Towards the Definition of a Security Incident Response Modelling Language

Myrsini Athinaiou[1], Haralambos Mouratidis[1], Theo Fotis[1], and Michalis Pavlidis[1]

Centre for Secure, Intelligent and Usable Systems, University of Brighton, UK

Abstract. This paper presents a cyber-physical systems modelling language for capturing and describing health-based critical infrastructures. Following this practice incident response plan developers are able to model and reason about security and recovery issues in medical cyber-physical systems from a security requirements engineering perspective. Our work builds upon concepts from the Secure Tropos methodology, where in this paper we introduce novel cyber-physical concepts, relationships and properties in order to carry out analysis of incident response plans based on security requirements. We illustrate our concepts through a case study of a radiological department's medical cyber-physical systems that have been infected with the WannaCry ransomware. Finally, we discuss how our modelling language enriches security models with incident response concepts, guiding plan developers of health-based critical infrastructures in understanding cyber-physical systems vulnerabilities and support decision making at a tactical and a strategic level, through semi-automated secure recovery analysis.

Keywords: Cyber-physical systems modelling language; Meta-model; Incident response; Security requirements engineering

1 Introduction

In this paper we present a Security Incident Response Modelling Language (SIRML) to capture incident response (IR) concepts. When a hacker is able to damage medical equipment by taking control of devices, the best a health care organization can hope for is that the cyber-physical attack will be identified early, the medical systems will shut down without causing further damages/harm and normal operation will be restored in short-time. When attacks have occurred, decisions cannot be made on the go. There is a need for a structured approach to thwart adversarial attempts making changes to systems and their components. Because, once the security perimeter of the corporate infrastructure is penetrated, the industrial systems are exposed to attacks that will be generated from inside an organization. Incident response planning is also important to become more automated and dynamic, as attackers can use common responses as triggers for further attacks [5].

This work builds upon the Secure Tropos [18] modelling language and extends concepts, attributes and relationships to allow the cyber-physical aspect to be

represented along with the view of IR as integral part of security. The SIRML is part of a broader research effort to create a methodology which will allow the modelling of health-based IR, supporting safety and security constraints, prioritizing to protect patients. This paper contributes to the current body of knowledge by **(C1)** providing a meta-model for cyber-physical systems (CPS), aligning concepts of security and IR, **(C2)** defining concepts, attributes and relationships to model at an operational, tactical and strategic level secure IR plans and **(C3)** presenting the graphical notation that can be used to instantiate the SIRML in order to facilitate further analysis, common understanding and assessment of recovery plans from heterogeneous incident response teams (IRTs), which can consist of managers, medical equipment technicians, engineers, IT sub-teams and security experts.

The remaining of the paper is structured as follows. Section 2 introduces the secure recovery meta-model and explains in more detail the newly introduced concepts, relationships and notation. In Section 3 a case study is discussed to demonstrate how the SIRML can be utilized. In Section 4 the related work is briefly presented. Finally, while Section 5 concludes the paper by summarizing its content and contributions, it also discussing language limitations and future research directions.

2 The Secure Incident Response Modelling Language

In this section the SIRML, that enables the modelling of CPS IR plans from a security requirements engineering point of view, is presented. Firstly, the meta-model of the secure recovery language is introduced, which extends and builds upon the Secure Tropos modelling language [18]. The current version of the SIRML meta-model is shown in Fig. 1. To distinguish the inherited from Secure Tropos concepts from those newly introduced, colours are used. The concepts of the model that are white are used as in Secure Tropos and their definitions can be found in [18], unless specified otherwise. On the other hand, the novel secure recovery concepts are coloured with green (bronze), grey (silver) and yellow (gold). The different colours indicate the recovery stage that a plan refers to and range from operational to tactical and strategic.

2.1 SIRML Introduced Concepts and Attributes

We now present our security IR extensions, highlighting each addition of a concept through a definition, description and explanation of the main reason that needs to be introduced at a metamodelling level. Furthermore, in this subsection we present attributes in order to describe an instance of a concept in more detail. Attributes allows the IRT to provide both abstract and technical details to instances of a concept, therefore supporting varying degrees of granularity when describing the system or/and infrastructure under design. The range of values of the attributes of the SIRML concepts are shown in Figure 2.

Fig. 1. The Secure Incident Response Language Meta-model.

Cyber Resource is a digital entity that is part of an infrastructures function that includes different types of software (i.e. operational system, forensic toolkits and digital badges). At a detection level, it can indicate the source of an attack precisely. It is a necessary concept for the designation of duties to IT subteam. *Mission* represents a sub-goal that cannot fail during an attack due to its criticality for a CI. This concept is helpful for an IRT to prioritize resources at a tactical recovery stage. *Physical Resource* stands for the kinetic aspect of a system which can include controllers, switches, actuators, timers, field devices, active medical devices, meters and similar entities. At a detection level it can assist the inspections planning of the engineering sub-team of an IRT. This concept is necessary for the designation of duties to engineering sub-teams with different areas of specialization e.g. mechanical, electrical, biomedical.

Incident stands for the intentional unauthorized access to a system, service or resource of a CI or the compromise of a system's security properties (i.e. CIA triad, CO2 triad). This concept differentiates an event and a threat clarifying that an incident is successful and has malicious intent. Aggregates one or more *attack*s as attempts to exploit a vulnerability and together they can constitute an incident. From there, the planning can face the particular aspects of an incident and examine its possible propagation in order to identify the root cause and predict any propagation attempts and cascading effects. *Detection Mechanism* includes the security means used primarily from an infrastructure within a particular threat scenario for detection purposes.It allows IRTs to design how they are going to collect information for a possible attack in order to support management decisions. *Mitigation Mechanism* is used for security means that are utilized primarily from an infrastructure within a particular threat scenario for mitigation purposes. It enables IRTs to plan relocation, isolation of resources, conducting impact assessment and signing to each resource a priority towards the stabilization of an attack. *Recovery Mechanism* is for security means used primarily from an infrastructure within a particular threat scenario for recovery purposes. It permits IRTs to model the uncontaminated, replaced, reintegrated resources and the means needed for those action to be achievable.

2.2 SIRML Introduced and Extended Relationships

In this subsection we outline the relationships linking together concepts from recovery and security requirements engineering domains. We now define the relationships in our modelling language by building upon the meta-model in the previous section.

Specialization of Mission: A *mission* is a critical part of an *activity* that needs to be protected even when an incident in the form of a cyber-physical attack has been successful. When recovery for an individual responder is designed this *mission* can become more fine-grained. This becomes particularly important in the long term planning and the concept of *strategic mission*. Based on this concept *activities* and relocation of resources can be modelled along with other decisions and specifying the actors that will perform them, showing the different time

interval for the applicability of such recovery plans, focused on security. Hence, the specialization of a mission is represented with an inheritance relationship.

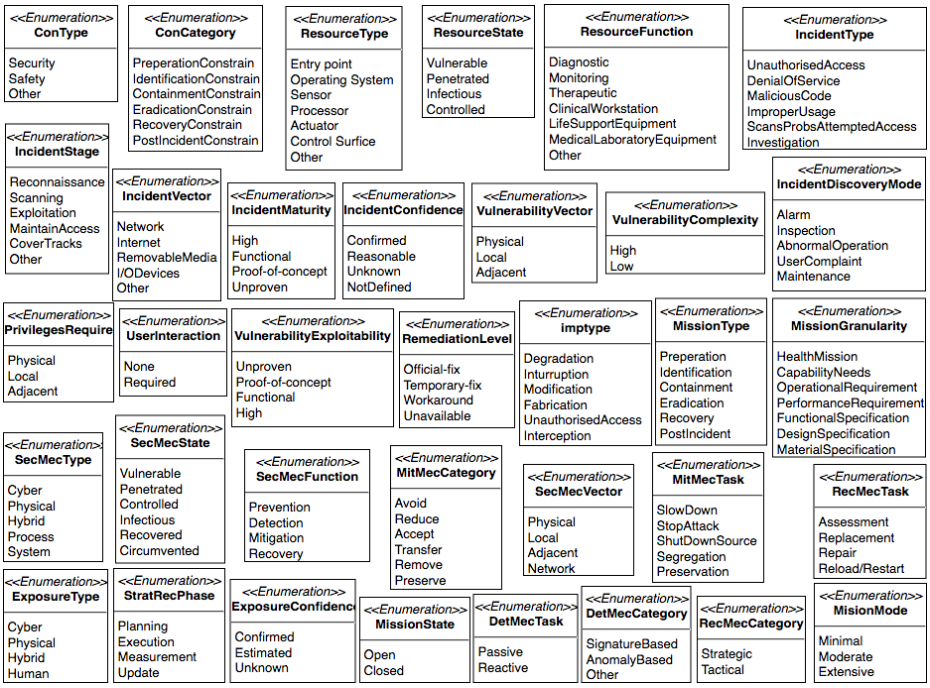

Fig. 2. The Secure IR Concepts' Attributes.

Specialization of a Security Mechanism: A *security mechanism* can be decomposed to *detection mechanism*, *mitigation mechanism* and *recovery mechanism*. These subcategories of *security mechanisms* have different functions within an incident response plan and its phases. These phases are commonly referred as preparation, identification, containment, eradication, recovery, lessons learned and should not be confused with the different child concepts of security mechanism. To indicate this sharing of common behaviours among detection, mitigation, recovery and security mechanisms we depict there relationships using the inheritance arrow.

Resource Reflexive Association: The reflexive association among resources occur as the *resource* class can perform different tasks and forms various associations among concepts of the same class, as well as other classes. For example, a medical device, which is a form of resource, works in a hospital and can be an X-ray scanner, a defibrillator, an anesthetic machine, a medical laser or a medical image storage device. If the X-ray scanner is storing images in a medical image storage device then there relationship could be modelled as two instances of the same class that communicate with each other.

Incident Reflexive Association: Incidents are referring to cyber-physical attacks that commonly described as been wrapped, usually inside exploit kits. The execution of one attack triggers a series of other attacks. In some cases they multiply and spread. In order for the modelling language to able able to show that an *incident* can associate, encapsulate, support or generate another *incident*, the reflexive association is used. For instance, a ransomware replicates itself in crucial locations of a system as for instance when it reboots and displays the ransom message. In this case the incident reflexive association will be used to connect the ransomware.exe with its copies in AppData, Start menu and root directory.

Exploits: The exploits relationship indicates that the *incident* concept connects with a *vulnerability* through this form of relationship. This relationship can specify if the actual attack is preventable. A *vulnerability* can be the target of one or more *incidents* due to the existence of relationships from a range of attacks, indicating that an *incident* aims to exploit a vulnerability and this attempts are expressed through the relationship *exploitss*. For instance, the WannaCry attack was exploiting with the EternalBlue exploit tool the vulnerability CVE-2017-0144 and with the DoublePulsar attack with related vulnerabilities CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0147 and CVE-2017-0148.

2.3 SIRML graphical and textual notation

In this section we present the visual notation of the modelling language. The diagramming notation is considered an important aspect of the modelling language as it can facilitate communication and assist problem solving [9]. The graphical notation is visualized using a set of diagrammatic components, where each concept in the modelling language is mapped to an unique visual vocabulary in the form of a notation. The shapes of the visual vocabulary were chosen based on the theory of the "physics" of notation [17]. The instance syntax is further discussed as it offers a textual encoding of the concrete syntax which provides an one-to-one mapping of a concept from our meta-model to an instance of a concept. The purpose of the instance syntax is to provide a formal representation of concept instances, in a machine readable format in order to perform analysis on recovery models through tool support. Thus, the instance syntax allows the unambiguous encoding of concepts in a textual format, which describes the instantiated concepts from a secure recovery model to facilitate security analysis within the context of recovery. The visual notation of the SIRML is shown in Figure 3.

Starting from the parent concept of *mission* and its child the *strategic recovery mission*, both are represented diagrammatically with the same shape, shown in Figure 3, but the child differentiate from the parent concept with the use of a line as part of the shape. Here another important difference can be noticed in the instance syntax. The *mission* description can take the form *MIS(DES,GR,TER,MOD,ST)*. In the instance notation DES gives the description of a mission, GR clarifies the granularity, TER specified the terrain or defer-

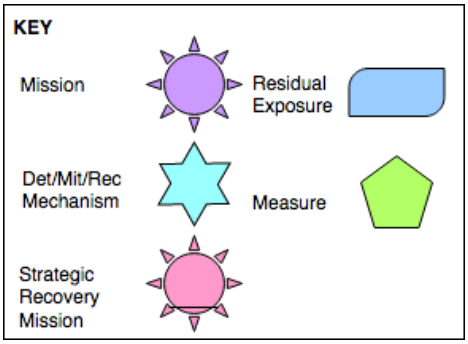

Fig. 3. Key of graphical notation extensions

ently the area/aspect analyzed, MOD specifies the mode of the mission and ST explicitly states the state of a mission. For example, a DES can be mitigation of ransomware attack, GR operational requirement, TER hospital, MOD extensive and ST open. This can be expressed from an instance syntax as *MIS(ransomware attack mitigation, operational requirement, hospital, extensive, open)*. In a similar manner the concept *strategic recovery mission* can be expressed in the form of an instance syntax as *STRMIS(PH,DES)*. Here the PH denotes the phase of a particular recovery mission and the DES describes in more detail the strategic recovery mission context. For instance, a strategic recovery mission can be at the PH of planning and in more detail its DES can be among the lines of patch vulnerability planning. This will generate an instance syntax *STRMIS(planning, patch vulnerability planning)*.

Another important concept in the modelling language is that of security mechanisms, borrowed from Secure Tropos and extended to the specialized detection, mitigation and recovery mechanisms. These types of mechanisms are useful at different stages of security that relate directly with the broader scope of incident response. The graphical syntax is presented in Figure 3 and their generic instance syntax can be described as *SECMEC(CAT,TYP,DES,NAM,ST, VEC,FUN,COS)*. Based on this syntax CAT stands for the category, TYP for the type of the security mechanism, DES for the description as in the rest of the concepts that have been explored, NAM for the name of the specific mechanism, ST for the state that the mechanism is assessed to be while under conditions of a cyber-physical attack, VEC is referring to the vector and FUN to the function that a mechanism primarily performs. Whereas this instance syntax take the form of *RECMEC(DES,TYP,CAT,NAM,VEC,FUN)* for the recovery mechanism and remains pretty similar for the rest of the security mechanisms.

Secure Tropos does not use risk as one of its concepts. However, a recovery plan needs to incorporate the concept of risk. This takes the form of the *residual exposure*. As a recovery plan attempts to face risks deriving from a cyber-physical attack, even after planning an IRT needs to be aware of what are the remaining areas of exposure for a system/infrastructure that is examined. In instance syn-

tax terms this concept takes the form *RE(DUR,TYP,CON)*, where DUR stands for the estimated duration that this exposure remains, TYP is the type of the exposure and CON denotes the confidence that such an attack will take place. The graphical representation is a blue rectangle with sharp and rounded corners. An example can be remaining malware code that was in the system for sixteen hours before detection. The type of threat is cyber as it has the form of software and there is actual confirmation that this exposure has occurred. In an instance syntax the same concept will be expressed as *RE(16 hours, Cyber, Confirmed)*.

The final concept that is modelled is the *measure*. The measure has as instantiation syntax *MET(LEGC,RESC,LREV,TRC,QREC)*. All the attributes represent complex types that are specified in the reasoning support of the framework that due to space limitations is not included in this paper. LEGC stands for the legal costs, RESC for the costs of the resources used or planned to be used, the LREV for the revenue that has been lost due to recovery processes, TRC for the trust that has been lost due to the cyber-physical attack including the employees of the CI and QREC for the quality assessment of an incident response plan. The graphical representation of the *measure* concept is shown in Figure 3.

3 WannaCry Case Study

In this paper we present a case study based on the WannaCry ransomware attack, which in 2017 affected CIs at a global scale [16]. According to the National Audit Office [20] *"at least 81 out of the 236 trusts across England"* were affected either because the attacked had infected them or because they disconnected/shut down resources to prevent a possible exploitation or aggravation of the attack [20]. As a result, *"603 primary care and other NHS organisations were also infected, including 595 GP practices."* It is important here to stress that this attack was not targeting health-care. Still the implications were disruptions of health-based CI's normal operational workflow that resulted to cancelled appointments for patients and the necessity to redirect ambulances [16]. This attacks have the capabilities to impact MCPS when looking for specific targets like unpatched legacy operating system devices.

Based on this attack, a fictional scenario is examined where a hospital's network has been hit from a datalocker ransomware attack. The attack was detected when a hospital employee who saw a pop-up message in a hospital's workstation, warning him/her that the computer is encrypted and a ransom needs to be paid before a timer expires. The hospital is cyber-aware and does not by its policy pay ransom. However, the hospital is conscious of the need to recover from the ransom-ware attack. SIRML approaches recovery at three levels, generating interconnected but different views of an event/incident. These views are: (a) **Operational Secure Recovery View (OSRV):** It instantiates the preventive measures that a CI uses along with processes, tools and mechanisms to identify that an event/incident is occurring. (b) **Tactical Secure Recovery View (TSRV):** It models the defensive posture of a CI in order to contain an

attack and/or eradicate it. (c) **Strategic Secure Recovery View (SSRV):** It models the after incident activities that take place in order to restore and recover from an attack. Moreover, it initiates a lessons learned process that feedback the OSRV and TSRV.

Due to space limitations, we have modelled partially the SSRV of the *mission* to recover from ransomware attack. The SSRV instance in Figure 5, was generated using the SIRML proposed in this paper. The strategic aspect of recovery consists of the longer planning and execution; it is supported by measures of events/incidents/attacks and improves with feedback the overall recovery plan. Firstly, we describe actors and their roles, such as *"IRT"* who plays a substantial role for the achievement of the overarching strategic recovery mission of managing operating systems through their life-cycle and in particular in terms of patching activities and bugfixes that relate even closer to security. We model the vulnerable resources identified individually in the TSRV within sets of resources that share common characteristics relevant to a long-term recovery plan. An example resource set, in our case study (see Fig5), contains *"legacy Windows operating systems"*.

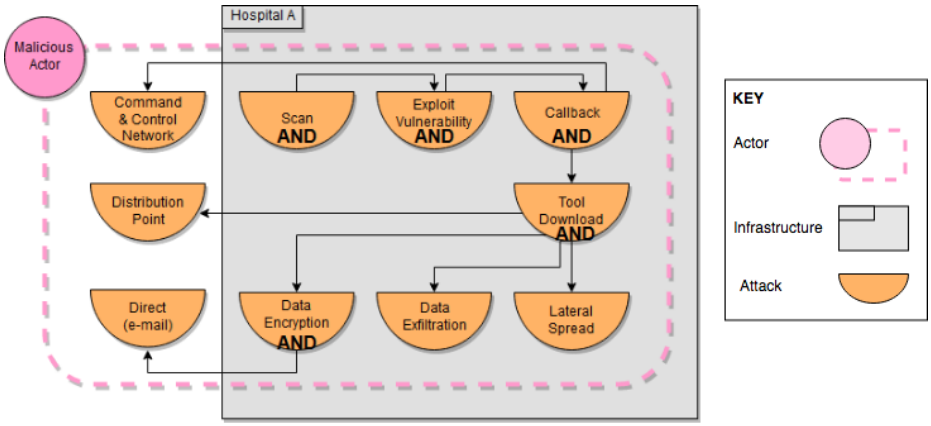

Fig. 4. General Ransomware TIRV

For a longer time frame of planning recovery activities and assessment of applied recovery plans, measures can guide the overall process. These measures can exist at an actor level, like the timelines of bugfixes and at an infrastructure level, such as quality measures for ransomware tactical recovery, which is affected from the effectiveness with which the IRT has collaborated with external to the Hospital H third parties, that were presented at the OSRV. But, a secure recovery needs also to consider the residual exposure from an attack. In the case of WannaCry, malware might be still within the hospitals network. Additionally, backdoors might persist in medical equipment. This can iteratively feedback the secure TSRV that can be enhanced to incorporate these newly identified residual

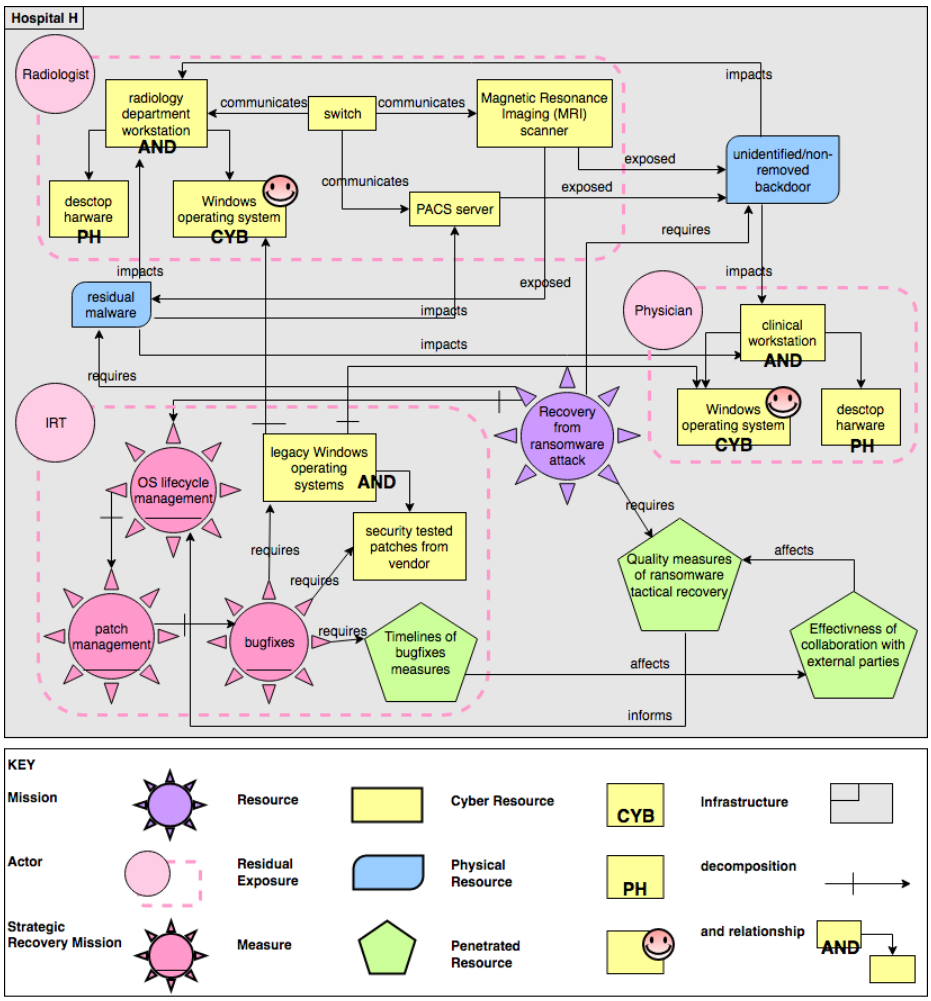

Fig. 5. WannaCry partial strategic secure-recovery view

threats in its containment policy and defensive posture. A represenatation of the ransomware is given in Fig. 4 as part of the TSRV.

The contribution here is that by associating recovery properties related to cyber-physical attacks with security concepts, we are able to refine the specific recovery needs of each resource and resource set including their dependencies at a modelling level. Thus, the benefit is three-fold: firstly, by using the SSRV, the recovery team can debrief, argue and reason about their decisions that relate to longer term plans. Their explanation can be based on Measures and their secure recovery performance can be assessed from Residual Exposure related attributes. For example, attacks against CIs like WannaCry, Petya/GoldenEye and ExPetr are examples of ransomware attacks that they have caused harm except from the NHS, UK's public health services provider, also to unnamed medical facilities in the U.S. [8], the Heritage Valley Health System, that runs hospitals and care facilities in Pittsburgh [10] and to Merck one of the largest pharmaceutical companies at a global scale [6]. If WannaCry SSRV indicating the residual exposure from the application of patches would have been understood, then an enhanced TSRV would have been modelled and applied to cover the vulnerable MCPS from attacks that spread in multiple ways. Meaning, that attacks like Petya/GoldenEye and ExPetr could be avoided or mitigated more effectively before they cause damage.

Secondly, the *OualityOfRecoveryPlan* could be estimated based on predefined indicators, sampling and application of relevant tools and methods. In this manner, a quantification approach to more quality-based requirements can be attempted. Besides, as this analysis will take place at a requirements stage, an estimation of which strategic plan is suitable but also affordable for an infrastructure can be found, in comparison with other alternatives that have as focal point the proposed conceptual modelling language.

Thirdly, our SIRML allows forensic teams to apply ICS-CERT [23] recommendations by collecting date, times, mitigation, response and recovery plans applied along with the particular resources involved. The forensic team can be in a position to decide based on these concepts from what active CPS data have to be captured before a tactical plan is applied and how to shut down a CPS to preserve forensic evidences. It also supports then to co-examine heterogeneous security mechanisms and decide when such a mechanism can change the environment of forensic interest in a way that will impede discovery. For example an anti-virus run might cause such modifications. Moreover, changes in other resources can have the same effect at a hardware, software and firmware level. By creating a recovery plan, IRTs can consult forensic investigators at a design level as of what actions might hinder a forensic inquire and identify those recovery processes that need to be supported from human resources with forensic expertise.

Overall by using all three views we are able to model in detail the data required for secure recovery at an organizational, tactical and strategic level, how the data permeates through the physical components of an MCPS, the jurisdiction due to the geographical location, and the specific information of the

enabling cyber components including networking and communications. Based on the information elicited through these concepts, we are then able to model the secure recovery concepts such as vulnerabilities in an MCPS along with threats, security constraints and detection, response and recovery plans through security mechanisms and recovery activities, derived from health-based goals and activities. Thus, our work provides a secure recovery modelling language for health-based CIs enabling IRTs to express and model MCPS recovery needs, understanding the close interconnection between MCPS and a patient's care.

4 Related Work

Many security requirements engineering (SRE) methodologies have been proposed with increasing intensity the last fifteen years. A number of these methodologies use as fundamental components the concepts of assets and their associated risks. Mead et al. published the Security Quality Requirements Engineering (SQUARE) methodology that utilises a broad spectrum of artefacts, such as use cases, misuse cases and attack trees [12]. The work of Mellado et al. proposed the Security Requirements Engineering Process (SREP) using the Common Criteria (ISO/IEC 15408) for the software life-cycle [14]. Mouratidis and Giorgini introduced the Secure Tropos approach for security and trust [18]. Compagna et al. connected the legal requirements with security and privacy, identifying patterns [4] and examined the relation of security with safety requirements [7]. Other researchers extended Secure Tropos for risk management [11]. Threat modelling was also used for security requirements with an orientation towards the role of attacker as an actor [21]. Other researchers developed methodologies driven by misuse cases [25] and reuse cases [24]. Yu et al. has also proposed a social ontology to connect software engineering with security[28]. MITRE's knowledge base and model, abbreviated as ATT&CK^{TM} expresses attack phases [15] and STIX^{TM} is a language to gather intelligence regarding an attack, such as suspected compromise and perpetrators, courses of actions and legitimate resources used for malicious purposes [19]. These approaches are indeed very important and focus on the elicitation and analysis of security requirements and analysis, they do not explicitly consider recovery concepts which can range from detection to mitigation and actual recovery. The lack of clearly incorporating security requirements with recovery limits the scope and coverage of the multidimensional character of security. Our approach attempts to provide an alignment of recovery with security that will enable IRTs to be more versatile and ready to face cyber-physical attacks, which initiate and propagate from the cyber domain, but are impacting and are triggered from the physical world.

The literature also provides a large body of resources from the domain of recovery engineering. A sustainable amount of works relevant to recovery are concerned with the traceability of code to requirements documentation [1][27]. Business process languages have been used to analyze the relevant concepts and identify potential gaps [22]. The disaster management meta-model (DMM) proposed by Othman and Beydoum, follows the NIST segmentation of recovery

stages [21]. Bareiss et al. introduced a model for failure recovery in order to minimize downtime of manufacturing systems [2]. Chen et al. have examined path-based failures [3] and Zhu et al. have used mathematical models to analyze failure and recovery strategies [29]. Mead proposed an approach for survivable systems separating their usage into legitimate and intrusive [13]. Hwang et. al. have presented work for failure handling of grid work-flow [26]. These works mainly provide advances related to safety requirements from hazards rather than security issues related to attacks. Therefore, they have limited role in supporting the identification of recovery requirements as part of a security-based design. Although, in some cases, they might combine safety with security at a quantitative, qualitative requirements base, they do not express how security can be an attack vector nor how recovery can assist an attack to aggravate. Also, although these methodologies are useful for the generation of specifications for cyber-physical systems they usually do not consider the human factor in the loop, that might be needed when everything else has failed and can also be the one initiating the security failures.

5 Conclusion

The proposed secure recovery modelling language in this paper enables IRTs to plan as part of their normal operations, in long and short time-frames, how they can securely recover under attack conditions, without causing more harm. We have defined a security modelling language to capture recovery and cyber-physical systems concepts (C1 and C2). The detailed relationships and attributes give to the language representational capabilities. By using the graphical notation (C3) in the case study, a shared understanding of a CPS and how it can recover can be designed and shared. The textual notation that underpins the graphical syntax allows decision-support with semi-formal representation of the language components towards automation, providing models in a machine-readable format. Reflecting on the current stage of this work, current limitations are the health-care focus and cyber-physical attacks modelling. This though does not mean that the modeling language is not suitable for other critical infrastructures, as its specialization is mainly at the level of attributes. From a social engineering attacks point of view, they can be modeled currently under the concept *Attack* as in the case study with the direct e-mail, along with other types of attacks. The focus remains though on the cyber-physical incidents. It is also necessary to provide in the future a systematic analysis that will structure the currently undefined implementation and usage of the language. Future work will emphasize on the semi-automated analysis and the formation of a process that will generate through instances of the SIRML optimal secure recovery plans for given CPS, threat circumstances and constraints. There one of the main challenges is expected to be the practical implementation of the language, which will possibly be supported through a tool.

Acknowledgments. The authors would like to thank the Engineering and Physical Sciences Research Council (EPSRC) for their support.

References

1. G. Antoniol, G. Canfora, G. Casazza, A. De Lucia, and E. Merlo. Recovering traceability links between code and documentation. *IEEE Transactions on Software Engineering*, 28(10):970–983, Oct. 2002.
2. P. Bareiss, D. Schutz, R. Priego, M. Marcos, and B. Vogel-Heuser. A model-based failure recovery approach for automated production systems combining SysML and industrial standards. pages 1–7. IEEE, Sept. 2016.
3. P. Chen, C. Scown, H. S. Matthews, J. H. Garrett, and C. Hendrickson. Managing Critical Infrastructure Interdependence through Economic Input-Output Methods. *Journal of Infrastructure Systems*, 15(3):200–210, Sept. 2009.
4. L. Compagna, P. El Khoury, A. Krausov, F. Massacci, and N. Zannone. How to integrate legal requirements into a requirements engineering methodology for the development of security and privacy patterns. *Artificial Intelligence and Law*, 17(1):1–30, Mar. 2009.
5. S. Crane, P. Larsen, S. Brunthaler, and M. Franz. Booby trapping software. pages 95–106. ACM Press, 2013.
6. D. Filipov, A. Roth, and E. Nakashima. Companies struggle to recover after massive cyberattack with ransom demands. *The Washington Post*, June 2017.
7. D. G. Firesmith. Engineering Safety and Security Related Requirements for Software Intensive Systems. pages 169–169. IEEE, May 2007.
8. T. Fox-Brewster. Medical Devices Hit By Ransomware For The First Time In US Hospitals. *Forbes*, May 2017.
9. D. Harel. On Visual Formalisms. *Communications of the ACM*, 31(5):514–530, May 1988.
10. J. Henley and O. Solon. 'Petya' ransomware attack strikes companies across Europe and US. *The Guardian*, June 2017.
11. R. Matulevicius, H. Mouratidis, N. Mayer, E. Dubois, and P. Heymans. Syntactic and semantic extensions to secure tropos to support security risk management. *Journal of Universal Computer Science*, 18(6):816–844, 2012.
12. N. R. Mead. Requirements engineering for survivable systems. Technical Report CMU/SEI-2003-TN-013, Carnegie Mellon University, Sept. 2003.
13. N. R. Mead and T. Stehney. Security quality requirements engineering (SQUARE) methodology. *ACM SIGSOFT Software Engineering Notes*, 30(4):1, July 2005.
14. D. Mellado, E. Fernndez-Medina, and M. Piattini. A common criteria based security requirements engineering process for the development of secure information systems. *Computer Standards & Interfaces*, 29(2):244–253, Feb. 2007.
15. MITRE. Adversarial Tactics, Techniques & Common Knowledge, https://attack.mitre.org/wiki/Main_page, last accessed 2018/05/30.
16. S. Mohurle and M. Patil. A brief study of Wannacry Threat: Ransomware Attack 2017. *International Journal of Advanced Research in Computer Science*, 8(5), June 2017.
17. D. Moody. The Physics of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering*, 35(6):756–779, Nov. 2009.

18. H. Mouratidis and P. Giorgini. Secure tropos: a security-oriented extension of the tropos methodology. *International Journal of Software Engineering and Knowledge Engineering*, 17(02):285–309, Apr. 2007.
19. OASIS. Structured Threat Information Expression, https://oasis-open.github.io/cti-documentation/stix/intro, last accessed 2018/05/30.
20. N. A. Office. Investigation: WannaCry cyber attack and the NHS. Department of Health Report HC414, National Audit Office, Oct. 2017.
21. S. H. Othman and G. Beydoun. A Disaster Management Metamodel (DMM) Validated. In B.-H. Kang and D. Richards, editors, *Knowledge Management and Acquisition for Smart Systems and Services: 11th International Workshop, PKAW 2010, Daegue, Korea, August 20 - September 3, 2010. Proceedings*, pages 111–125. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
22. J. Recker, M. Indulska, M. Rosemann, and P. Green. Business Process Modeling - A Comparative Analysis*. *Journal of the Association for Information Systems*, 10(4):333–363, Apr. 2009.
23. H. Security. Recommended Practice: Improving Industrial Control System Cybersecurity with Defense-in-Depth Strategies. Technical report, Department of Homeland Security (DHS) National Cybersecurity, Communications Integration Center (NCCIC) and Industrial Control Systems Cyber Emergency Response Team (ICS-CERT), Sept. 2016.
24. G. Sindre, D. G. Firesmith, and A. L. Opdahl. A Reuse-Based Approach to Determining Security Requirements. *Requirements Engineering*, 10:34–44, June 2004.
25. G. Sindre and A. L. Opdahl. Eliciting security requirements with misuse cases. *Requirements Engineering*, 10(1):34–44, Jan. 2005.
26. Soonwook Hwang and C. Kesselman. Grid workflow: a flexible failure handling framework for the grid. pages 126–137. IEEE Comput. Soc, 2003.
27. S. Winkler and J. von Pilgrim. A survey of traceability in requirements engineering and model-driven development. *Software & Systems Modeling*, 9(4):529–565, Sept. 2010.
28. E. Yu, L. Liu, and J. Mylopoulos. A social ontology for integrating security and software engineering. In *ntegrating Security and Software Engineering: Advances and Future Visions*, pages 70 –.
29. Z. Zhu, K. Sivakumar, and A. Parasuraman. A Mathematical Model of Service Failure and Recovery Strategies*. *Decision Sciences*, 35(3):493–525, Aug. 2004.