On synergies of cyber and physical security modelling in vulnerability assessment of railway systems ☆

Stefano Marrone [a,*], Ricardo J. Rodríguez [b], Roberto Nardone [c], Francesco Flammini [d], Valeria Vittorini [c]

[a] *Dip. di Matematica e Fisica, Seconda Università di Napoli, Caserta, Italy*
[b] *Research Institute of Applied Sciences in Cybersecurity, University of León, Spain*
[c] *DIETI, Università di Napoli "Federico II", Naples, Italy*
[d] *AnsaldoSTS, Naples, Italy*

A B S T R A C T

The multifaceted nature of cyber-physical systems needs holistic study methods to detect essential aspects and interrelations among physical and cyber components. Like the systems themselves, security threats feature both cyber and physical elements. Although to apply *divide et impera* approaches helps handling system complexity, to consider just one aspect at a time does not provide adequate risk awareness and hence does not allow to design the most appropriate countermeasures. To support this claim, in this paper we provide a joint application of two model-driven techniques for physical and cyber-security evaluation. We apply two UML profiles, namely SecAM (for cyber-security) and CIP_VAM (for physical security), in combination. In such a way, we demonstrate the synergy between both profiles and the need for their tighter integration in the context of a reference case study from the railway domain.

Keywords:
Cyber-physical systems
Vulnerability assessment
UML profile
Bayesian networks
Generalized stochastic Petri nets

1. Introduction

Cyber-physical systems emerged as a branch of the embedded systems research specifically focused on the interaction between the computational elements and the physical entities [1]. When research on cyber-physical systems overlaps with the emerging paradigms of smart-cities, Internet-of-Things and intelligent transportation, to name a few, then security issues become critical whereas distributed systems can be exposed to both physical and cyber-threats. It is a matter of fact that while researchers seem to be well-aware of the physical effects of cyber-threats, much of the research on information, on ‘‘logical’’, or on ‘‘cyber’’ security are not related to physical sensing. However, many threats to cyber-physical systems (especially when they are isolated from the Internet) are also originated from physical intrusions, e.g., intruders accessing control terminals in technical rooms. This kind of information should be fused with the one coming from logical intrusion detection to provide a superior situation awareness and early warnings; thus, merging physical with logical access control allows to recognise otherwise undetectable identity frauds.

Many safety–critical systems, as the ones used for railway control, are unreachable from the Internet but have technical equipment located in geographically distributed shelters and used for actuation, power, and telecommunications. This

* Corresponding author.
E-mail address: stefano.marrone@unina2.it (S. Marrone).

equipment is normally used by maintainers and other authorised personnel, but can be potentially targeted by unauthorised personnel through the same physical access points. Since trackside shelters and technical rooms are nowadays equipped with physical security and environmental monitoring devices, security threat analysis can be advantageously fed with both physical and logical elements [2].

Nowadays, holistic modelling of complex systems is still a challenging research issue, being largely accepted that the more promising and scalable approaches focus on modularity and composability (both in modelling and solving). Another promising research effort aims at using as much as possible *de facto* standards in systems modelling, as the Unified Modelling Language (UML) together with its extensions and Domain Specific Modelling Languages (DSMLs), in order to provide a modeller with easy-to-use, reusable tools. This enables to build cohesive system views while hiding the underlying complexity of the analysis process, often based on model-to-model (M2M) transformations and orchestration of different solvers for different formalisms biased on the evaluation objectives.

In this paper, we take advantage of two novel UML profiles, namely *Security Analysis and Modelling* (SecAM) [3] and *Critical Infrastructure Protection – Vulnerability Analysis and Modeling* (CIP_VAM) [4], to address the modelling of digital and physical security in combination. The approach moves from separate usage of the two profiles, through a loosely coupled one, pointing towards a fully and strictly integrated profile including the modelling potential of both SecAM and CIP_VAM. Besides, we also show how each profile benefits by the information contained in the other in the formal models generated and used for quantitative security evaluations. We combine the usage of SecAM and CIP_VAM to exploit synergies in modelling and analysis of cyber and physical security aspects: from UML models annotated by both profiles, a cyber and physical security analysis can be performed coping with the complexity of critical infrastructure protection. We finally evaluate our approach in an intrusion scenario in railway trackside/lineside shelters.

The rest of this paper is structured as follows. Section 2 reviews the related work, introduces SecAM, CIP_VAM, and background needed to follow the rest of the paper. Section 3 describes the reference case study of the railway shelter used to motivate our research. Section 4 introduces the vulnerability modelling, considering separately physical and cyber security. Then, Section 5 considers them jointly, and proposes some modelling enhancements. Section 6 demonstrates the effectiveness of our approach by means of sensitivity analyses. Finally, discussion and conclusions are drawn in Section 7.

2. Related work and background

2.1. Related work

Model-based evaluation of computer and network security has a long story, dating back to the first techniques to model and evaluate system dependability [5]. Dependability and security model-based evaluation approaches encompass *combinatorial methods* (e.g., based on Reliability Block Diagrams, Fault Trees, or Attack Trees), *State-Based Stochastic Methods* (e.g., through Markov Reward Models or Stochastic Petri Nets), *Model Checking* (e.g., automatic attack graphs generation [6]), or a combined used of several methods and formalisms [7]. However, security of critical infrastructures like those for mass-transit transportation is a multi-facet problem that requires an integrated approach taking into account digital (i.e., cyber) security as well as physical security, which is strictly related to system protection against intentional threats of physical nature. In physical vulnerability assessment, a *quantitative* notion of vulnerability is used and commonly defined as the likelihood that an attempted attack is successful [8]. In this direction, practical applications for vulnerability analysis use statistical approaches and mathematical modelling [9,10]. Nevertheless, model-based approaches for cyber-security and physical security are separately considered and applied.

A recent trend in critical system modelling for security and dependability analysis envisions top-down model-driven approaches that automatically derive quantitative models. These approaches rely on DSMLs or UML profiles for specification and modelling of a kind of systems. Model-driven processes enable automated modelling and analysis of different solutions during the overall system development life-cycle (for instance, security solutions or design choices) and they maybe easily integrated in industrial settings. So far, few DSMLs or profiles exist specifically tailored for modelling security and vulnerability aspects of critical infrastructures. CORAS [11] assists in modelling and analysing the risk of changing systems in terms of their Quality of Service and fault tolerance characteristics. MARTE [12] is an OMG standard profile for modelling and analysing non-functional properties of real-time embedded systems. Similarly, Dependability Analysis and Modelling (DAM) [13] is a non-standard specialisation of MARTE that supports dependability analysis. Regarding UML profiles addressing security, UMLsec [14] allows to specify security information during the development of security-critical systems and provides tool-support for formal security verification. An UML extension is also proposed in [15] for model-based security assessment. UMLintr [16] is a further profile for specifying intrusion scenarios. Other UML profiles focus on security in grid computing [17] or distributed systems [18]. In this sense, CIP_VAM [4,19] is a recent UML profile that addresses physical protection of critical infrastructures and provides tool support for automatic generation of vulnerability models based on Bayesian Networks (BNs). However, it does not consider cyber-security issues. Another recent UML profile, SecAM [20,3], overcomes this issue since it allows specifying cyber-security aspects while enabling their assessment.

At the best of our knowledge, there are a lot of scientific works comparing UML profiles in different contexts but there are only few of them exploring the synergies of a joint use: in [21], MARTE, SysML, and UMLSec are used to model non-functional properties of telecommunication systems; in [22], MARTE and MARTE-DAM are mixed to allow evaluation of performance

and dependability. With respect to these works, the proposed approach also pursues the objective of improving existing transformational approaches by the joint use of different UML profiles rather than to use existing transformations separately.

2.2. The CIP_VAM profile and Bayesian networks

CIP_VAM [4] is an UML profile for vulnerability analysis and modelling in the field of critical infrastructure protection; conceived and developed within the European project METRIP[1] under the "Prevention, Preparedness and Consequence Management of Terrorism and other Security-related Risks Programme" to support design and evaluation of physical protection systems. CIP_VAM extends UML with concepts for modelling critical assets (*Infrastructure* package), attack scenarios (*Attack* package), and protection devices (*Protection* package). A more detailed description of CIP_VAM subpackages can be found in [4].

By applying proper model transformations on a CIP_VAM-annotated UML model, it is possible to obtain a BN model. BNs [23] provide a graphical representation of a joint probability distribution over a set of random variables with a possible mutual causal relationship. The network is a directed acyclic graph whose nodes represent random variables and arcs represent a probabilistic dependence between two random variables. A conditional probability distribution is defined for each node in the network: for discrete random variables, it is often represented by a Conditional Probability Table (CPT). Founded on the Bayes' theorem, BNs allow for inferring the posterior conditional probability distribution of a hypothesis (outcome variable) based on observed evidence and a prior belief in the probability of different hypotheses.

2.3. The SecAM profile and generalized stochastic Petri nets

SecAM [20,3] is an UML profile designed for the security analysis and modelling of software systems. It allows attack/resilience, cryptography, security mechanisms, and access control issues to be expressed within UML models. SecAM is built in the top of MARTE [12] and DAM [13] profile, which makes it a powerful framework where performance, dependability, and security properties can be expressed. A more detailed description of SecAM subpackages can be found in [3].

By applying proper model transformations on a SecAM-annotated UML model, it is possible to obtain a Generalized Stochastic Petri nets (GSPN) model. A GSPN [24] is a graphical and mathematical formalism that allows to specify both temporal and logic evolution within the same model. A GSPN is a bipartite graph in which the nodes (*places* and *transitions*) are connected by directed *arcs*. *Tokens* (drawn as black dots within places) are used to specify and to evolve the state of a Petri net (PN) by means of the *firing rule*: firing of the transitions determines the change of the number of tokens in places. A GSPN distinguishes two kind of transitions: *immediate*, which fire at zero time (i.e., its firing does not consume any time) and are represented by thin black bars; and *timed*, which can follow different firing distributions and are represented by white boxes. In this paper, we consider that delays of timed transitions are exponentially distributed random variables.

3. Case study: Lineside shelter scenario

This section describes the case-study of the lineside shelter protection system. Shelters are small buildings located in railway track sides. They usually contain electronic equipment performing control functions on interlocking devices (e.g., light signals, railway switches, etc.). As the complexity of railway systems increases, such functions are more and more demanded to computer-based systems, which are often connected to open networks. However, some commands to the railway devices are disallowed in a remote control scenario. In these situations, shelter computers send diagnostic data to a Central Command Center, while maintenance actions are performed only via a local terminal. Hence, since shelters feature critical security issues, appropriate protections must be employed.

Fig. 1 depicts the layout of a common lineside shelter, used as a reference example in this paper. The shelter is a small building, accessed by a door controlled by a Door Access control device (e.g., a card reader or keypad). The first room accessible by the door features intrusion presence detection sensors (e.g., passive infrared), a Fog Generator (i.e., a device to temporarily blind intruders by generating an artificial mist). A rack is also located at this room, containing two servers: a Signalling Server (or Traffic Server) that controls the railway devices; and a Cyber-Physical Protection (CPP) Server that is responsible of monitoring the shelter security. Fig. 2 depicts an UML Deployment diagram describing the cyber-physical architecture of the shelter.

In nominal operating conditions, the shelter is accessed by maintenance personnel with their access keys. Maintainers log in the system via the terminal to perform the required interventions, e.g., to fix some problems. Several threat scenarios can be considered where physical or cyber security are addressed separately: for example, the case of an unauthorised access to the room to steal some stuff, or a remote intrusion to sabotage the Traffic Server). Proper countermeasures can be activated to cope with these threats (e.g., fog generator and user disconnection, respectively). Nevertheless, scenarios that combine physical and cyber attacks are also possible. Since we focus on combined attacks, let us consider the following scenario:

Insider Threat. A CyberPhysical Trudy, defined as an "intruder able to perform an attack requiring both physical and cyber actions", wants to sabotage the Traffic Server. We assume here that the attacker is part of the personnel who is allowed to

[1] http://metrip.unicampus.it/.

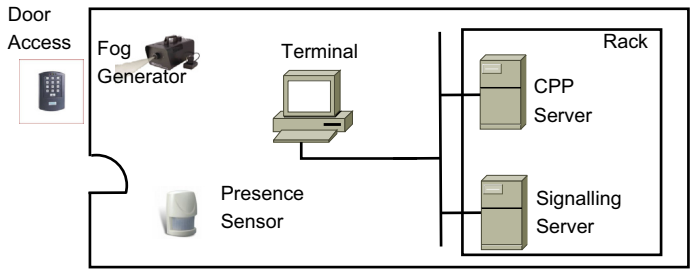

Fig. 1. The lineside shelter reference plant.

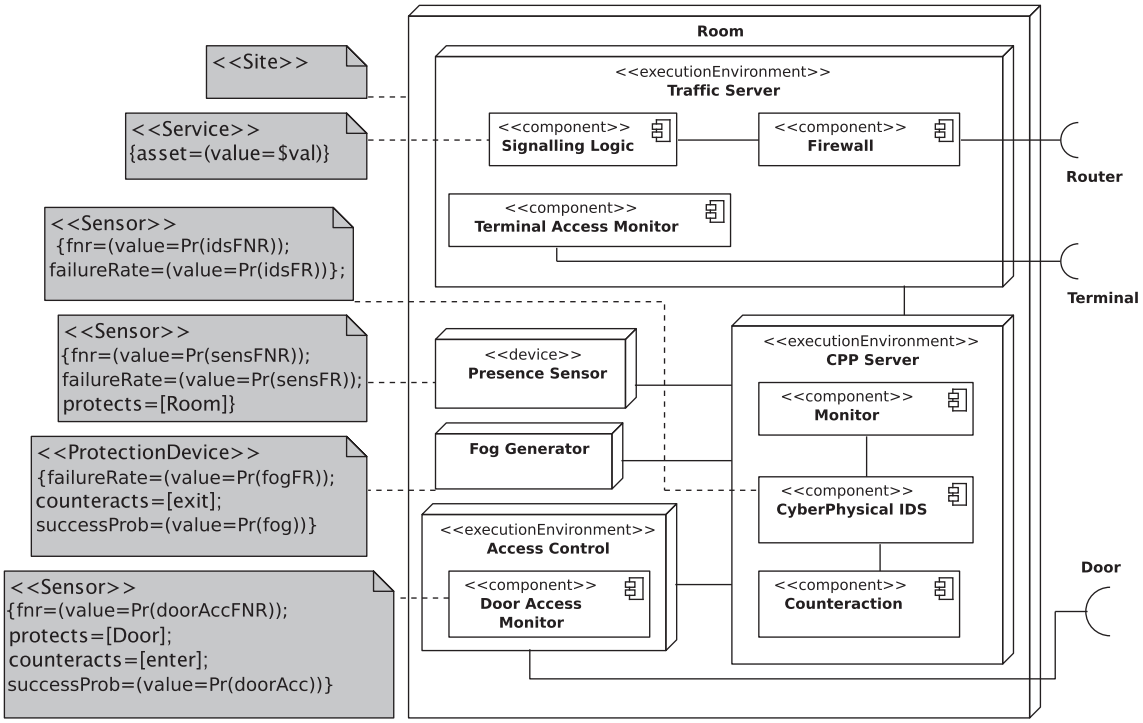

Fig. 2. UML deployment diagram of the shelter.

physically access the shelter room but not allowed to login the computer system with high privilege credentials (e.g., administrator). By correlating physical and logical credentials, the CPP Server recognises the intrusion and activates cyber-countermeasures (i.e., user session disconnection and notification to the control room).

A misuse/anomaly network Intrusion Detection Systems (IDSs), used by most companies, detects logical intrusions into the server. However, traditional IDSs do not correlate physical access control and logical access control information and therefore they are unable to detect threats as the one described above.

4. Vulnerability modelling of the Insider Threat scenario

Fig. 3 depicts the UML Sequence Diagram (UML-SD) illustrating the behaviour of the Insider Threat scenario. This UML diagram is annotated with both CIP_VAM and SecAM UML profiles in order to show the advantages of a joint modelling; this notwithstanding, the first steps are constituted by showing how these two profiles work separately.

4.1. Physical protection in CIP_VAM

This subsection shows how the CIP_VAM approach is used in the Insider Threat scenario to evaluate the vulnerability of the system against the attack. At this aim the CIP_VAM profile is used to annotate the infrastructure (the shelter room), the

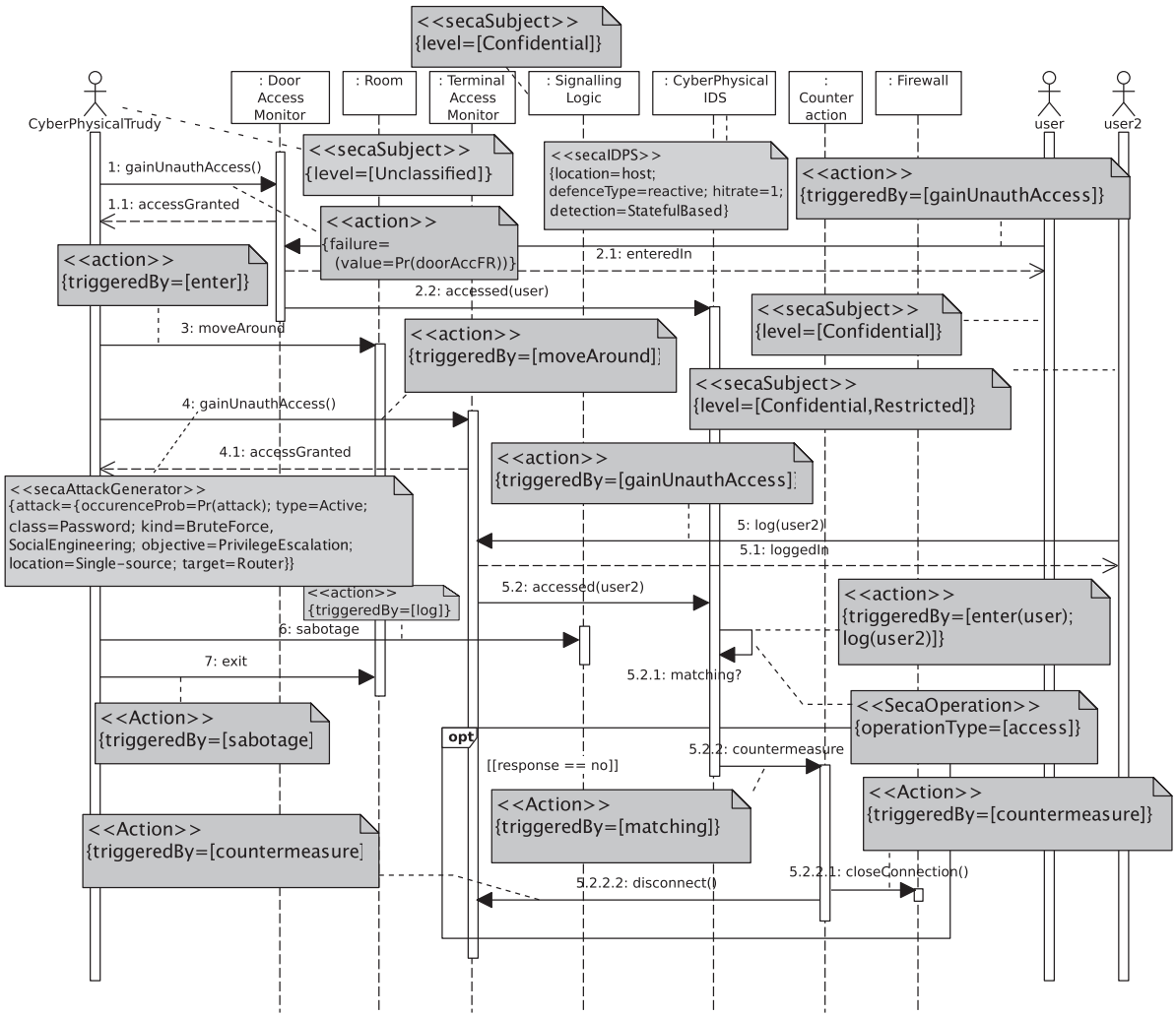

Fig. 3. CIP_VAM and SecAM annotated UML-SD of the Insider Threat scenario.

asset to defend (the signalling service provided by the shelter), the protection system (the access control mechanism, the presence sensor, and the fog generator), as well as the sequence of steps performed by the intruder.

In [4], attacks are represented by Use Cases and an Activity is associated to each Use Case when a behaviour is detailed. This paper proposes to model the steps of an attack by means of the messages of a Sequence Diagram as well as activities of an Activity Diagram. This update mainly consists in modifying the «`Action`» stereotype from the *Attack* package so that it also extends the UML metaclass *Message*. From the annotated UML diagrams, a transformation generates a BN model suitable for evaluating the success probability of an attack.

At this regard, two diagrams are considered: the Deployment Diagram already presented in Fig. 2 (CIP_VAM annotations are grey-highlighted); and the Sequence Diagram modelling the Insider Threat scenario in Fig. 3 where the following CIP_VAM stereotypes are used: «`Site`» is used to annotate the Room representing the shelter; «`Service`» is used to annotate the Signalling Logic (it specifies the main asset of the system by setting a proper value for the *asset* tagged value); «`ProtectionDevice`» is used for protection mechanisms that are not sensors but security actuators (e.g., the fog generator); «`Sensor`» is used for the *DoorAccess*, the *CyberPhysical IDS*, and the *Presence Sensor*. Each of these sensors *protects* an infrastructural item, i.e., the *Room* (*Presence Sensor*) and the *Door* (*DoorAccess*).

Each step that is part of the attack scenario (also from a protection point of view) is annotated with the «`Action`» stereotype and enriched with useful information by means of proper tagged values.

An M2M transformation is used to derive the BN model, according to the rules sketched below.

Rule "Action" nodes. A true/false BN node is generated for each «`Action`» *A*: the true value means that the corresponding attack step has been successfully accomplished, otherwise the attack step failed/has been neutralised. An arc is generated to *A* from each BN node corresponding to an «`Action`» *TA* in the *TriggeredBy* list of *A*, and from each BN node corresponding to a

«Protection» *PA* that has *A* in its *Counteracts* list.[2] Hence, the CPT of *A* takes into account the *SuccessProb* value of $PA, Pr(succ_{PA})$, and *OccurrenceProb* of $TA, Pr(occ_{TA})$. An example is reported in Table 1. The case of more than a *TA* and/or *PA* can be dealt with supposing that at least one true *TA* is needed to activate *A* and that at least one *PA* is needed to be true to inhibit *A*; hence, the CPT of *A* can implement a "noisy-OR" mechanism.

Rule "Sensor" nodes. A pair (S_S, S_E) of true/false BN nodes is generated for each «Sensor» *S*. S_S represents the availability/unavailability of *S*; the true value of S_S (with probability *FailureRate*) means that the device is working; otherwise, it is down. An arc is generated from S_S to S_E which in turn models the effect of the sensor *S*: the true value of S_E means the sensor successfully senses/measures the event/quantity for which it is in charge of. An arc is also added to S_E from each BN node corresponding to any «Action» that may be performed against infrastructural items («Site», «Object» or «Service») in its *Protects* list. Finally, the CPT of S_E takes into account the trustworthiness of *S* in terms of its false positive rate (*Fpr*) and false negative rate (*Fnr*) [19].

Rule "Protection Devices" nodes. A pair (D_D, D_E) of BN nodes is generated for each «Action» *D* different from «Action». D_D and D_E have the same meaning respectively of S_S and S_E; hence, this rule works as the previous one. An example is reported in Table 2.

Fig. 4 depicts the BN model related to the Insider Threat scenario.

4.2. Cyber protection in SecAM

Herein, the SecAM approach is used in the Insider Threat scenario to perform a vulnerability assessment. To this aim, SecAM is used to explicitly annotate the step sequence performed by an intruder.

The Insider Threat scenario described by the UML-SD in Fig. 3 has been considered as starting point.[3] For the sake of readability, the UML models contains only the UML annotated elements and tagged values needed to understand the generation process of the GSPN model (annotations starting with *seca* prefix). This scenario has been enriched with *Resilience* and *SecurityMechanisms* SecAM packages [20,3].

The SecAM::Resilience package contains attack, vulnerability, and intrusion concepts, as well as their causal relationships. Thus, these stereotypes enable to characterise attacks, vulnerability steps, and intrusions within UML models.

Similarly, *SecurityMechanisms* package contains a set of stereotypes to specify different security devices. These devices share some characteristics, such as the deployment location, type, defence type, hit rate, and operational rate; while others are product specific.

A CyberPhysical Trudy, stereotyped as an unclassified subject, has authorisation to access to the room. This triggers the enter action of the user, which is notified by the *Door Access Monitor* to the *CyberPhysical IDS* (CP_IDS for short). The *CyberPhysical IDS* boosts traditional IDSs by adding the capability to correlate physical and logical events enabling the modelling approach proposed in this paper. Then, the subject gains authorisation access to the terminal access by performing an attack with a probability of *Pr*(*attack*) (sensitive analysis parameter). Let us assume that the attacker is able to obtain a valid password by means of brute-force attack or by social engineering techniques. A valid password allows her to get privilege escalation, as she is able to obtain a granted access through the *Terminal Access Monitor*. Once the attacker has logged in, action log of user2 is triggered, whereas a notification is sent by the *Terminal Access Monitor* to the CP_IDS. The CP_IDS is stereotyped as located at the host, having a hit rate of 1 (an ideal CP_IDS), reactive defence type, and stateful-based detection.

The intruder proceeds now to sabotage the *Signalling Logic*, a security object stereotyped with a Confidential level. Meanwhile, the CP_IDS checks whether the logged user matches with the last user who accessed through the door into the room. When there is not a match (which happens with the same probability than attack in fact, annotated with secaStep stereotype), the CP_IDS alerts the *Firewall* that closes the active connection. Lastly, the CP_IDS propagates the user disconnection to the rest of system elements.

The UML-SD described in Fig. 3 can be transformed to a formal model suitable for evaluation. In this paper, we used well-established approaches [25,26] to obtain a Petri Net (namely, a GSPN [24]) from the previous UML-SD. Fig. 5 depicts the GSPN obtained as a transformation from the aforementioned UML-SD, which is suitable for quantitative and qualitative evaluation.

UML-SD lifelines are represented by resource places p_{16} to p_{21} (places initially marked, we have as well labelled them with a comprehensive text), while CyberPhysicalTrudy access is represented by firing of transition t_1. An access request will eventually reach place p_7 that represents the attempt of login into the terminal, annotated with a occurrence probability of $\pi_7 = Pr(attack)$ (see the optional UML-SD in Fig. 3). Thus, a malicious access request is finally discarded as indicated by transition t_8 (with probability $\pi_8 = 1 - Pr(attack)$).

The acquire (release) of a resource has been transformed into a transition with an input (output) arc. For instance, transition t_1 represents the acquire of the *Door Access Monitor*, while T_2 represents the release of such resource after some action (represented by the own T_2). Activities and self-messages have been transformed into an exponential transition in the Petri net model. The duration of these transitions may be indicated in the UML model using MARTE [12]. For the sake of readability, we have not depicted these annotations.

[2] *TriggeredBy* is a tagged value of «Action» while *Counteracts* is a tagged value of «Protection»; the related lists are the collections UML uses to link elements connected by an association.

[3] Regarding this diagram, only SecAM-related stereotypes are used in this subsection.

Table 1
CPT related to the Rule "Action".

TA	*PA*	*A = true*	*A = false*
True	True	$(1 - Pr(succ_{PA})) \cdot Pr(occ_{TA})$	$1 - (1 - Pr(succ_{PA})) \cdot Pr(occ_{TA})$
True	False	$Pr(occ_{TA})$	$1 - Pr(occ_{TA})$
False	–	0	1

Table 2
CPT related to the Rule "Protection Device".

D	D_S	D_E = *true*	D_E = *false*
True	True	1	0
True	False	0	1
False	–	0	1

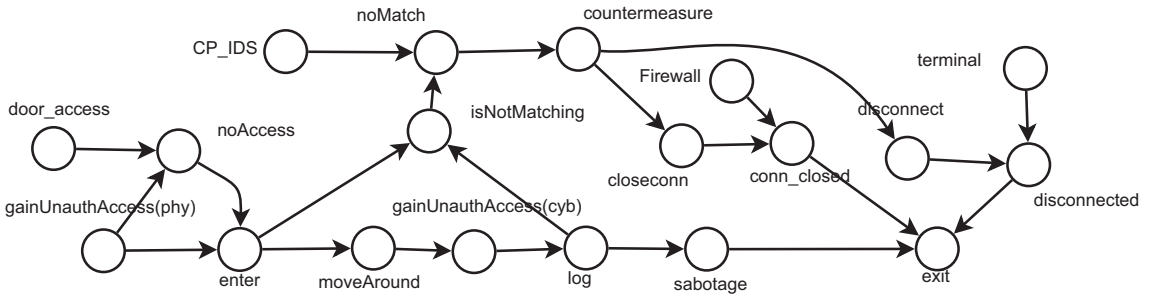

Fig. 4. BN model of the Insider Threat (CIP_VAM approach).

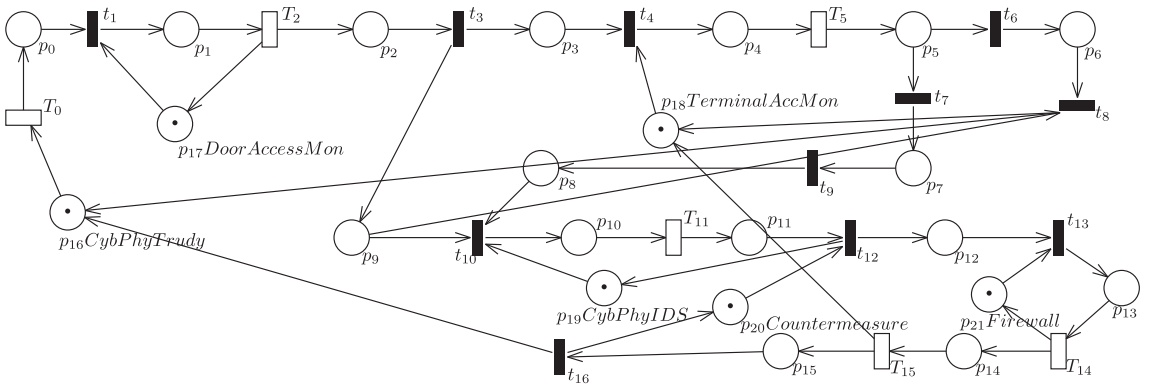

Fig. 5. Petri Net model of the cyber attack scenario.

5. Joint use of SecAM and CIP_VAM

In this section we show the advantages of the jointly usage of the CIP_VAM and SecAM approaches. The M2M transformation generating the BN (resp. GSPN) from a UML model annotated with CIP_VAM (resp. SecAM) stereotypes can be defined by taking into account information coming from SecAM (resp. CIP_VAM) annotations, as proposed in the next subsections.

5.1. Improving the CIP_VAM approach

Here, a "SecAM-aware" extension of the CIP_VAM approach is proposed in order to improve its effectiveness. One of the features that SecAM deals with in a more clear and straightforward way, is the management of the access control mechanisms.

The weakness of the BN model shown in Fig. 4 is in the node *isNotMatching*. In fact, its aim is to model the procedure by which the CP_IDS detects a mismatch in the identities used for physical and cyber accesses. Without extensions, only a pure

probabilistic mechanism can be embedded in the BN by defining a proper CPT for this node. This approach does not take into account the real logic behind an access control ruleset. The centre of the proposed extension is the explanation of the two roles (used for the physical and the cyber accesses) by means of two UML actors (*user* and *user2*) that are annotated with the SecAM «`SecaSubject`» stereotype. Then, the second step is to annotate the «`Action`» that generates the node `isMatching` (i.e., the *isMatching* UML Message) by means of the SecAM «`SecaOperation`» stereotype. The use of SecAM annotations generates a BN with more nodes and arcs connections so accounting for more variables and casual relationships. A new rule is added to the ones described in Section 4.1: for each «`Action`» *A* also stereotyped with «`SecaOperation`» a subnet is generated consisting of a node corresponding to *A* and three additional nodes for each triggering «`Action`» originated by UML actors tagged as «`SecaSubject`» (see as example the BN excerpt depicted in Fig. 6). Of course, additional arcs are also generated between nodes, but we omit a deeper description of the rule for sake of brevity.

5.2. Improving the SecAM approach

Similarly, in this section we consider a "CIP_VAM-aware" extension of the model transformation shown in Section 4.2 in order to improve its effectiveness. The major advantage of CIP_VAM is its ability to annotate the concrete events that trigger interaction among system components. By bringing these annotations to a UML model, a PN model can be conformed where system state transitions depend on the occurrence of specific events.

Fig. 7 depicts an excerpt of the Petri net model obtained from the transformation of the Insider Threat UML-SD (see Fig. 3), accounting for CIP_VAM stereotypes. For the sake of space, we only show an excerpt of the generated model. Three black dots in the figure represent the omitted part of the model.

The use of CIP_VAM annotations produces a Petri net with more places and arc connections. Each action expressed in the *TriggeredBy* CIP_VAM tagged-value creates a new place. This place is connected by an input arc to the transition that represents the execution of the trigger action, and by a test arc to the transitions that represent the execution of the action triggered. For instance, the place $p'_{gainUnathAccess}$ represents that the execution of the step *gainUnathAccess()* in the UML-SD of Fig. 3. Therefore, an input arc connects transition t'_1 with this place, given that such a transition represents the execution of the action. Finally, a test arc connects the place to transition t_{10}, since this transition represents the step triggered by the previous *gainUnathAccess()* method in the UML-SD. Besides, the `failure` CIP_VAM tagged-value adds also a decision in the model, represented by place p' and transitions t'_1, t'_2, having a probability of occurrence given by the value of `failure` (i.e., $\pi_{t'_1} = Pr(doorAccFR)$ and $\pi_{t'_2} = (1 - Pr(doorAccFR))$).

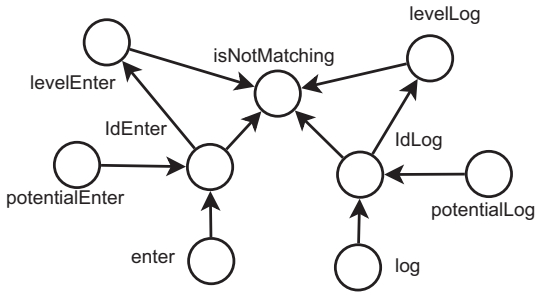

Fig. 6. Excerpt of the BN model of the Insider Threat (improved CIP_VAM approach).

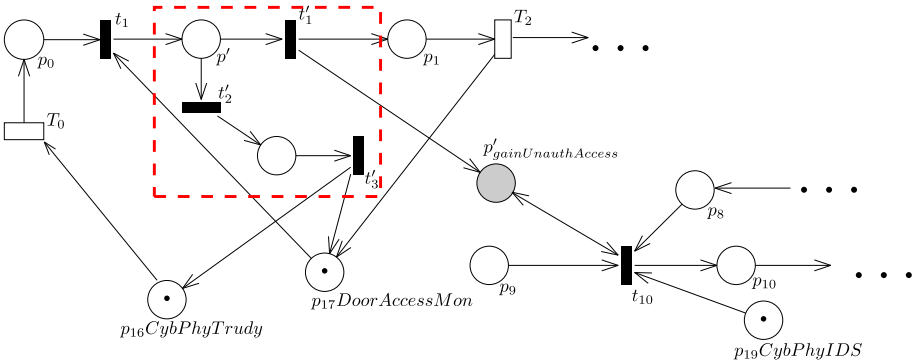

Fig. 7. Excerpt of Petri net model of the Insider Threat scenario.

Table 3
Model quantitative parameters.

Name	Description	Value
$Pr(idsFNR)$	False Negative Rate of the cyber-physical IDS component of the CPP Server	$15 \cdot 10^{-1}$
$Pr(idsFR)$	Failure Rate of the cyber-physical IDS component of the CPP Server	10^{-7}
$Pr(doorAccFR)$	Failure Rate of the door access device	10^{-5}
$Pr(doorAccFNR)$	False Negative Rate of the door access device	10^{-2}
$Pr(doorAcc)$	Success probability of the door access device	0.9
$Pr(firewallFR)$	Failure Rate of the firewall	10^{-4}
$Pr(firewall)$	Success probability of the firewall	0.8
$Pr(terminalFR)$	Failure Rate of the terminal	10^{-5}
$Pr(terminal)$	Success probability of the terminal	0.9

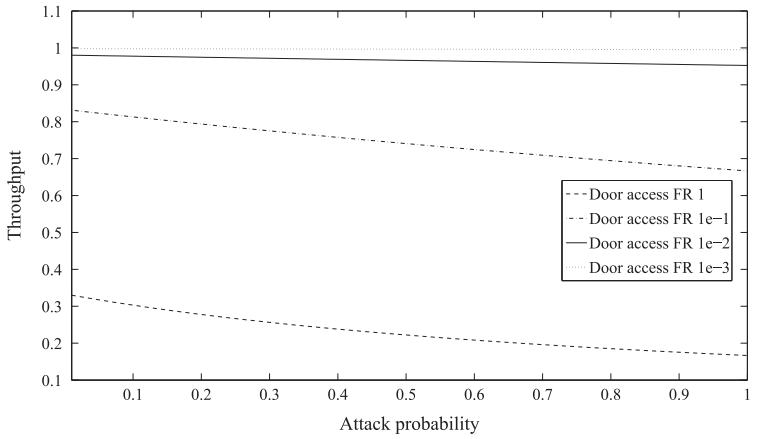

Fig. 8. Analysis results of models m_C and m'_C.

The obtained PN model when considering both CIP_VAM and SecAM profiles is a suitable model for qualitative and quantitative evaluation. For instance, a probabilistic model checker tool can be used to verify that the sequence of triggered events is correct. In the next section, we quantitatively evaluate the model by comparing the effect of failure rate in attack success.

6. Analysis of the Insider Threat scenario

This section demonstrates how the evaluation of the vulnerability related to the Insider Threat scenario can benefit from the joint modelling of cyber and physical security concerns. To pursue this objective, we analyse the models m_P, m'_P, generated by applying the transformations from CIP_VAM to BN described in Sections 4.1 and 5.1, as well as the models m_C, and m'_C generated by applying the transformations from SecAM to GSPN described in Sections 4.2 and 5.2. Models m_P, m'_P, are analysed against the variation of a "cyber-related" parameter while m_C, m'_C, against the variation of a "physical-related" parameter. The values used in this study are reported in Table 3. Some of them refer to frequencies (i.e., they are expressed in hrs^{-1}). As they all refer to the same time period (one hour), they are considered non-dimensional and used to define probabilities.

The models m_P and m'_P are analysed against variations of the access control policy. In particular, three cases are considered: (i) no logical access control; (ii) confidentiality levels are used to determine the logical access (i.e., the CyberPhysical Trudy succeeds in her intent only if she uses two accounts with confidentiality at least equal to the one requested to access the resource); and (iii) identities are used to determine the logical access (i.e., CyberPhysical Trudy succeeds in her intent only if she uses the same identities both for the physical and for the cyber access). The first case is modelled by m_P. The second and third cases refer to the m'_P model: the difference between them consists in having different CPTs for the A nodes as generated by the additional rule of the enhanced model transformation. The results come from the following formula: $Pr(exit = true|gainUnauthAccess(phy) = true)$ (see Fig. 4).[4]

The obtained results for the probability of a success in case of attack are: (i) $1.82 \cdot 10^{-2}$; (ii) $6.36 \cdot 10^{-2}$; and (iii) $1.90 \cdot 10^{-1}$. The model m'_P is more accurate than m_P since it takes into account the adopted access control policy.

[4] The formula follows the more generic one $Pr(success = true|attack = true)$). In this specific case, the names of the BN nodes are computed from the first and the last «Action» stereotyped messages in the attack scenario model in Fig. 3.

Specifically, m_P underestimates the probability of having a successful attack, given that the attacker performs the physical and cyber accesses with the same identity. The difference between the two vulnerability evaluations is one order of magnitude.

The models m_C and m'_C are analysed against variations of the failure rate of the door access device: its value is chosen in an neighbourhood of its default value. Specifically, m'_C is analysed considering these values: 10^{-1}, 10^{-2} and 10^{-3} in order to show how variation influences the overall vulnerability analysis. These results are also compared with the results of the analysis of m_C (where it is assumed that the attacker is always successful in compromising the door access control) in order to show the effectiveness of the approach.

Fig. 8 depicts the analysis results. We measured the system's performance by computing the throughput of transition t_1 under different door access failure rates. For the sake of simplicity, we assume that all timed transitions take 1 ms to fire. In all cases, performance decreases as the attack probability increases. Results of model m_C represent the case where failure rate is always occurring and thus the system performance is the lowest. As expected, the performance of the system increases when door access failure rate decreases. Let us remark that the use of CIP_VAM enables to consider physical failures in the model, clearly improving it since it is closer to real world situations where security of physical elements can be compromised.

7. Discussion and conclusions

While holistic approaches in system security are theoretically able to take a picture of all the relevant system aspects and security threats, it does not seem to exist any single modelling language able to manage the complex structure and behaviours of such systems as a whole. Therefore, the only viable solution is to extend existing modelling languages and possibly to design novel hybrid formalisms.

In this paper, we have addressed the limits of model-driven approaches when modelling cyber-only or physical-only security aspects in cyber-physical systems, and more specifically in railway applications. At this aim, two modelling approaches, oriented to different security-related aspects, have been chosen and jointly used. They both are based on UML profiling and quantitative model generation: the CIP_VAM approach has been enriched by AccessControl information provided considered by the SecAM approach in generating a more accurate BN model for vulnerability evaluation; the SecAM approach has been enriched by the action-triggered annotation and physical failure rates provided by CIP_VAM in generating a richer GSPN model.

In this context, we have highlighted some issues in separated cyber and physical security modelling, and proposed a possible solution. We are aware that the way is still long and further steps are needed also in order to propose this solution in other applicative domains. Of course, the loosely coupled approach here presented must be improved: model transformations must be further detailed, multi-formalism techniques may be investigated to provide inter-operation between derived models, a tight integration of the two UML profiles can allow to define a more comprehensive and usable modelling language. Nevertheless, we believe that the work presented in this paper is a concrete and first step towards cyber-physical vulnerability modelling and analysis.

Acknowledgements

This work was partially supported by the Spanish National Institute of Cybersecurity (INCIBE) according to rule 19 of the Digital Confidence Plan (Digital Agency of Spain) and the University of León under contract X43.

References

[1] Wolf W. Cyber-physical systems. Computer 2009;42(3):88–9.
[2] Bocchetti G, Flammini F, Pragliola C, Pappalardo A. Dependable integrated surveillance systems for the physical security of metro railways. In: Proc of the 3rd ACM/IEEE int conf on distributed smart cameras (ICDSC); 2009. p. 1–7.
[3] Rodríguez RJ, Merseguer J, Bernardi S. Modelling security of critical infrastructures: a survivability assessment. Computer J. http://dx.doi.org/10.1093/comjnl/bxu096.
[4] Marrone S, Nardone R, Tedesco A, D'Amore P, Vittorini V, Setola R, et al. Vulnerability modeling and analysis for critical infrastructure protection applications. Int J Crit Infrastruct Prot 2013;6(3–4):217–27.
[5] Nicol DM, Sanders WH, Trivedi KS. Model-based evaluation: from dependability to security. IEEE Trans Dep Secure Comput 2004;1(1):48–65.
[6] Jha S, Sheyner O, Wing J. Two formal analyses of attack graphs. In: Proc of the 15th IEEE WS on comp security foundations. CSFW '02. Washington, DC, USA: IEEE Computer Society; 2002. p. 49–63.
[7] Deavours D, Clark G, Courtney T, Daly D, Derisavi S, Doyle JM, et al. The Möbius framework and its implementation. IEEE Trans Soft Eng 2002;28(10):956–69.
[8] Lewis T, Darken R, Mackin T, Dudenhoeffer D. Model-based risk analysis for critical infrastructures. Critical infrastructure security. WIT Press; 2011.
[9] Ezell B. Infrastructure vulnerability assessment model (I-VAM). Risk Anal 2007;27(3):571–83.
[10] Brown GG, Carlyle WM, Salmerón J, Wood K. Analyzing the vulnerability of critical infrastructure to attack and planning defenses. In: Tutorials in operations research. INFORMS, INFORMS; 2005. p. 102–23.
[11] Lund MS, Solhaug B, Stølen K. Risk analysis of changing and evolving systems using CORAS. In: Foundations of security analysis and design VI – FOSAD. LNCS, vol. 6858. Springer; 2011. p. 231–74.
[12] OMG. UML profile for MARTE: modeling and analysis of real-time embedded systems, version 1.1, formal/11-06-02; June 2011.
[13] Bernardi S, Merseguer J, Petriu D. A dependability profile within MARTE. J Softw Syst Model 2011;10(3):313–36.
[14] Jürjens J. Secure systems development with UML. Springer; 2005.

[15] Houmb S, Hansen K. Towards a UML profile for security assessment. In: Work on critical systems development with UML; 2003. p. 815–29.
[16] Hussein M, Zulkernine M. UMLintr: a UML profile for specifying intrusions. In: Proc of the 13th Int symp on eng of computer based systems, ECBS, IEEE-CS; 2006. p. 279–88.
[17] Rosado DG, Fernandez-Medina E, Lopez J, Piattini M. Developing a secure mobile grid system through a UML extension. J Univ Comput Sci 2010;16(17):2333–52.
[18] Lodderstedt T, Basin D, Doser J. SecureUML: a UML-based modeling language for model-driven security. In: Proc of the 5th int conf on the unified modeling language. Springer-Verlag; 2002. p. 426–41.
[19] Vittorini V, Marrone S, Mazzocca N, Nardone R, Drago A. A model-driven process for physical protection system design and vulnerability evaluation. Railway infrastructure security, vol. 27. Springer; 2015. p. 143–69. http://dx.doi.org/10.1007/978-3-319-04426-2_8.
[20] Rodríguez RJ, Merseguer J, Bernardi S. Modelling and analysing resilience as a security issue within UML. In: Proc of the 2nd int workshop on soft eng for resilient systems. SERENE '10. ACM; 2010. p. 42–51.
[21] Saadatmand M, Cicchetti A, Sjödin M. Uml-based modeling of non-functional requirements in telecommunication systems. In: The 6th int conf on software engineering advances (ICSEA); 2011.
[22] Berardinelli L, Bernardi S, Cortellessa V, Merseguer J. UML profiles for non-functional properties at work: analyzing reliability, availability and performance. In: Int work NfP in DSML; 2009.
[23] Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufman Pubs. Inc.; 1988.
[24] Marsan MA, Balbo G, Conte G, Donatelli S, Franceschinis G. Modelling with generalized stochastic Petri nets. John Wiley and Sons; 1995.
[25] López-Grao JP, Merseguer J, Campos J. From UML activity diagrams to stochastic Petri nets: application to software performance engineering. In: Int ws on software and performance (WOSP). ACM; 2004. p. 25–36.
[26] Distefano S, Scarpa M, Puliafito A. From UML to Petri nets: the PCM-based methodology. IEEE Trans Soft Eng 2011;37(1):65–79.

Stefano Marrone is an assistant professor in Computer Engineering at Seconda Università di Napoli. His interests include the definition of model driven processes for the design and the analysis of transportation control systems, complex communication networks and critical infrastructures. He is currently involved in research projects with both academic and industrial partners.

Ricardo J. Rodríguez received M.S. and Ph.D. degrees in computer science from the University of Zaragoza, Spain, in 2010 and 2013, respectively. He is currently a senior/postdoctoral researcher at the Research Institute of Applied Sciences in Cybersecurity, University of León, Spain. His research interests include performance analysis and optimisation of large and distributed systems, computer software security, and dependability.

Roberto Nardone is a post-doctoral fellow at Universitá di Napoli Federico II. His research interests are in the quantitative evaluation of non-functional properties, with a focus on dependability and performability assessment and threat propagation analysis, by means of model-based and model-driven techniques.

Francesco Flammini got his Ph.D. in Computer Engineering from the Universitá di Napoli Federico II. Since 2003, he has worked in Ansaldo STS on the reliability, safety and security of railway systems. He has taught Computer Science, Software Engineering and Risk Assessment. He has authored/edited several publications on dependable systems. He is a Senior Member of the IEEE.

Valeria Vittorini is an associate professor at the Universitá di Napoli Federico II. She teaches Formal Modeling and Computer Programming. Her research interests include dependability and performance evaluation of computer systems, validation and verification of critical systems, critical infrastructures protection and model-driven approaches applied to the automatic generation of formal models.