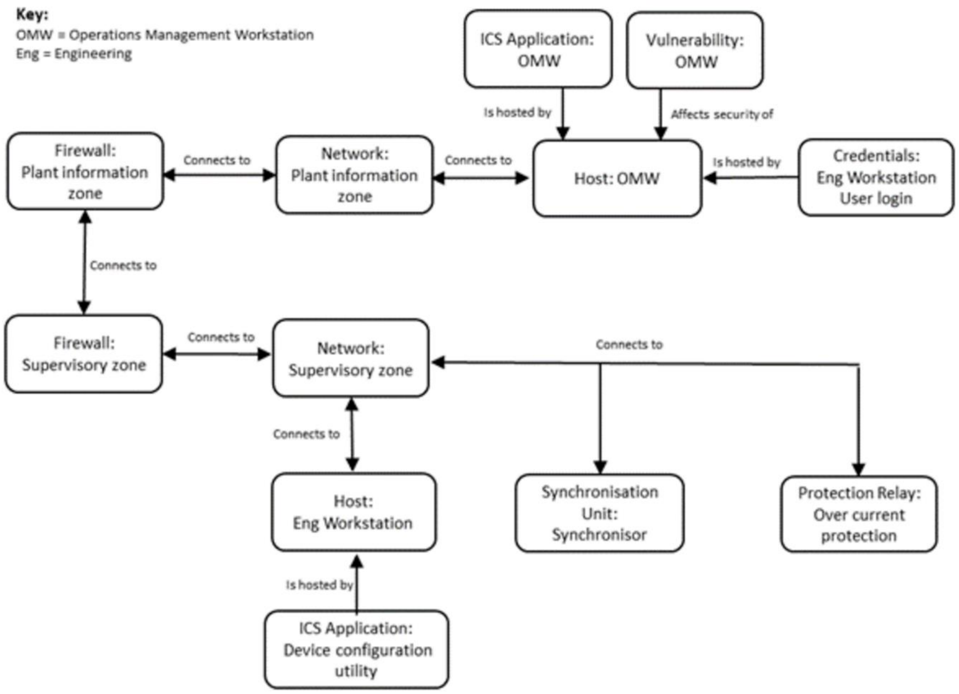

Key:
OMW = Operations Management Workstation
Eng = Engineering
ICS Application: OMW
Vulnerability: OMW
Is hosted by
Affects security of
Firewall: Plant information zone
Connects to
Network: Plant information zone
Connects to
Host: OMW
Is hosted by
Credentials: Eng Workstation User login
Connects to
Firewall: Supervisory zone
Connects to
Network: Supervisory zone
Connects to
Connects to
Host: Eng Workstation
Synchronisation Unit: Synchronisor
Protection Relay: Over current protection
Is hosted by
ICS Application: Device configuration utility

Figure 6 **Example asset connections.**

Full-size DOI: 10.7717/peerjcs.1249/fig-6

The definitions of asset types, attack steps, defences and relationships are known as the meta-model; *i.e.,* a model from which other models can be built. The structure of the meta-model as used in HVA_CPS is shown in Table 4.

The machine-readable version of the model is stored in an XML document with a schema aligned to its meta-model. Figure 6 illustrates how connections between selected assets can be represented in a view of the model generated by the modelling tool from its XML representation. This view includes the synchroniser and over-current protection

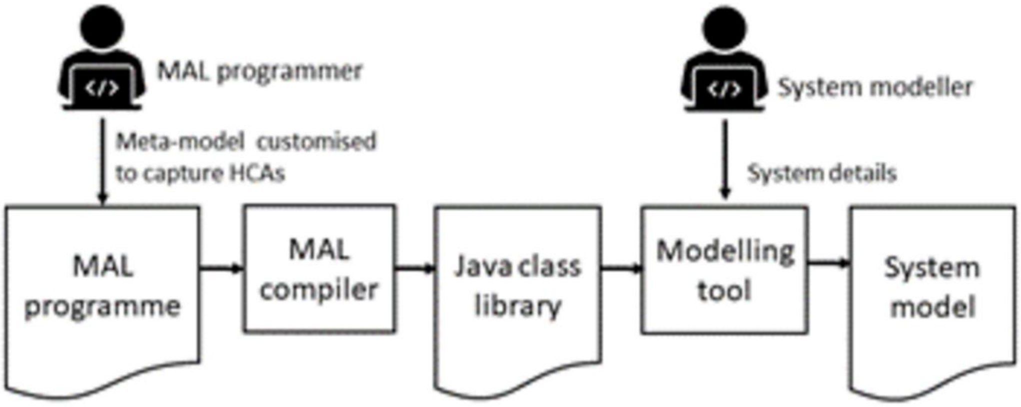


Figure 7 MAL programming process.

Full-size DOI: 10.7717/peerjcs.1249/fig-7

relay, which are manipulated by the cyber-attacker to cause HCA-1 and HCA-2, as shown in Table 3.

Figure 6 presents a view of asset connections in part of the system in Fig. 2, the HV diagram. In practice, to provide meaningful analysis of whether a system will be resilient to a particular type of cyber-attack, the system needs to be decomposed into much greater detail than the 12 assets shown in Fig. 6. For comparison, the model of a representative part of a generator control system built for the case study in *Few et al. (2021)* included 270 assets and ~500 relationships between them, which were specified in ~1,000 lines of XML.

A novel aspect of HVA_CPS is the ability to capture HCAs in a format consistent with the automated attack path analysis. This is achieved in step 4c by customising the meta-model so that the HCAs can be represented as a combination of attack steps against specific asset types. The benefit is that the automated search can be targeted to search for HCAs, and hence hazardous vulnerabilities.

In HVA_CPS, the meta-model is defined by a programme written in the Meta Attack Language (MAL). MAL programmes can be compiled into a Java class library and which is inputted into a modelling tool. Models can then be built using the object type, attack steps and defences defined in the MAL programme. One MAL programme can extend another, giving access to larger libraries of attack steps. The use of MAL is illustrated in Fig. 7.

The model developed to trial the modelling of HCAs used asset types and attack steps defined in two MAL programmes. One programme defines asset types which are common to most types of cyber systems and is called coreLang. The other defines asset types and attack steps specific to industrial control systems and is called icsLang. This includes a synchronisation unit, as shown in the HV diagram of Fig. 2. In combination, the two programmes define 34 asset types and over 100 attack steps. This permits modelling of the critical attack steps which enable an attacker to effect HCA-1 and HCA-2. Table 4 shows some example asset types and attack steps defined in icsLang and related to these control actions. If required, these could be customised further by more detailed MAL programming.

Table 5 Asset types and attack steps defined in Icslang.

Asset name	Asset description	Attack step name	Attack step description	HCA enabled
IcsSystem	Extends coreLang's System asset type with more attack vectors.	System Firmware	An adversary may exploit the firmware update feature on accessible devices to upload malicious or out-of-date firmware. Enables attack step E in HV diagram of Fig. 2.	HCA-1
IcsApplication	Extends coreLang's Application with Operational Technology attack vectors.	Manipulation of control	Enables attack step F in HV diagram of Fig. 2 by injecting a command to close the circuit breaker connecting the generator to the grid when they are not synchronised.	HCA-2

In the HVA_CPS process, once the initial model has been built, the control signal structure is added to the model, in step 4b. The automated processes are represented by ICS applications and the control signals are represented by control signal assets which are carried by network connections. In Fig. 5, the control signal diagram for generator synchronisation, the synchroniser and over current protection relay both implement automated processes. Using icsLang, the synchroniser is represented by an instance of the asset type 'IcsApplication', as shown in Table 5, and the control signal is represented by an instance of IcsControlData.

A novel aspect of HVA_CPS is the ability to capture HCAs in a format consistent with the automated attack path analysis. This is achieved in step 4c by assigning a user-defined impact level to attack steps which can trigger HCAs. For example, HCA-1 can be triggered by an attack step that updates the firmware in an IcsSystem representing the over current protection relay. HCA-2 can be triggered by an attack step to manipulate the time on the synchroniser. The benefit to HVA_CPS is that the automated search can be targeted to search for HCAs and, hence, hazardous vulnerabilities. As HCAs vary between CPSs, it may be necessary to customise the meta-model to accurately represent them. This can be done through the use of the Meta Attack Language (MAL).

Once the model has been created, step 4d of HVA_CPS uses an automated algorithm to generate an attack graph. The attack graph captures all possible attack steps defined within the model and reachable by the attacker from the point of entry. Generation of the attack graph is illustrated in Fig. 8 as follows:

- In step I, the attacker's entry point to the system is selected by the risk assessor and is represented by a specific asset within the model. In the example of the attack path in the HV diagram of Fig. 2, the entry point is a connection from the attacker *via* the internet to the firewall protecting Level 4, the enterprise zone. The firewall rule set will permit connections to further assets, possibly subject to authentication.
- In step II, each possible connection available to the attacker from the entry point is added in the form of a link.
- In step III, the asset at the end of each connection is added. This will typically be a port on a host device such as the web application host shown as the first attack step in Fig. 2.

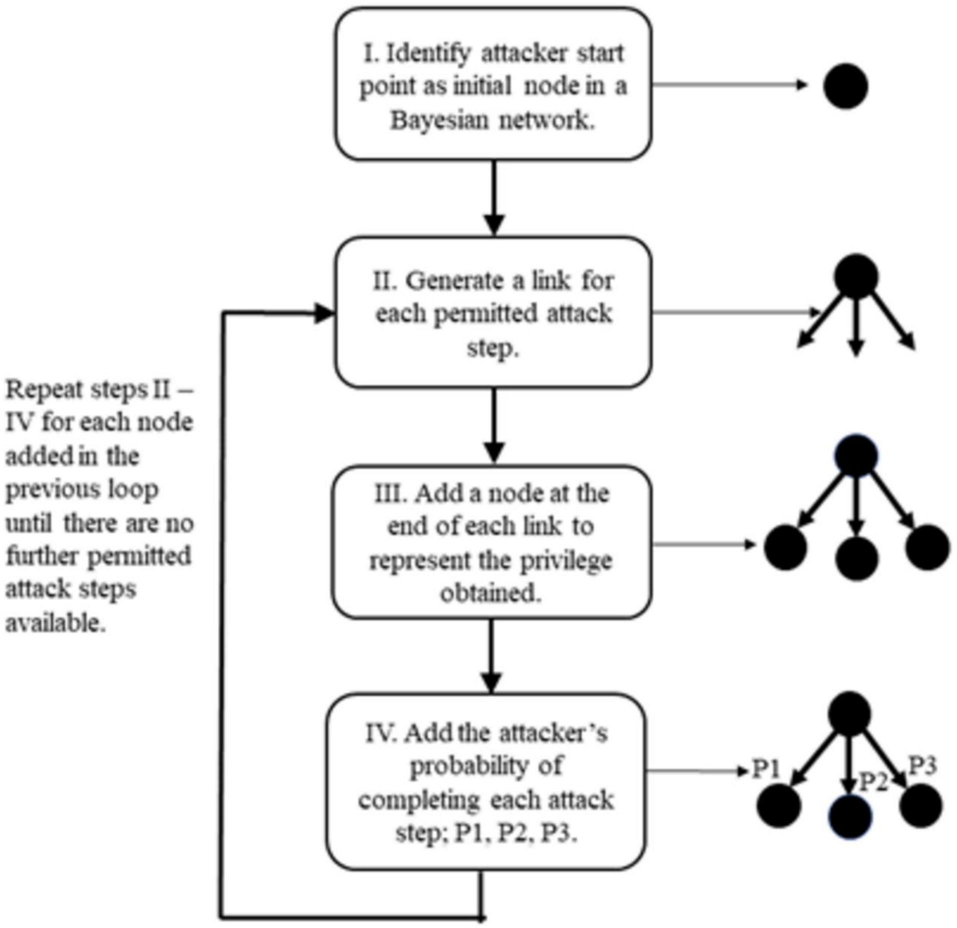

I. Identify attacker start point as initial node in a Bayesian network.
II. Generate a link for each permitted attack step.
III. Add a node at the end of each link to represent the privilege obtained.
IV. Add the attacker's probability of completing each attack step; P1, P2, P3.
Repeat steps II – IV for each node added in the previous loop until there are no further permitted attack steps available.
P1
P2
P3

Figure 8 **Attack graph generation.**

Full-size DOI: 10.7717/peerjcs.1249/fig-8

- The difficulty of the attack step is added in step IV of the attack graph generation process. If authentication is not required, no skill is required by the attacker to make the connection and the difficulty level is zero. If authentication is required, the difficulty of this attack step will be determined by the type of authentication and how it is implemented. In the tool used, attack step difficulty is quantified by the expected time for a skilled attacker to complete it. This is represented by a probability distribution function which states the probability of an attacker completing the attack step in a given amount of time. The functions are based on structured interviews with experienced penetration testers, but they can be adjusted by the modeller if desired. For example, in the HV diagram of Fig. 2, the second attack step exploits a vulnerability in the web application to gain sufficient privilege on its host to make connections to further devices. The difficulty of this attack step would take into account whether there were known vulnerabilities in the web application and the existence of exploits.

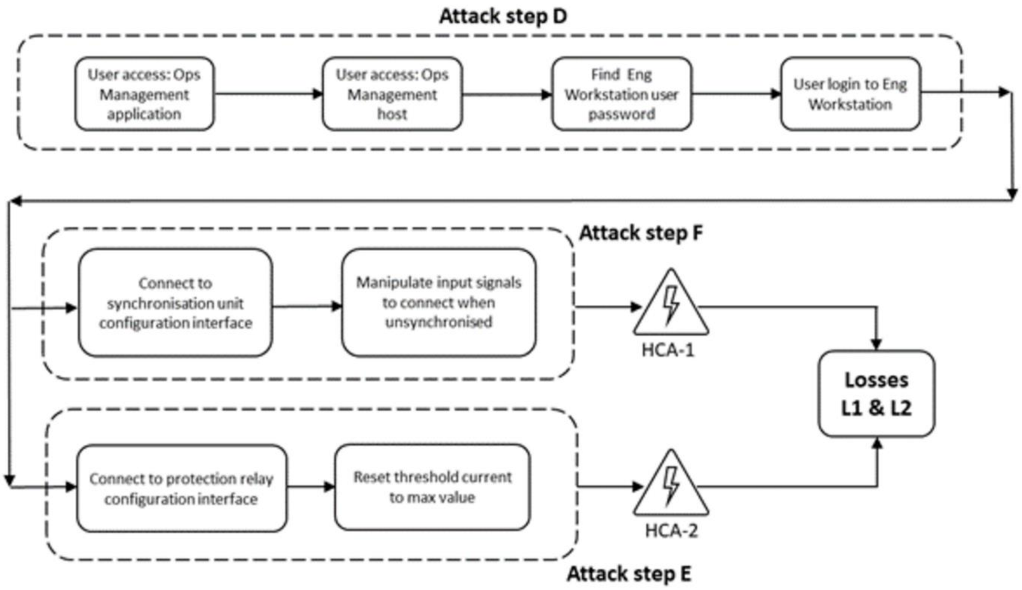

Attack step D
User access: Ops Management application
User access: Ops Management host
Find Eng Workstation user password
User login to Eng Workstation
Attack step F
Connect to synchronisation unit configuration interface
Manipulate input signals to connect when unsynchronised
HCA-1
Losses L1 & L2
Connect to protection relay configuration interface
Reset threshold current to max value
HCA-2
Attack step E

Figure 9 **Simplified segment of attack path in the control zone.**
Full-size DOI: 10.7717/peerjcs.1249/fig-9

- Steps II –IV are then repeated until no further attack steps are possible within the model. Attack steps which have been assigned an impact level because they can lead to HCAs will appear in the attack graph with these labels attached.

Once the attack graph has been generated, step 4e uses a further automated algorithm to search for the shortest paths from selected attacker entry points to HCAs; *e.g.*, closing the circuit breaker connecting the generator to the grid when they are not synchronised.

In HVA_CPS, this process is based on Dijkstra's shortest path algorithm. The path length is based upon the expected time for a skilled attacker to complete the sequence of attack steps that comprise the attack path from the attacker's entry point to the selected end points. A simplified version of a segment of the attack path from the HV diagram of Fig. 2 is shown below in Fig. 9. In practice, the system would be decomposed into greater detail than shown. The primary output from this step is whether there are exploitable attack paths that enable the attacker to complete both attack steps 5 & 6 as shown in the HV diagram of Fig. 2. If so, the secondary output is the list of attack steps needed to do so and what attacker capabilities are needed to complete them. For example, finding a new vulnerability is beyond the skills of a basic attacker and exploiting a hazardous vulnerability generally requires a detailed understanding of the CPS.

Attack graphs generated through this approach for a typical industrial control system can have hundreds or thousands of nodes and a corresponding number of attack paths. The search for the shortest paths through them often reveals nodes at which many attack paths converge and then diverge. Such nodes are known as choke points. Identifying choke points is valuable because improving their security is often among the best options for improving the security of the system as a whole.

Table 6 **Illustrative risk log entry.**

Risk Log					
Risk number	**Threat scenario**	**Hazardous vulnerability**	**Hazardous control action**	**Loss**	**Minimum needed threat actor sophistication level**
1	Threat actor sophistication & motivation as defined in icsLang	Attack path in Fig. 2	HCA-1 & HCA-2	L1 & L2	Advanced
2	…	…	…	…	…

A key benefit of the high-fidelity modelling enabled by HVA_CPS is that investment to improve system security can be targeted to where it is most effective. This can be demonstrated by summarising the output of the HVA_CPS process in a risk log, as illustrated in Table 6. Risk 1 is read as follows: 'a threat actor of sophistication level and motivation defined in MAL programme icsLang, exploits a hazardous vulnerability to complete the attack path in Fig. 2, causing both HCA-1 and HCA-2, which results in losses L1 and L2. The minimum threat actor sophistication level needed to complete the attack is Advanced, as defined in *STIX (2021)*'.

Risks are ranked according to the level of loss and the level of threat actor sophistication required to cause it. The overall system risk is then reduced by targeting security improvements on the attack paths with the lowest threat actor sophistication level and which cause the highest losses. This increases the threat actor sophistication level needed to cause the highest losses.

EVALUATION OF THE PROPOSED PROCESS—HVA_CPS

An evaluation of whether the HVA_CPS process has each of the attributes defined in Table 2 is presented below. It is summarised in Table 7.

Threat models are defined in terms of attacker capabilities and system access through the Meta Attack Language (MAL). This includes specific attack steps that the threat actor is capable of completing. An example threat model specified in MAL is the combination of coreLang and icsLang.

The knowledge needed to produce the wanted outputs of the HVA_CPS process can be captured in the system model. Common attacker tactics and techniques are captured in MAL programmes such as icsLang and vehicleLang. The MAL programme, coreLang, creates an abstract vulnerability asset which can be extended in other MAL programmes. This can be used to represent known vulnerabilities or the probability of vulnerabilities existing. This can be applied to both generic vulnerabilities and those specific to CPS. HCAs are derived through STPA and represented in the model as attack steps with a selected level of impact.

The HVA_CPS process can be applied at design stage and updated as the system evolves. In principle, updating models of operational systems could be largely automated by converting data from network monitoring systems or vulnerability scanners into the appropriate XML format. However, the authors have not demonstrated this.

Table 7 Summary of effectiveness of HVA_CPS process.

Attribute name	Effectiveness of HVA_CPS Process
Threat model	Detailed threat models can be specified in MAL programmes such as coreLang and icsLang.
Knowledge base	System models can capture information from each of the relevant knowledge bases.
Full lifecycle support	The HVA_CPS process can be applied at design stage and updated as the system evolves throughout its lifetime.
Reproducible	Conversion of the model to an attack graph and the search for the most exploitable attack paths are automated. Decisions on exactly how to model the system are still left to human judgement.
Tool support	Tools have been developed to support or automate each step of the process.
Maturity	All steps of the process have been demonstrated on a complex distributed CPS, apart from steps 4c & 4d. These have been implemented on HCAs based on a complex distributed CPS. The formalism is readily applicable to other HCAs.
High fidelity model	MAL programmes enable development of high-fidelity models which capture component vulnerabilities.
Traceability of threat to loss	Traceability is provided through identification of HCAs and attack paths that attackers can exploit to cause them. This is demonstrated in the output risk log.
Predictive power	HVA_CPS makes predictions of whether the CPS is resilient to specified threat scenarios including those targeting hazardous vulnerabilities.

The HVA_CPS process is largely repeatable and reproducible through automation of attack graph generation and the search for its most exploitable attack paths. Decisions on exactly how to model the system are still left to human judgement.

Tool have been developed to minimise the human effort and expertise needed to apply the methodology. The system was modelled using a tool designed for the purpose. The same tool automates generation of the attack graph from the system model and identifies the shortest attack paths through the model. The open-source compiler for the Meta Attack Language enables use of detailed MAL programmes. Building high fidelity models still requires significant human input.

All steps of the HVA_CPS process have been demonstrated on a complex distributed CPS, apart from steps 4c & 4d. These have been implemented on HCAs based on a complex distributed CPS. The formalism is readily applicable to other HCAs.

High-fidelity models can be developed using detailed meta-models specified in MAL programmes. Component vulnerabilities can be captured by instantiating vulnerability objects defined in a MAL programme, such as coreLang. The exploitability of vulnerabilities in the model can be represented by modifying the difficulty of attack steps that exploit them.

Traceability of threat to loss is achieved through a combination of the identification of HCAs and the attack paths that attackers can exploit to cause them. This is demonstrated in the output risk log.

A prediction of whether the CPS is resilient to specified threat scenarios is achieved through a combination of automated and human analysis. Automated analysis identifies the most exploitable attack paths. Human analysis of these reveals the hardest attack steps and hence what level of threat actor sophistication is required to complete them. For example, the attack path described in Section 3 does not require discovery of previously unknown vulnerabilities and, therefore, could reasonably be assumed to be exploitable by an advanced level threat actor, as defined in *Rocchetto & Tippenhauer (2016)*. This analysis could also be automated by ranking attack paths according to the difficulty of their hardest attack step and converting attack step difficulty to threat actor sophistication level.

HVA_CPS meets all the wanted attributes with the caveat that, although each step of the process has been applied to at least one complex CPS, the whole process has yet to be fully applied to the same CPS. The most novel step is the incorporation of HCAs into the system model, such that the automated search for attack paths to specific assets is converted to a search for hazardous vulnerabilities. This has been demonstrated on HCAs relating to a gas transmission network. The demonstration directly led to the initiation of a project to model the network control system in detail.

For a CPS that is free from hazardous vulnerabilities, HVA_CPS provides assurance of its security. For major CPSs, such as a national electricity grid or a chemical processing plant, this level of assurance is a valuable property. For CPSs that do have hazardous vulnerabilities, it enables detailed analysis of how best to remove or mitigate the risk.

A summary of how HVA_CPS compares with the processes described in selected related works is shown in Table 8.

The high level of modelling fidelity achievable in HVA_CPS is indicated by the number of parameters in the meta-model. More parameters enable a wider range of attack scenarios to be modelled in greater detail. The HVA_CPS meta-model is a descendant of those developed in *Holm et al. (2013)*; *Holm et al. (2014)* and *Ekstedt et al. (2015)*. The number of parameters in each of these meta-models is shown in Fig. 10. It is noted that P2CySeMoL had sufficient fidelity to achieve a statistically significant correlation between predicted and actual times to compromise of six systems used in a cyber defence exercise (*Holm et al., 2014*).

Through its combination of attributes, HVA_CPS has the most rigorous capability for predicting whether a distributed CPS is resilient to a specified threat scenario. This is achieved through detailed modelling of the threat scenario, the system subject to the threat, and the impact of the threat to the system. The human effort required to implement the process was greatly reduced by the use of tools designed for the purpose.

CONCLUSIONS

This article has described the need to identify hazardous vulnerabilities in Cyber-Physical Systems (CPS). The attributes needed for a rigorous Hazardous Vulnerability Analysis

Table 8 **Comparison of Selected Processes with Rigorous Attributes.**

Attributes	**Attribute category**	**Process**							
		Data driven vulnerability exploration (*Bakirtzis et al., 2020*)	Safety & security analysis (*Kriaa, 2016*)	Safety & security based on STPA & Event B (*Butler et al., 2019*)	Safe & secure model driven design (*Li, 2018*)	Attacker modelling framework (*Deloglos, Elks & Tantawy, 2020*)	Hazard driven threat modelling (*Castiglione & Lupu, 2020*)	Use of attack graphs for predicting CPS security (*Few et al., 2021*)	**HVA_CPS**
Threat model	Input	No	Basic	Basic	Basic	Advanced	Basic	Basic	**Advanced**
Knowledge base		Yes	Yes	Limited	Limited	Yes	Yes	Yes	**Yes**
Full lifecycle support	Process	Yes	Yes	Yes	Yes	Yes	Yes	Yes	**Yes**
Reproducible		Yes	Yes	Yes	Yes	Yes	Yes	Yes	**Yes**
Tool support		Yes	Yes	Yes	Yes	No	Limited	Yes	**Yes**
Maturity – demonstrated on a complex CPS		Yes	No	No	No	No	No	Yes	**Partially demonstrated**
High fidelity	Outputs	No	No	No	Yes	No	Limited	Limited	**Yes**
Traceability of threat to hazard		Limited	Partial	Partial	Partial	Partial	Yes	Partial	**Yes**
Predictive power		Medium	Medium	Low	Low	Medium	Medium	Medium	**High**

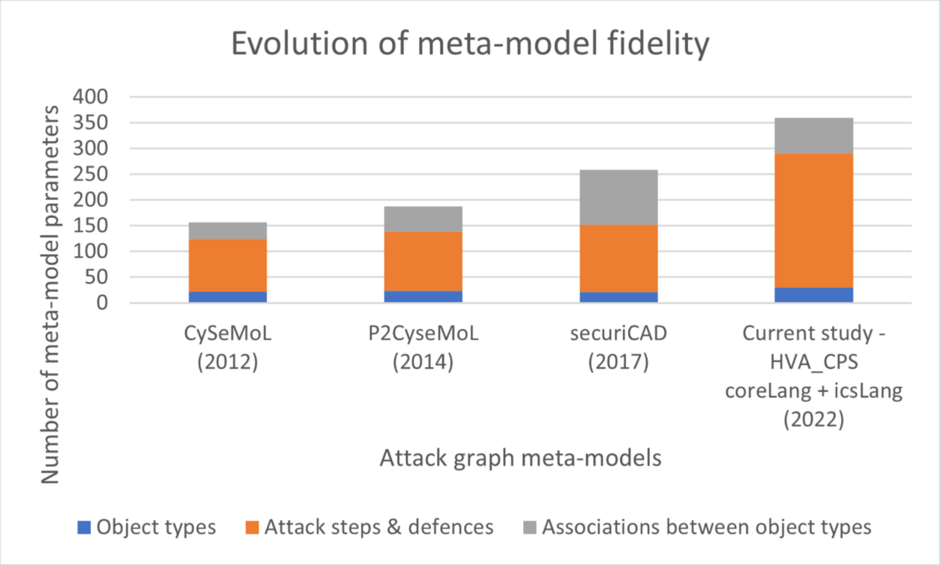


Figure 10 **Number of meta-model parameters.**
Full-size DOI: 10.7717/peerjcs.1249/fig-10

(HVA) process have been captured and justified. An informal literature search of related works has not found an existing process that meets all the wanted attributes. A novel process, HVA_CPS, has been developed combining two existing techniques: control signal analysis and attack path analysis. The link between the two techniques is through the use of hazardous control actions (HCAs). These are output from the control signal analysis and input to the attack path analysis. The method for combining these techniques has been explained using example HCAs and is readily adaptable to other HCAs. The HVA_CPS process meets all the attributes needed to reliably identify hazardous vulnerabilities and hence reduces the gap identified earlier. In particular, HVA_CPS provides traceability in a high level of detail, from threat scenarios to possible attacker actions, and to their impact upon the CPS under study. For CPSs that have no hazardous vulnerabilities, HVA_CPS gives risk owners a high level of confidence that this is indeed the case. For CPSs that do have hazardous vulnerabilities, HVA_CPS enables detailed analysis of how best to mitigate the risk.

LIMITATIONS

A limitation of this article is that HVA_CPS has yet to be applied in full to a CPS. This could be addressed through future case studies. Integration of the modelling of both the cyber and physical aspects of CPS would enable more detailed predictions of physical impact. A further opportunity for improvement is the application of centrality algorithms which search the attack graph for the system components which have most influence upon the security of the CPS as a whole. These are typically the components at which the most attack paths converge and diverge.

ACKNOWLEDGEMENTS

The bulk of the Hazardous Vulnerability Analysis in Distributed Cyber-Physical Systems research work was carried out by the Office of Gas and Electricity Markets (Ofgem) and supported by Nottingham Trent University (NTU) and the National Grid (NG) by providing supervisory resources. We would like to thank all other parties who contributed to this article. This article is not intended as relevant guidance or as state-of-the-art within the meaning of NIS Regulation 10 (3) and (4) (*MITRE Organisation, 2020*). Any reference to any organisation, service or product does not constitute or imply the endorsement, recommendation, or favouring by Ofgem or any of its employees or contractors acting on its behalf.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

The authors received funding for this work from Ofgem. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
Ofgem.

Competing Interests

Tawfik Al-Hadhrami is an Academic Editor for PeerJ. Kenny Awuson-David is employed by Ofgem. Chris Few is employed by National Grid. Alan Jamieson moved to another company, but was previously employed by Ofgem as part of this research project.

Author Contributions

- Alan Jamieson conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, authored or reviewed drafts of the article, and approved the final draft.
- Chris Few conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Kenny Awuson-David conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.
- Tawfik Al-Hadhrami analyzed the data, performed the computation work, authored or reviewed drafts of the article, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

There is no code or data for this security framework article.

REFERENCES

Ahmed M, Panda S, Xenakis C, Panaousis E. 2022. MITRE ATT & CK-driven cyber risk assessment. In: *17th International conference on availability, reliability and security*. Vienna, Austria.

Andrew BWM, Katsikeas S, Hacks S. 2022. GitHub; mal-lang/icsLang. *Available at https://github.com/mal-lang/icsLang/blob/master/src/main/mal/icsLang.mal* (accessed on 14 March 2022).

Awuson-David K. 2022. Facilitate security event monitoring and logging of operational technology (OT) legacy systems. In: *International conference of reliable information and communication technology*. Malaysia: 461–472.

Awuson-David K, Al-Hadhrami T, Alazab M, Shah N, Shalaginov A. 2021. BCFL logging: an approach to acquire and preserve admissible digital forensics evidence in cloud ecosystem. *Future Generation Computer Systems* **122**:1–13 DOI 10.1016/j.future.2021.03.001.

Bakirtzis G, Simon B, Collins A, Fleming C, Elks C. 2020. Data-driven vulnerability exploration for design phase system analysis. *IEEE Systems Journal* **14(4)**:4864–4873 DOI 10.1109/JSYST.2019.2940145.

Baybutt P. 2015. A critique of the Hazard and Operability (HAZOP) study. *Journal of Loss Prevention in the Process Industries* **33**:52–58 DOI 10.1016/j.jlp.2014.11.010.

Bronk C, Tikk-Ringas E. 2013. The cyber attack on Saudi Aramco. *Survival, Global Politics and Strategy* **55(2)**:81–96.

Butler M, Howard G, Colley J, Sassone V. 2019. A methodology for assuring the safety and security of critical infrastructure based on STPA and Event-B. *International Journal of Critical Computer-Based Systems* **9**:1–2.

Carnegie-Mellon University, Software Engineering Institute. 2006. The Architecture Analysis & Design Language (AADL): an introduction. US Defence Technical Information Centre, Virginia, USA.

Castiglione LM, Lupu EC. 2020. Hazard driven threat modelling for cyber physical systems. In: *Joint workshop on CPS & IoT security and privacy (CPSIOTSEC'20)*. Orlando, Florida, USA.

Cherdantseva YV, Burnap P, Blyth A, Eden P, Jones K, Soulsby H, Stoddart K. 2016. A review of cyber security risk assessment methods for SCADA systems. *Computers & Security* **56**:1–27.

CNN. 2007. Staged cyber attack reveals vulnerability in power grid, 23 September 2007. *Available at https://www.youtube.com/watch?v=fJyWngDco3g* (accessed on 15 November 2021).

Deloglos C, Elks C, Tantawy A. 2020. An attacker modeling framework for the assessment of cyber-physical systems security. In: *SAFECOMP; computer safety, reliability, and security*. Lisbon, Portugal.

Dijkstra EW. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* **1**:269–271 DOI 10.1007/BF01386390.

DiPinto A, Dragoni Y, Caracano A. 2018. TRITON: the first ICS cyber attack on safety instrument systems. In: *Black Hat*. USA: Nozomi Networks.

Dolev D, Yao A. 1983. On the security of public key protocols. *IEEE Transactions on Information Theory* **29(2)**:198–208 DOI 10.1109/TIT.1983.1056650.

Dunjó J, Fthenakis V, Vilcheza JA, Arnaldos J. 2010. Hazard and operability (HAZOP) analysis. A literature review. *Journal of Hazardous Materials* **173(1–3)**:19–32 DOI 10.1016/j.jhazmat.2009.08.076.

Ekstedt M, Johnson P, Lagerstrom R, Nydren J, Gorton D, Shahzad K. 2015. Securi CAD by Foreseeti: a CAD tool for enterprise cyber security management. In: *IEEE 19th international enterprise distributed object computing workshop*. Piscataway: IEEE.

Few C, Thompson J, Awuson-David K, Al-Hadrami T. 2021. A case study in the use of attack graphs for predicting the security of cyber-physical systems. In: *International congress of advanced technology and engineering*. Taiz, Yemen.

Geismann J, Bodden E. 2020. A systematic literature review of model-driven security engineering for cyber—physical systems. *Journal of Systems and Software* **169**:110697 DOI 10.1016/j.jss.2020.110697.

Greenberg A. 2020. How 30 Lines of Code Blew Up a 27-Ton Generator, Wired, 23 October 2020. *Available at https://www.wired.com/story/how-30-lines-of-code-blew-up-27-ton-generator/* (accessed on 15 November 2021).

Greenberg A. 2019. Sandworm: a new era of cyberwar and the hunt for the kremlin's most dangerous hackers, Anchor Books.

Hacks S, Katsikeas S. 2021. Towards an ecosystem of domain specific languages for threat modeling. In: *International conference on advanced information systems engineering*. Melbourne, Australia.

Hacks S, Katsikeas S, Ling E, Lagerström R, Ekstedt M. 2020. powerLang: a probabilistic attack simulation language for the power domain. *Energy Informatics* **3**:30 DOI 10.1186/s42162-020-00134-4.

Holm H, Shahzad K, Buschle M, Ekstedt M. 2014. P2CySeMoL: predictive, probabilistic cyber security modeling language. *IEEE Transactions on Dependable and Secure Computing* **12(6)**:626–639.

Holm H, Sommerstad T, Ekstedt M, Nordström L. 2013. CySeMoL: a tool for cyber security analysis of enterprises. In: *22nd International conference and exhibition on electricity distribution (CIRED 2013)*. Stockholm, Sweden.

Ishimatsu T, Leveson NG, Thomas J, Katahira M, Miyamoto Y, Nakao H. 2010. Modeling and hazard analysis using STPA. In: *4th IAASS conference, making safety matter, Ishimatsu, Takuto; Leveson, Nancy G.; Thomas, John; Katahira, Masafumi; Miyamoto, Yuko; Nakao, Haruka*.

Johnson P, Lagerström R, Ekstedt M. 2018. A meta language for threat modeling and attack simulations. In: *Availability, reliability and security*. 38. Hamburg, Germany: ACM, 1–8.

Katsikeas S, Hacks S, Johnson P, Ekstedt M, Lagerström R, Jacobsson J, Wällstedt M, Eliasson P. 2020. An attack simulation language for the IT domain. In: *International workshop on graphical models for security*. Boston, USA.

Katsikeas S, Johnson P, Hacks S, Lagerström R. 2019. Probabilistic modeling and simulation of vehicular cyber attacks: an application of the meta attack language. In: *5th international conference on information systems security and privacy, Prague; Czech Republic*.

Khamespanah E, Sirjani M, Kaviani ZS, Khosravi R, Izadi M-J. 2015. Timed Rebeca schedulability and deadlock freedom analysis using bounded floating time transition system. *Science of Computer Programming* **98**:184–204 DOI 10.1016/j.scico.2014.07.005.

Kriaa S. 2016. *Joint safety and security modeling for risk assessment in cyber physical systems*. Paris, France: Université Paris Saclay.

Kriaa S, Bouissou M, Laarouchi Y. 2015. A model based approach for SCADA safety and security joint modelling: S-Cube. In: *10th IET system safety and cyber-security conference*. Bristol.

Kriaa S, Pietre-Cambacedes L, Bouissou M, Helgrand Y. 2015. A survey of approaches combining safety and security for industrial control systems. *Reliability Engineering and System Safety* **139**:156–178 DOI 10.1016/j.ress.2015.02.008.

Langner R. 2011. Stuxnet: dissecting a Cyberwarfare Weapon. *IEEE Security & Privacy* **9**:49–51.

Lee EA. 2008. Cyber physical systems: design challenges. In: *11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC)*. Piscataway: IEEE.

Li L. 2018. *Safe and secure model-driven design for embedded systems*. Paris, France: Université Paris-Saclay.

Lyu X, Ding Y, Yang S. 2019. Safety and security risk assessment in cyberphysical systems. *IET Cyber-Physical Systems: Theory & Applications* **4(3)**:221–232 DOI 10.1049/iet-cps.2018.5068.

M. P. 2018. Rating hackers, rating defences, 6 September 2018. *Available at https://www.ncsc.gov.uk/blog-post/rating-hackers-rating-defences* (accessed on 26 August 2021).

MITRE Organisation. 2020. ATT & CK® for Industrial Control Systems, MITRE, 7 January 2020. *Available at https://attack.mitre.org/docs/ATTACK_for_ICS_Philosophy_March_2020.pdf* (accessed on 15 August 2021).

Mohamed MA, Challenger M, Kardasa G. 2020. Applications of model-driven engineering in cyber-physical systems: a systematic mapping study. *Journal of Computer Languages* **59**:100972 DOI 10.1016/j.cola.2020.100972.

Mohamed MA, Kardas G, Challenger M. 2021. Model-driven engineering tools and languages for cyber-physical systems—a systematic literature review. *IEEE Access* **9**:48605–48630 DOI 10.1109/ACCESS.2021.3068358.

Moradi F, Abbaspour Asadollah S, Sedeghatbaf A, Čaušević A, Sirjani M, Talcott C. 2020. An actor-based approach for security analysis of cyber-physical systems. In: *Formal methods for industrial critical systems*. Vienna, Austria: Springer.

Nguyen PH, Kramer M, Klein J, Le Traon Y. 2015. An extensive systematic review on the model-driven development of secure systems. *Elsevier Information and Software Technology* **68**:62–81 DOI 10.1016/j.infsof.2015.08.006.

Ou X, Boyer WF, McQueen MA. 2006. A scalable approach to attack graph generation. In: *13th ACM conference on computer and communications security*. Virginia, Alexandria, USA.

Pfeffer A. 2009. Figaro: an object-oriented probabilistic programming language. Charles River Analytics Technical Report.

Pietre-Cambacedes L, Deflesselle Y, Bouissou M. 2011. Security modeling with BDMP: from theory to implementation. In: *Conference on network and information systems security*. La Rochelle, France.

Potvin M. 2019. The AURORA vulnerability: the sword of Damocles over the head of rotating machines. In: *CIGRE Canada conference*. Montreal, Canada.

Rocchetto M, Ferrari A, Senni V. 2019. Challenges and opportunities for model-based security risk assessment of cyber-physical systems. In: *Resilience of cyber-physical systems. Advanced sciences and technologies for security applications*. Cham: Springer, 25–47.

Rocchetto M, Tippenhauer NO. 2016. On attacker models and profiles for cyber-physical systems. In: *European symposium on research in computer security*. Heraklion, Greece.

Shandilya V, Simmons CB, Shiva S. 2014. Use of attack graphs in security systems. *Journal of Computer Networks and Communications* **2014**:818957 DOI 10.1155/2014/818957.

Structured Threat Information eXpression (STIX). 2021. STIX/Resources, 20 May 2021. [Online]. *Available at https://oasis-open.github.io/cti-documentation/resources#stix-21-specification* (accessed on 15 August 2021).

Vernotte A, Valja M, Korman M, Bjorkman G, Ekstedt M, Lagerstrom R. 2018. Load balancing of renewable energy: a cyber security analysis. *Energy Informatics* **1**:5 DOI 10.1186/s42162-018-0010-x.

Williams TJ. 1994. The Purdue enterprise reference architecture. *Computers in Industry* **24(2–3)**:141–158 DOI 10.1016/0166-3615(94)90017-5.

Yadav T, Mallari RA. 2015. Technical aspects of cyber kill chain. In: *International symposium on security in computing and communications*. Kochi, India.

Young W, Leveson N. 2014. An integrated approach to safety and security based on systems theory. *Communications of the ACM* **57(2)**:31–35.