A Pyramidal-based Model to Compute the Impact of Cyber Security Events

Gustavo Gonzalez-Granadillo
Atos Research and Innovation
C/ Pere IV, 291-307, 08020 Barcelona, Spain
gustavo.gonzalez@atos.net

Jose Rubio-Hernan, and Joaquin Garcia-Alfaro
Institut Mines Telecom, Telecom SudParis
CNRS UMR 5157 SAMOVAR, Evry, France
first_name.last_name@telecom-sudparis.eu

ABSTRACT

This paper presents a geometrical model that projects malicious and benign events (e.g., attacks, security countermeasures) as pyramidal instances in a multidimensional coordinate system. The approach considers internal event data related to the target system (e.g., users, physical, and logical resources, IP addresses, port numbers, etc.), and external event data related to the attacker (e.g., knowledge, motivation, skills, etc.) that can be obtained a priori and a posteriori. Internal data is used to model the base of the pyramid, whereas external data is used to model its height. In addition, the approach considers state transitions taken by the attacker to model the steps of a multi-stage attack to reach to its final goal. As a result, for each modeled state, new countermeasures are evaluated and the attacker's knowledge a posteriori changes accordingly, making it possible to evaluate the impact of the attack at time T_i, where i denotes the stage at which the attack is executed. A graphical representation of the impact of each evaluated event is depicted for visualization purposes. A use case of a cyber-physical system is proposed at the end of the paper to illustrate the applicability of the proposed geometrical model.

CCS CONCEPTS

• **General and reference** → **Measurement**; **Metrics**; • **Information systems** → **Decision support systems**; • **Security and privacy** → **Formal security models**; • **Computing methodologies** → **Modeling and simulation**;

KEYWORDS

Pyramidal Model, Visualization, Geometrical Model, Countermeasure Selection, Event Impact Representation, Decision Support Tool.

ACM Reference Format:
Gustavo Gonzalez-Granadillo and Jose Rubio-Hernan, and Joaquin Garcia-Alfaro. 2018. A Pyramidal-based Model to Compute the Impact of Cyber Security Events. In *Proceedings of The 13th International Conference on Availability, Reliability and Security (ARES 2018) (ARES 2018).* ACM, New York, NY, USA, 10 pages. https://doi.org/https://doi.org/10.1145/3230833.3230847

1 INTRODUCTION

In order to properly prioritize the impact of cyber security events, organizations must be able to (i) understand the types of cyber risk they are exposed to; (ii) measure their associated rate of occurrence; and (iii) understand the business losses those risks are likely to produce. Costs associated with data breaches are the most widely understood impacts, however, information theft is not always an attacker's goal. Instead, attackers might be interested on events of intellectual property theft, espionage, data destruction, attacks on core operations, or attempts to disable critical infrastructures [1]. Such malicious events could have a more significant impact on organizations, but the actions taken by attackers are very difficult to model and quantify.

In addition, the deployment of appropriate countermeasures depends directly on the way cyber security information is displayed to the user. In this context, if the security information is displayed in the form of scrollable message lists, detailed reports or colored status icons, the security analyst will be inundated with huge amount of data that require to be processed before any meaningful action can be taken against detected threats [2]. Important information is often lost among the flood of non-important messages.

It is therefore important to design and develop rich visualization tools, able to handle diverse types of data that consider, for instance, attributes associated to each data type, temporal and spatial contexts, relational features, and information about the attacker's knowledge and motivation. As a result, appropriate visualization tools must be able to display not only information about the impact quantification of the detected event, but also its graphical representation to help operators in their analysis.

In this paper, we propose a geometrical model to compute and represent the impact of cyber security events as pyramidal instances. The model considers information about the target system (internal data) and information about the attacker (external data). It uses geometrical operations to compute the size of all instances, and thus to analyze the impact of all events detected in the system.

The contributions on this article include: (1) a methodology to compute the impact of cyber security events based on internal and external information; (2) a graphical representation of the impact of cyber security events as pyramidal instances; (3) a process that performs geometrical operations to analyze multiple simultaneous events based on their coverage, residual risk and potential collateral damage; and (4) the deployment of the model in an attack scenario with several events and multiple dimensions.

Paper organization — Section 2 provides related work. Section 3 details the construction of our model. Section 4 presents the graphical representation of events resulting from the proposed

ARES 2018, August 27-30, 2018, Hamburg, Germany

ACM ISBN 978-1-4503-6448-5/18/08…$15.00
https://doi.org/https://doi.org/10.1145/3230833.3230847

approach. Section 5 presents the geometrical computation of the impact of cyber security events. Section 6 provides an illustration of the proposed model. Section 7 concludes the paper.

2 RELATED WORK

Visualization models have different advantages that according to Kolomeec et al. [3], depend on the metrics used in the model construction and the context in which the model is used. Klein et al.[2] propose a visualization model to represent the current security situation of all protected resources based on event messages received from security components and appropriate available background information. By using this model, it is possible to employ intuitive visualization techniques to assist security operators with their tasks, (e.g., mitigation of attack effects, initiation of countermeasures, network information assurance provisioning). The main drawback of this model is that it does not consider external data in the attack impact computation.

Several research works rely on geometry to quantitatively measure the impact of cyber events. Gonzalez-Granadillo et al.[4], for instance, propose a prismatic model to compute the size and thus the impact of an instantiated event. For this, authors use a variety of data types obtained internally (from the system) and externally (as probabilistic values about the attacker's knowledge). The approach presents however the following drawbacks: (i) it uses an outdated CVSS version with variables and values that are no longer in used; (ii) it considers the a priori and a posteriori contribution as parameters that could reach a value of 100%, where in a realistic environment, the a priori knowledge of an attacker is far from reaching such a value; (iii) it does not consider the transitions taken by the attacker to successfully execute the attack.

While some researchers (e.g, Dini et al. [5]) rely on simulation tools to analyze and estimate the impact of cyber events, some others (e.g., Texeira et al. [6]) use frameworks to model the cyber physical attack space. The former approach allows to evaluate the attack effect and assesses them based on their severity. The main drawback of this approach is that the impact of countermeasures is barely considered in the simulation process. The latter approach classifies attacks in a cyber-physical system based on the adversary's knowledge, the disclosure, and the disruption of resources. However, authors concentrate only on the adversary a priori knowledge leaving aside the a posteriori knowledge.

Adversary models that allow understanding the attacker's behavior have been proposed by Krautsevich et al. [7] and Sarraute et al. [8]. The approaches model the attacker's knowledge. Pasqualetti et al. [9] survey the attack models in the cyber-physical systems, depending on the attacker's a priori knowledge about the system. And Rubio-Hernan et al. [10] define attackers able to acquire knowledge about the system behavior only by listening and analyzing the eavesdropped data in the communication channel. The main shortcoming of these researches is that their analysis is only based on the adversary's point of view.

Based on the aforementioned shortcomings, we propose a geometrical approach to project the impact of cyber events as an n-dimensional pyramidal instance. The approach uses geometrical operations to compute the volume of the pyramid, and thus the impact of the represented event. As a result, we are able to project multiple events (e.g., attacks, countermeasures) based on a variety of dimensions (e.g., users, channels, resources, attacker's knowledge, etc.) that consider not only internal information about the target system, but also external information about the attacker's knowledge and behavior.

3 PYRAMIDAL MODEL CONSTRUCTION

Pyramids are formed by connecting the polygonal base with a point, called the apex. Pyramids are classified according to [11] as: (i) the number of sides of the base (e.g., triangular, square, pentagonal, hexagonal); (ii) the location of the top (apex) of the pyramid (i.e., right, oblique); (iii) the shape of the base (i.e., regular, irregular), and (iv) the interior angles of the base (i.e., concave, convex).

The construction of the Pyramidal model considers two input parameters: The contribution of the internal event data and the contribution of the external event data. The former is used to build the base of the pyramid, and the latter is used to define the height of the pyramid (apex). The remaining of this section details the computation of each parameter and the graphical representation of the pyramidal instances.

3.1 Internal Event Data

All information of a security event associated to the target system is considered as internal data. This latter include information about the users interacting with the system (e.g., internals, externals, operators, etc); information about the system's resources (e.g., logical and physical resources); and information about the channels used by users to access resources (e.g., IP address, port number, credentials, protocols).

In addition, we consider the notion of contexts proposed in [12], such as temporal context (which depends on the time at which an action is executed); spatial context (which depends on the location); user-declared context (which depends on the objective of the user, e.g., user purpose); pre-requisite context (which depend on characteristics that join the subject, object and action, e.g., dependencies among entities); provisional context (which depend on previous actions performed by the user in the system).

Information security properties (e.g., confidentiality, integrity, availability, non-repudiation) are also key aspects in the analysis of the internal data associated to a cyber security event. An event can be associated to a particular issue compromising the system's confidentiality (e.g., unauthorized access to sensitive information, disclosure resources, etc), integrity (e.g., unauthorized change of the data contents or properties, etc), availability (e.g., unavailable resources, denial of service, etc), non-repudiation (e.g., services providing proof of the integrity and origin of data, authentication confirmed to be genuine with highly assurance).

Internal event data can be of two kinds: Logical or Physical. The former corresponds to all intangible data associated to the target system that can be used by an adversary to execute an attack whereas the latter corresponds to all tangible elements that interact with the target system and whose intrinsec vulnerabilities can be used by an adversary to execute an attack. Examples of logical data are application logs, memory and storage capacity, I/O bandwidth, security policies, IP addresses, open port numbers, etc. Examples of physical data are geographical location of people and devices, number of servers, printers and other technical equipment, physical characteristics of storage devices, etc.

Data types contribute differently to the execution of a security event. The contribution Co of each internal data type T_i in the execution of an event E is a value that ranges from zero (if no element of the axis is affected by a given event), to one (when all elements of the axis are affected to a given event). Notice that we consider dynamicity in the model by computing the impact of an attack at time T_i (i corresponds to the steps taken by the attacker to successfully exploit the systems' vulnerabilities). The contribution of the data type T_i is calculated as stated in Definition 1.

Definition 1 (Internal Event Data Contribution). *Let $X = x_1, x_2, ..., x_i$ be a finite set of size i, namely Total_E, and composed by the total number of elements integrating the system; and let $Y = y_1, y_2, ..., y_j$ be a finite set of size j, namely Affected_E, and composed by all affected elements of the system. Knowing that the set Y is a subset of X, thus $Y \subseteq X$, and that $WF(X_i)$ and $WF(Y_j)$ correspond to the weighting factor associated to each element from X and Y respectively, then, the contribution Co_{in} at time T in the execution of event E is computed using Equation 1.*

$$Co_{in}(T_i, E) = \frac{\sum_{j=1}^{n} Y_j \times WF(Y_j) \quad \forall j \in Y}{\sum_{i=1}^{n} X_i \times WF(X_i) \quad \forall i \in X} \tag{1}$$

In order to apply Equation 1 in a practical case, let us consider, for instance, the user account entity, whose contribution can be evaluated as the number of users affected by a given attack (Y) over the total number of active users from the system (X).

Each entity of the system is assigned a weighting factor based on its contribution to the given event. The weighting factor WF follows the CARVER methodology [13] that considers multiple criteria (i.e., Criticality, Accessibility, Recuperability, Vulnerability, Effect, and Recognizability) to assess each and every element using a scale of one to ten (one being of lowest importance, and ten being of highest importance) to the execution of the cyber security event. Examples of practical implementations of this methodology can be found in the work of Gonzalez-Granadillo et al., [14].

The base of the pyramid will be therefore equivalent to the polygon formed by linking all the entities from the internal event data contribution.

3.2 External Event Data

External event data correspond to the information related to the attacker in the execution of a malicious event. Depending on the time at which the analysis is performed, the external event data can be either a priori (i.e., set of information possessed by the attacker before exploiting a given vulnerability in the system) or a posteriori (i.e., set of information acquired by the attacker after successfully exploiting a given vulnerability in the system)[7].

3.2.1 External Event Data before executing the attack (a priori). According to the Common Vulnerability Scoring System (CVSS) v.3. [15], three main groups of data can be used to evaluate the a priori knowledge of an attacker: (i) Exploitability, which reflects the ease and technical means by which a vulnerability can be exploited; (ii) Impact, which reflects the direct consequence of a successful exploit, and represent the consequence to the thing that suffers the impact; and (iii) Temporal, which measures the current state of exploit techniques or code availability, the existence of patches or workarounds, and the confidence given in the description of a vulnerability.

The exploitability (Exp) group is modeled as a four-tuple: $Exp = (A_V, A_C, P_R, U_I)$, where A_V refers to the attack vector that could lead to a vulnerability exploitation (e.g., Network, Adjacent, Local, Physical); the values on this category will be larger the more remote an attacker will be to exploit a vulnerability component. A_C refers to the attack complexity, and describes the conditions beyond the attacker's control that must exist in order to exploit a given vulnerability (e.g., Low, High); the values on this category are larger for the least complex attacks. P_R refers to the level of privileges required by an attacker before successfully exploiting a given vulnerability (e.g., None, Low, High); the values are greater if no privileges are required. U_I refers to user interaction and captures the requirement for a user, other than the attacker, to participate in the successful compromise of a vulnerable component (e.g., None, Required), the value is greater when no user interaction is required.

The impact (Imp) group is modeled as a three-tuple: $Imp = (I_c, I_i, I_a)$, that measures the impact in terms of confidentiality (I_c), integrity (I_i), and availability (I_a) of the information resources managed by a software component due to a successfully exploited vulnerability (e.g., High, Low, None); the value increases with the consequence to the impacted component.

The temporal (Tem) group is modeled as a three-tuple: $Temp = (E_{CM}, R_L, R_C)$, where E_{CM} refers to the Exploit Code Maturity and measures the likelihood of the vulnerability being attacked (e.g., Not Defined, High, Functional, Proof-of-Concept, Unproven); the more easily a vulnerability can be exploited, the higher the E_{CM} score. R_L refers to the remediation level of a vulnerability (e.g., Not defined, Unavailable, Workaround, Temporary Fix, Official Fix); the less official and permanent a fix, the higher the R_L score. R_C refers to the report confidence and measures the level of credibility of the known technical details (e.g., Not Defined, Confirmed, Reasonable, Unknown); the more a vulnerability is validated by the vendor or other reputable sources, the higher the R_C score.

Besides the aforementioned CVSS metrics, a priori external event data also considers the attacker's final Goal Go in executing the attack. Go is modeled as a three-tuple: $Go = (M_O, S_K, R_E)$, where M_O considers the motivation that encourage an attacker to exploit a vulnerability on the system such as low (e.g., just for fun), medium (e.g., political motives), and high (e.g., for monetary profit; anger, revenge and other emotional drivers; sexual impulses; psychiatric illness) [16, 17]. S_K defines the level of skills and/or experience (e.g., high, medium, low) required by the attacker to execute a given attack. R_E considers the minimum level of resources (e.g., high, medium, low) required by the attacker to exploit a system's vulnerability;

The a priori contribution Co'_{ex} of an external entity at time Ti, in the execution of an event E is calculated using Definition 2.

Definition 2 (External Event Data Contribution (a priori)). *Let Exp be the level of exploitability associated to a vulnerability; Imp be the impact level in terms of confidentiality, integrity and availability; Go the attacker's final goal to accomplish the attack; and Tem be the temporal parameter associated to the vulnerability, the* *a priori external event data contribution (Co'_{ex}) is computed as the sum of the Exp, Imp, and Go parameters, times the Tem parameter, as depicted in Equation 2.*

$$Co'_{ex}(T_i, E) = \left(\frac{1}{2}Exp + \frac{1}{4}Imp + \frac{1}{4}Go\right) \times Tem \quad (2)$$

where
$Exp = A_V \times A_C \times P_R \times U_I$
$Imp = 1 - [(1 - I_c) \times (1 - I_i) \times (1 - I_a)]$
$Tem = E_{CM} \times R_L \times R_C$
$Go = M_O \times S_K \times R_E$

From Equation 2, values of *Exp*, *Imp*, and *Go* are normalized using the 1/2 and 1/4 multipliers, so that each of them contribute equally to the metric (i.e., each parameter ranges from 0 to 0.25). Their sum will therefore range from 0 to 0.75 and will be affected to the *Tem* parameter. The reason for the a priori metric to reach up to 0.75 is justified by the fact that a priori, the attacker does not have a total knowledge of the system being attacked. This metric is improved by the a posteriori knowledge acquired by the attacker during the execution of the attack. The following assumptions have been considered:

- While constructing a profile of the typical attacker, we look at general characteristics of all attackers, historical events, and statistics that provide generalities of all possible categories of attacks. These characteristics are seen as probabilities, not as absolute values. There are exceptions to each case. However, a majority of attackers exhibit common patterns that can be identified in advance. More information of attack profiles, goals and motivations can be found in [17];
- An attacker spends certain amount of time units to perform the attacks. He/she spends a unit of time for executing a single attack step. The attacker stays in a goal state if he/she reaches it before spending all units of time;
- The real system is separated from the attacker belief about the system. According to Krautsevich et al. [7], attackers can be either omniscient (when they know all system's vulnerabilities), deterministic (when they have a belief knowledge of the system and they choose the best possible action to break in), or adaptive (when they adapt the strategy to complete). Omniscient attackers are not realistic as they must have the total view of the system. In a more realistic case, the attacker's view does not fully coincide with the real system. The attacker's knowledge about the system determines the set of vulnerabilities that the attacker believes present in the system.

Based on the previous assumptions, we have set the attacker's a priori knowledge about the system to reach up to 75% (only for deterministic and adaptive attackers). The a priori knowledge may be improved once the attack is executed and the attacker gains some privileges on the system (i.e., a posteriori knowledge). This latter could eventually reach up to 100%. Table 1 depicts the possible values of all a priori parameters (P).

Note that levels and values refer to the corresponding qualitative and quantitative assessments of each parameter. Our approach considers the scores proposed in the CVSS v3.0 for the exploitability, Impact and Temporal parameters. For the rest of values, we consider three levels (e.g., low, medium, high) in which the lowest score is 1/3, the medium score is 2/3 and the highest score is 3/3.

3.2.2 *External Event Data after executing the attack (a posteriori).*

During the execution of an attack, the attacker learns about the countermeasures the targeted system has deployed as a defense mechanisms. Countermeasures can be defined to protect the system or to react against malicious activities. The knowledge possessed by an attacker to overpass the system's protecting measures represents the external event data a posteriori.

Several research studies [7, 16, 17] have been conducted to estimate the attacker's behavior, knowledge, motivations and skills in exploiting system's vulnerabilities. A wide number of metrics (e.g., anonymity, diversity, closeness, etc) is proposed in the literature to evaluate and analyze the attacker's actions a posteriori, but also to predict them before an event happens. In addition, information-theoretic metrics such as Shannon's entropy, min-entropy, or mutual information, could be construed as particular cases of this estimation.

According to [7], an attacker observes a system and can influence its behavior by making actions at a given moment. The system responds to an action probabilistically. Attackers do not make decisions about actions blindly. Instead, they take into account past, current, and possible future states of the system, as well as possible rewards that are connected with the actions. The goal of the attacker is to maximize the expected total reward according to a sole criterion. For empirical studies that discuss models to measure the ability of an attacker to guess the password of a user, and the application of guessing entropy as the measurement of the expected number of guesses an attacker would need in order to guess a password, please check [18, 19].

As a result, it is possible from the security analyst perspective, to estimate the attacker's knowledge under certain situations. In the end, these estimations are probabilities that are based on assumptions, expert knowledge, statistical data, and/or simulation models that help in the decision making process.

Following the approach presented in [4], we modeled the attacker's a posteriori knowledge as a four-tuple $A''_K = (P_A, P_S, R_W, D_E)$, where P_A considers the probability that the attacker executes an action in the state S_{Ti}, the higher the number of possible actions for the attacker to exploit a vulnerability, the greater the P_A score. P_S corresponds to the probability that the system transits from state S_{Ti} to S_{Ti+1} in response to attacker's actions, the higher the probability that the system will react against the attacker's actions, the lower the P_S score. R_W considers a set of rewards functions dependent on the state and the actions, the higher the level of reward, the greater the R_W score. D_E corresponds to a set of decisions available to the attacker, the higher the number of decisions, the greater the D_E score. The a posteriori contribution Co''_{ex} of an external entity at time Ti, in the execution of an event E is calculated as stated in Definition 3.

Definition 3 (External Event Data Contribution (a posteriori)). *Let P_A and P_S be respectively the probability of the attack and the system to execute a given action in response to the enemy's action; and let R_W and D_E be respectively the reward and the decisions available to the attacker, the a posteriori external event data contribution Co''_{ex} for an event E at time T_i is computed as the sum of*

Table 1: Quantitative values of parameters a priori

P	Level	Value	Level	Value	Level	Value	Level	Value	Level	Value
A_V	Network	0.85	Adjacent Net.	0.62	Local	0.55	Physical	0.20		
A_C	Low	0.77	High	0.44						
P_R	None	0.85	Low	0.62	High	0.27				
U_I	None	0.85	Required	0.62						
I_c, I_i, I_a	High	0.56	Low	0.22	None	0.00				
E_{CM}	Not defined	1	High	1	Functional	0,97	Proof-of-concept	0,94	Unproven	0.91
R_L	Not defined	1	Unavailable	1	Workaround	0.97	Temporary	0.96	Official Fix	0.95
R_C	Not defined	1	Confirmed	1	Reasonable	0.96	Unknown	0.92		
M_O	Low	0.34	Medium	0.67	High	1.0				
S_K	High	0.34	Medium	0.67	Low	1.0				
R_E	High	0.34	Medium	0.67	Low	1.0				

the attacker's knowledge contribution before and after the execution of a given attack, as depicted in Equation 3.

$$Co''_{ex}(T_i,E) = Co'_{ex}(T_i,E) + \frac{1}{8}\left[(P_A \times P_S) + (R_W \times D_E)\right] \tag{3}$$

Note that from Equation 3, parameters such as P_A, P_S, R_W, and D_E are normalized by the 1/8 multiplier, so that each of them contributes equally to the computation of the $Co''_{ex}(T_i,E)$ metric, with $Co'_{ex}(T_i,E)$ ranging from 0 to 0.75, and the sum of the rest of parameters ranging from 0 to 0.25. The height of the pyramidal instance will range from zero to one and will be equivalent to the value of $Co''_{ex}(T_i,E)$. Table 2 depicts the possible values of the parameters composing Equation 3.

Table 2: Quantitative values of parameters a posteriori

Parameter	Level	Value	Level	Value	Level	Value
P_A	Low	0.34	Medium	0.67	High	1.0
P_S	High	0.34	Medium	0.67	Low	1.0
R_W	None	0.0	Partial	0.5	Total	1.0
D_E	None	0.0	Single	0.5	Multiple	1.0

3.3 Attack Transitional Steps

In order to consider the transitions taken by the attacker to successfully execute the attack, we need to include the parameter time (T_i) in the analysis. At T_0, for instance, the attacker has not yet exploited any vulnerability. He/she has a priori knowledge about the system and the defense mechanisms that could be used in case of attack. Since this knowledge could be very limited, its contribution is expected to be no greater than 0.75. The contribution about the a priori knowledge of the attacker at time T_0 is therefore computed using Equation 2.

Once the attacker has penetrated the system, he/she starts gaining some knowledge about the defense mechanisms already in place. The contribution about the a posteriori knowledge of the attacker at time T_0 is therefore computed using Equation 3. This latter corresponds to the sum of the *a priori* and *a posteriori input* of the attacker's knowledge, as depicted in Figure 1. As most of the current attacks perform their actions in a sequential way, which requires several stages to successfully execute the attack, we consider the attacker's knowledge transitions from State T_{i-1} to State T_i. In this context, we consider that the attack has started at T_1, therefore the transition state T_0 is modeled as the steps performed by the attacker prior exploiting a system's vulnerability.

We can define mathematically the external entity contribution $C_{o_{ex}}(T_i,E)$ at time T_i as:

$$C_{o_{ex}}(T_i,E) = C''_{o_{ex}}(T_i,E) \qquad \forall i \geq 0 \tag{4}$$

Where $i \in \mathbb{T} = [0, \tau, ..., n\tau]$ $\forall n \in \mathbb{N}$. T_i: $i > \tau$ is the attack moment.

$$C_{o_{ex}}(T_i,E) = \begin{cases} \underbrace{\underbrace{C'_{o_{ex}}(T_0,E)}_{\textit{a priori}} + \underbrace{C^*_{o_{ex}}(T_i,E)}_{\textit{a posteriori input}}}_{\textit{a posteriori}} & i = 0 \\ \underbrace{\underbrace{C'_{o_{ex}}(T_i,E)}_{\textit{a priori}} + \underbrace{C^*_{o_{ex}}(T_i,E)}_{\textit{a posteriori input}}}_{\textit{a posteriori}} & i > 0 \end{cases} \tag{5}$$

Following Definition 2 and Definition 3 and the execution of the attack previously explained, we can state that $C'_{o_{ex}}(T_i,E)$ and $C^*_{o_{ex}}(T_i,E)$ depend on the instant T_i as follows:

$$C'_{o_{ex}}(T_0,E) = \left[\left(\frac{1}{2}Exp + \frac{1}{4}Imp + \frac{1}{4}Go\right) \times Tem\right] \tag{6}$$

$$C'_{o_{ex}}(T_i,E) = \frac{3}{4}C''_{o_{ex}}(T_{i-\tau},E) \qquad \forall i > 0 \tag{7}$$

$$C^*_{o_{ex}}(T_i,E) = \frac{1}{8}\left[(P_A \times P_S) + (R_W \times D_E)\right] \tag{8}$$

Where Equation 6 and Equation 7 are the *a priori* external contribution at T_0 and T_i $\forall i > 0$ respectively; and Equation 8 is the *a posteriori input* of the external contribution of an attack produced at time T_i. Note that going from the *a priori* to the *a posteriori* knowledge within a given transitional state (T_i) implies that the attacker's knowledge value will be normally increased due to the

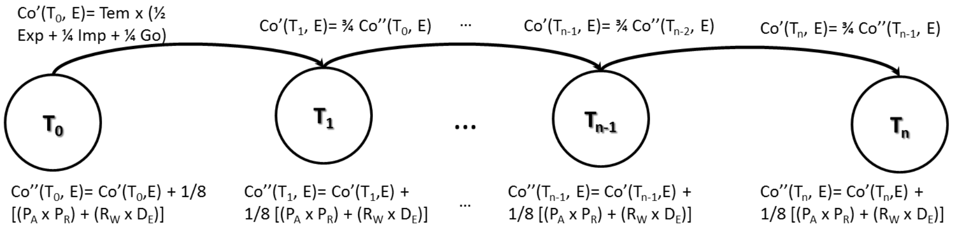

$Co'(T_0, E)$= Tem x (½ Exp + ¼ Imp + ¼ Go)
$Co'(T_1, E)$= ¾ $Co''(T_0, E)$
...
$Co'(T_{n-1}, E)$= ¾ $Co''(T_{n-2}, E)$
$Co'(T_n, E)$= ¾ $Co''(T_{n-1}, E)$
T_0
T_1
...
T_{n-1}
T_n
$Co''(T_0, E)$= $Co'(T_0,E)$ + 1/8 [(P_A x P_R) + (R_W x D_E)]
$Co''(T_1, E)$= $Co'(T_1,E)$ + 1/8 [(P_A x P_R) + (R_W x D_E)]
...
$Co''(T_{n-1}, E)$= $Co'(T_{n-1},E)$ + 1/8 [(P_A x P_R) + (R_W x D_E)]
$Co''(T_n, E)$= $Co'(T_n,E)$ + 1/8 [(P_A x P_R) + (R_W x D_E)]

Figure 1: Attack Transitional Steps

knowledge acquired by the attacker thanks to the observation of the system's behavior.

Going from one state to a subsequent one (i.e., from T_i to T_{i+1}) implies that the attacker's knowledge value will be normally reduced, since new countermeasures could be taken by the target system and the attacker needs to learn such actions in order to reach to its final stage. In this context, the contribution of the attacker's knowledge a priori at time T_i will be equivalent to 3/4 of the contribution of the attacker's knowledge a posteriori at time T_{i-1} (not applicable to T_0 since there is no previous state). As a result, the a priori contribution at state T_i will range from 0 to 0.75, and the a posteriori contribution at state T_i will range from 0 to 1. Being this latter the value used to compute the height of the pyramidal instance.

4 GRAPHICAL REPRESENTATION

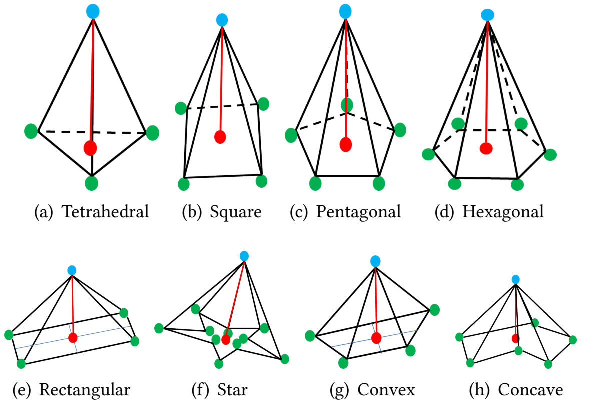
A variety of geometrical instances results from the analysis of the internal and external information related to a given cyber security event. The construction of our pyramidal model is limited to the following geometrical instances:

Regular/Irregular Pyramids: These instances include pyramids with a regular/irregular n-sided polygon as the base. Examples of these isntances are square pyramid, pentagonal pyramid, hexagonal pyramid, etc. If the base of the pyramid is a triangle, the instance is called a tetrahedron [20].

Right Pyramids: These instances include pyramids (regular and/or irregular) that have their apex aligned directly above the center of its base. Examples of right pyramids are rectangular pyramids, rhombic pyramids, star pyramids. These latter are right pyramids with star polygon bases e.g., pentagrammic, heptagrammic) [21].

Convex and Concave Pyramids: These instances include pyramids whose interior angles of the base are either less than 180 degrees (convex)or greater than 180 degrees (concave).

The following geometrical instances are discarded from our pyramidal model: Bipyramid (i.e., formed by joining an n-gonal pyramid and its mirror image base-to-base); Cubic Pyramid (i.e., bounded by one cube on the base and six square pyramid cells which meet at the apex); Octahedral Pyramid (i.e., bounded by one octahedron on the base and eight triangular pyramid cells which meet at the apex); Icosahedral Pyramid (i.e., a four-dimensional convex polytope, bounded by one icosahedron as its base and by 20 triangular pyramid cells which meet at its apex); Oblique Pyramid(i.e., a pyramid with the apex not directly above the center of its base); and Polyhedral Pyramids (i.e., 4-dimensional pyramid).



(a) Tetrahedral (b) Square (c) Pentagonal (d) Hexagonal

(e) Rectangular (f) Star (g) Convex (h) Concave

Figure 2: Graphical representation of Pyramidal Instances

Plotting the contribution of three or more internal entities (e.g., user, physical resource, logical resource, channel, time, location, etc.) results into an n-sided polygon (e.g., pentagon, hexagon, heptagon, etc.). Polygons can be regular or irregular. The contribution of the external axis (attacker's knowledge) allows to project the polygon in a 3D space, resulting in an n-sided pyramid, n being the number of sides of the polygonal instance (e.g., pentagonal pyramid, hexagonal pyramid, heptagonal pyramid, etc.), as depicted in Figure 2.

5 IMPACT COMPUTATION

We propose to compute the impact of a security event as the size of its corresponding geometrical instance. A pyramidal instance can be measured either by its surface area or by its volume. The surface area of a pyramid is computed as shown in Equation 9.

$$A = \begin{cases} B + \frac{P \times L}{2} & \text{(Regular Pyramid)} \\ 2B + L & \text{(Irregular Pyramid)} \end{cases} \quad (9)$$

Where
B = the base area of the pyramid, which depends on the shape of the polygonal base;
P = the base perimeter;
L = the lateral area of the pyramid. When all side faces are the same, $L = \sqrt{h^2 + r^2}$, where h is the pyramid altitude and r is the inradius of the base. When side faces are different, we must add the area of each triangle to find the total lateral area.

The volume of a pyramid is computed as one third of the product of its base area and its height, as shown in Equation 10. This latter works for any polygon (regular or irregular), and any location of the apex, provided that h is the height measured as the perpendicular distance from the plane containing the base.

$$V = \frac{b \times h}{3} \tag{10}$$

Where
b = area of the base, considering that the base of the pyramid is formed by linking all the entities from the internal event data contribution $Co_{in}(T_i, E)$.
h = the height from the base to the apex, equivalent to $Co''_{ex}(T_i, E)$.

Following the approach presented in [22], b is computed as the sum of the contribution value of the internal event data E_i times the contribution value of the internal event data E_{i+1} divided by two, as shown in Equation 11.

$$b = \frac{\sum_{i=1}^{n} Co_{in}(T_i, E_i) \times Co_{in}(T_i, E_{i+1})}{2} \tag{11}$$

For the previous Equation, note that in the last term (i.e., $Co_{in}(T_i, E_n)$), the expression must wrap around back to the first term (i.e., $Co_{in}(T_i, E_i)$). This method works correctly for triangles, regular and irregular polygons, as well as convex and concave polygons, but it will produce wrong answers for self-intersecting polygons, where one side crosses over another. However, such cases are excluded from our research.

6 USE CASE

This section provides a use case of a cyber-physical system in a railway infrastructure. We define a progressive multi-step attack, whose goal is to disrupt the system slowly and stealthily. The attack mixes ICT and control domains and uses sequential steps to disrupt the system. Hereinafter we refer to this attack as a Cyber-Physical Sequential *CPS* attack. Countermeasures have been defined for each step of the attack. Attacks and countermeasures are modeled and analyzed using the proposed pyramidal model for impact assessment. We aim at demonstrating how our model computes the impact of cyber security events addressed in a cyber-physical system. In this scenario, the attacker is able to acquire knowledge (e.g., physical behavior, network communication) about the system and improve their actions.

The described use-case is based on a real testbed developed in our labs where an attack of a cyber-physical system has been implemented and analyzed. The testbeds aim at validating the security of existing SCADA protocols, such as the Modbus, DNP3, and IEC 104 protocols and allowed us to model the use case proposed in the paper. More information about the testbed can be found at: http://j.mp/TSPScada.

6.1 General Description

The system used in our use case is a railway infrastructure. It has two main block elements: controller environment, and data environment. The former has two separated but correlated sub-blocks: (1) a physical process and data controller; and (2) a network Observer. The latter has also two sub-blocks: (1) network parameters and devices (e.g., protocols, switches); and (2) physical process data and devices (e.g., sensors, actuators, sensor data). The framework uses SCADA technology to implement the hierarchical architecture of our cyber-physical system and the SCADA protocols for the communication between the physical devices (e.g., sensors, actuators) and the controller. The network observer checks the network in order to verify that the system works properly. Such observer is connected to the physical controller in order to detect threats and activate the corresponding security countermeasures. Note that the cyber-physical system used in our testbed has sensors and actuators as physical devices distributed across several nodes using the network to connect them to the physical controller. In Figure 3 we show a use case representation where [S_1, S_2, ..., S_{10}] represent the sensors, [M_1, M_2, ..., M_6] represent the track change engines and [T_0, T_1] represent the trains (cf. http://j.mp/TSPScada).

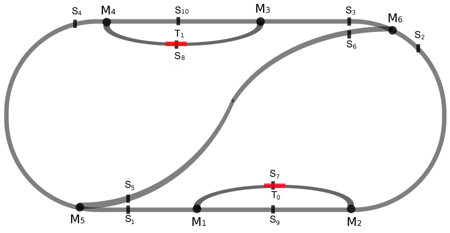


Figure 3: Railway use-case

6.2 Description of Cyber Security Events

Attack Scenario: Cyber-Physical Sequential - CPS Attack. We assume that the attacker has the capability to intercept the communication between the different devices. The CPS attack has a passive and active operation mode executed in the following slow and stealthy steps:

(i) *Assumptions:* this is the initial stage (transition T_0) explained in Section 3.3, in which the attacker decides to break into the system based on the assumptions he/she has about the target system and its protection measures.

(ii) *Network Recognition:* this is the first action performed by the attacker once he/she is inside the system (transition T_1). The adversary listens to the communication channels in order to eavesdrop the system without modifying any communication parameter. The goal of this stage is to identify any resource and/or device containing potential sensitive information.

(iii) *Network and Data Analysis:* The attacker analyzes the communications and processes the obtained information (between transition T_1 and T_2). This increases his/her knowledge about the channels and the end devices used. The data analysis allows the attacker to acquire knowledge about the behavior of the global system and its vulnerabilities.

(iv) *Vulnerability Exploitation:* The attacker moves to an active mode. He/she tries to exploit some SCADA protocol vulnerabilities to gain as much information as possible about the target system (transition T_2). After which, the attacker starts to disrupt stealthily the system (transition T_3): (a) by exploiting database vulnerabilities that allows him/her to escalate privilege and obtain administrator access to sensitive information; and (b) by exploiting the knowledge acquired about the behavior of the global system.

(v) *Attack Completion:* After exploiting the system vulnerabilities, the attacker succeeds in obtaining the access to the command and control (transition T_4), he/she is able to mislead the control center with false data, forcing it to send corrupt data able to disrupt the trains behavior.

Countermeasures: Several actions can be implemented by the target system in order to mitigate the malicious actions introduced by the attacker. A non-exhaustive list of countermeasures (hereinafter denoted as CM) is provided below:

CM1 Activate physical watermark-based detection to check the physical process and increase communication security;
CM2 Signal and model-based intrusion detection to improve traceability/integrity of communication data;
CM3 State relation-based detection to identify anomalies;
CM4 Physical process control distribution to prevent an adversary from gaining enough knowledge of the system;
CM5 Message counter to avoid replay attacks;
CM6 Activate heartbeat messages among the different devices in order to create a security communication among them;
CM7 Install pre-computed traffic path implementing specific security policies to control the traffic through the network;
CM8 Activate degraded mode of the protocol to mitigate integrity attacks;
CM9 Update firewall to stop the DDoS attacks;
CM10 Increase physical redundancy with trusty sensors, actuators, and remote devices to detect and mitigate anomalies;
CM11 Network segmentation to avoid attack propagation.

The aforementioned list of countermeasures is proposed to be used depending on two elements: (i) the malicious action(s) detected by the system; and (ii) the history of the malicious actions throughout the time (i.e., transitional stage $[T_0, T_1, T_2, ...T_n]$). For instance, CM_1 and CM_2, are implemented in the system before the beginning of the attack, that is at time T_0; CM_3, CM_5, and CM_9 are proposed for the transition T_1; CM_8, and CM_{11} are proposed for the transition T_2; CM_4, CM_6, and CM_7 are proposed for the transition T_3; and CM_{10} is proposed for the transition T_4. Note that at time T_i it is also possible to implement the non-implemented countermeasures for the previous stage (T_{i-1}). The remaining of this section shows their impact values and graphical representation.

6.3 Impact Computation and Graphical Representation

This section will detail the impact computation of all cyber security events and its graphical representation as pyramidal instances. Table 3 shows the information about the entities composing the target system.

Entities belong to one of the four main dimensions (i.e., physical resources, logical resources, channels, and users). Each entity has at least one element (N) with an associated weighting factor (WF), and a value in the coordinate system.

Based on statistical data and expert knowledge, we have identified the affected entities for each cyber security event analyzed in the system. In order to compute the impact of each event, we need to compute the length of each side of the polygonal base. In this case, since we have information of four dimensions (i.e., physical resource, logical resource, channel, and user), the resulting figure will be a polygon of four sides (i.e., quadrilateral).

Table 3: System Information

Dimension	ID	Description	N	WF	Coordinate
Physical Resource	P1	Server	1	3	0:3
	P2	Router	1	3	3:6
	P3:P4	RTU	2	3	6:12
	P5:P6	Controller	2	4	12:20
	P7:P8	Workstation	2	4	20:28
	P9:P18	Sensor	10	5	28:78
Logical Resource	L1	Front End	1	2	0:2
	L2	Database	1	3	2:5
	L3:L5	System PLC	3	4	5:17
	L6	Back End	1	4	17:21
	L7:L8	Train PLC	2	4	21:29
	L9:L16	Actuator	8	5	29:69
	L17:L25	Firewall	9	5	69:114
Channel	C1:C20	UDP Port	20	2	0:40
	C21:C29	User Cred.	9	3	40:67
	C30:C283	Private IP	254	3	67:829
	C284:C288	Public IP	5	4	829:849
	C289:C318	TCP Port	30	4	849:969
	C319	DNP3	1	4	969:973
	C320	Modbus	1	4	973:977
	C321	Adm. Cred.	1	5	977:982
User	U1:U2	External	2	1	0:2
	U3:U10	Internal	8	2	2:18
	U11:U13	Owner	3	3	18:27
	U14	Stakeholder	1	3	27:30
	U15:U18	SCADA Op.	4	4	30:46
	U19	Admin	1	5	46:51

In order to compute the area of a quadrilateral using its coordinates, we need to first identify the affected entities and the contribution of each entity type by using Equation 1. For instance, A_{ST1} affects P3, P4, P9:P18, which corresponds to 72% of physical resources; L1, L6, L17:L25, which corresponds to 51% of logical resources; C284:C318, which corresponds to 14% of channels; and U1:U10 which corresponds to 35% of users. Then, we use Equation 11 to obtain the area of the instantiated event, which in this case is equivalent to 0.34 $units^2$. Results of the area of all studied cyber security events are summarized in Table 4.

Note that for CM_1 and CM_2, since they are deployed at time T_0, they are discarded in our analysis. The rest of countermeasures are proposed to be evaluated at a given stage of the attack (if and only if the attack reaches such a stage). The last column of Table 4 shows the coverage (COV) of one event over another. The coverage of attack A is computed based on the internal entities from system S that are affected by such attack at a given stage. The coverage of a countermeasure is computed as the percentage of the attack mitigated by a countermeasure. For instance, A_{ST1} affects 17.22% of the entities from system S, whereas CM_3 covers 91.21% of the internal entities affected by A_{ST1}.

Table 4: Internal Event Data

Event	**Physical Res.**	**Logical Res.**	**Channel**	**User**	**Area** ($units^2$)	**COV** (%)
S	P1:P18	L1:L25	C1:C321	U1:U19	2.00	-
A_{St1}	P3, P4, P9:P18	L1, L6, L17:L25	C284:C318	U1:U10	0.34	17.22
A_{St2}	P1:P4, P7:P18	L1: L6, L17:L25	C30:C320	U1:U10, U15:U18	1.13	56.55
A_{St3}	P1:P4, P7:P18	L1: L6, L9:L25	C1:C320	U1:U18	1.73	86.66
A_{St4}	P1:P18	L1:L25	C1:C321	U1:U19	2.00	100.00
CM_3	P1:P18	L2, L17:L25	C30:C283, C289:C320	U1:U10, 15:U19	1.13	91.21
CM_4	P5:P18	L2:L25	C30:C320	U3:U10, U15:U19	1.51	73.06
CM_5	P1:P6	L1:L8, L17:L25	C284:C318	U3:U19	0.32	24.26
CM_6	P3:P4, P9:P18	L2, L17:L25	C30:C320	U15:U18	0.60	34.69
CM_7	P3:P4, P9:P18	L17:L25	C30:C283, C289:C320	U15:U18	0.58	33.20
CM_8	P1:P4, P9:P18	L3:L5, L7:L25	C30:C283, C289:C320	U15:U18	1.05	61.20
CM_9	P9:P18	L17:L25	C284:C320	U3:U10	0.28	80.61
CM_{10}	P1:P18	L9:L25	C30:C283	U3:U10, U15:U19	1.32	65.32
CM_{11}	P1:P4, P7:P8	L1, L17:L25	C30:C320	U15:U18	0.43	37.71

In addition to internal data, we compute the height of the pyramidal instance based on the information about the attacker's knowledge (external data). Table 5 shows the information of the attacker's a priori and a posteriori knowledge about system S, attack A, and countermeasures (from CM_1 to CM_{11}). Note that the values of parameters for attack A are obtained from the National Vulnerability Database[1] and are given to each state of the sequential attack. The values for the system S are obtained by taking a pessimistic approach (the worst case scenario); and the values of parameters for the countermeasures CM_1 to CM_{11} are first assessed qualitatively and then transformed into their corresponding quantitative values according to Tables 1 and 2.

Table 5: External Event Data

A Priori Knowledge

Event	A_V	A_C	P_R	U_I	I_c	I_i	I_a	E_{CM}	R_L	R_C	M_O	S_K	R_E
S	0.85	0.77	0.85	0.85	0.56	0.56	0.56	1.0	1.0	1.0	1.0	1.0	1.0
A_{St1}	0.85	0.44	0.27	0.62	0.56	0.0	0.0	0.91	1.0	1.0	0.67	1.0	1.0
A_{St2}	0.85	0.44	0.62	0.85	0.56	0.0	0.0	0.97	1.0	1.0	0.67	1.0	1.0
A_{St3}	0.85	0.77	0.62	0.85	0.56	0.56	0.56	1.0	0.96	1.0	1.0	0.34	0.34
A_{St4}	0.85	0.77	0.62	0.85	0.56	0.56	0.56	1.0	0.96	1.0	1.0	0.34	0.34

In order to compute the height h and the volume of the pyramidal instance, we need to calculate the contribution of the external event data a priori and a posteriori using Equations 2 and 3 respectively.

A Posteriori Knowledge

Event	P_A	P_S	R_W	D_E	$Height$	$Volume$ ($units^3$)	COV (%)
S	1.0	1.0	1.0	1.0	0.97	0.65	-
A_{St1}	1.0	1.0	0.5	1.0	0.50	0.06	8.87
A_{St2}	1.0	1.0	0.5	1.0	0.58	0.22	34.01
A_{St3}	0.67	0.34	1.0	1.0	0.57	0.33	50.81
A_{St4}	0.34	0.34	1.0	0.5	0.49	0.33	50.52
CM_3	1.0	1.0	0.50	1.0	0.56	0.21	91.30
CM_4	0.67	0.34	1.0	1.0	0.58	0.29	73.22
CM_5	1.0	1.0	0.50	1.0	0.56	0.06	24.28
CM_6	0.67	0.34	1.0	1.0	0.58	0.12	34.77
CM_7	0.67	0.34	1.0	1.0	0.58	0.11	33.27
CM_8	0.67	0.67	0.50	1.0	0.55	0.19	57.70
CM_9	1.0	1.0	0.50	1.0	0.56	0.05	80.69
CM_{10}	0.34	0.34	1.0	0.5	0.46	0.19	58.65
CM_{11}	0.67	0.67	1.0	1.0	0.55	0.08	35.56

The value of h for system S and all transitional steps taken by the attack A is equivalent to the sum of the a priori and a posteriori external event data contribution. For countermeasures CM_1 to CM_{11}, we compute the h of the pyramid as the sum of 3/4 of the a priori external event data contribution and 1/8 of the a posteriori external event data contribution. The volume of the pyramidal instance is computed using Equation 10.

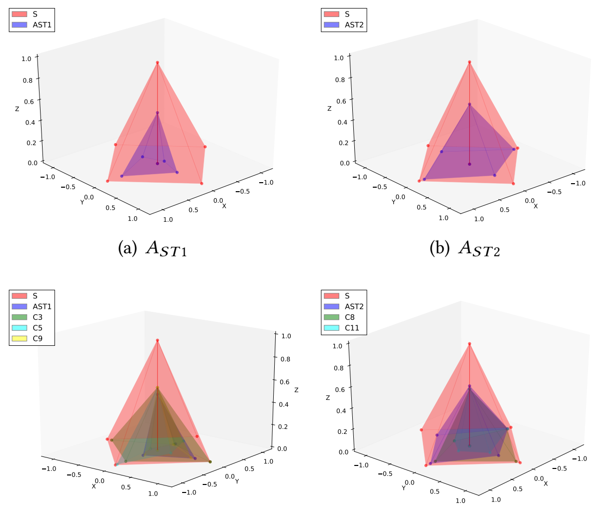
The coverage shown in Table 5 compares the volume of one event over another. The coverage of each stage of attack A is compared against the volume of system S, whereas the coverage of a countermeasure is compared against the volume of the attack it mitigates. For instance, A_{ST2} affects 34.01% of the internal and external entities from system S, whereas CM_8 protects 57.70% of the internal and external entities affected by A_{ST2}.

Figure 4 shows the graphical representation of attacks A_{ST1}, A_{ST2} and their proposed countermeasures. The system is represented in pink, the different stages of the attack A is represented in purple, and countermeasures are represented in green, blue and yellow colors. As it can be seen from Figures 4(a) to 4(b), the volume of each stage of attack A is comprised withing the boundaries of the volume of system S.

The visualization of cyber attacks and countermeasures in the same geometrical space helps security analysts in the evaluation and selection of countermeasures as a response to cyber attacks. It is possible to identify priority areas, and perform reaction strategies accordingly. From Figures 4(a) to 4(d), it is possible to visualize the portion of the system that is attacked and the portion of the attack that is being controlled by a countermeasure, as well as the portion of the attack left with no treatment (e.g., residual risk).

Figure 4(c), for instance, shows the graphical representation of the system S (in red), attack A_{ST1} (in blue), and countermeasures CM_3 (in green), CM_5 (in light blue) and CM_9 (in yellow). This representation allows us to identify the portion of the system compromised by A_{ST1} and the coverage of each countermeasure. In this example, even with the simultaneous implementation of the three countermeasures, attack A_{ST1} will not be totally covered. These types of representations are very useful in identifying residual risks (i.e., the portion of the system's elements being attacked that are

[1] https://nvd.nist.gov/



(a) A_{ST1} (b) A_{ST2}

(c) CMs against A_{ST1} (d) CMs against A_{ST2}

Figure 4: Impact representation of all stages of Attack A

not covered by any countermeasure), as well as potential collateral damage (i.e., the portion of the system that is not under attack but requires modifications e.g., configuration changes, during the implementation of the security controls).

In addition, implementing several countermeasures simultaneously generally implies that the protected area will increase, which reduces the residual risk and makes the solution look more attractive than their individual implementation. However, this is not always the case. Therefore, the graphical representation of security events as pyramidal instances will help security analysts to identify scenarios for optimal countermeasure implementation. Figure A_{ST2}, for instance, shows that the implementation of CM_{11} is useless if combined with the implementation of CM_8, as this latter protects a wider region of the system under attack and totally covers the region protected by CM_{11}.

7 CONCLUSION

In this paper we present a geometrical model to compute the impact of cyber events as pyramidal instances. The approach considers internal data about the target system (e.g., users, resources, channels), and external data about the attacker (e.g., knowledge, motivation, skills). The former are used to compute the base of the pyramid, whereas the latter are used to compute its height.

The model differentiates between the information possessed by the attacker before executing the attack (a priori) and after its execution (a posteriori). It uses the Common Vulnerability Scoring System (CVSS) v.3 to assess exploitability, impact and temporal parameters related to the potential and/or executed attacks.

In addition, we consider the attack transitional steps to model the actions taken by the target system and its adversary at a particular period of time. As a result, we add dynamicity to the model by computing the impact of an attack at time T_i where i corresponds to the steps taken by the attacker to successfully exploit the systems' vulnerabilities, considering that at each stage of the attack, the target system will deploy new defense mechanisms.

Cyber security events are projected in a multi-dimensional coordinate system to form pyramidal instances whose volume represent the size and thus the impact of the studied events. It is then possible to evaluate multiple and complex attacks and select the best mitigation strategy. Future work will compare different geometrical approaches to identify their limitations and verify their usefulness.

Acknowledgements — The authors acknowledge support from the Cyber CNI Chair of the Institut Mines-Télécom (cf. http://chaire-cyber-cni.fr/). The chair is supported by Airbus Defence and Space, Amossys, EDF, Orange, La Poste, Nokia, Société Générale and the Regional Council of Brittany. It has been acknowledged by the Center of excellence in Cybersecurity.

REFERENCES

[1] E. Mossburg, J. Gelinne, H. Calzada, "Beneath the surface of a cyberattack. A deeper look at business impacts", *Deloitte Technical Paper*, Last accessed Feb. 2018.

[2] G. Klein, C. Ruckert, M. Kleiber, M. Jahnke, J. Toelle, "Towards a Model-Based Cyber Defense Situational Awareness Visualization Environment", *RTO Workshop Visualising Networks: Coping with Chance and Uncertainty*, 2010.

[3] M. Kolomeec, G. Gonzalez-Granadillo, E. Doynikova, A. Chechulin, I. Kotenko, H. Debar, "Choosing Models for Security Metrics Visualization", *Mathematical Methods, Models and Architectures for Computer Networks Security Conference*, 2017.

[4] G. Gonzalez-Granadillo, J. Rubio-Hernan, J. Garcia-Alfaro, H. Debar, "Considering internal vulnerabilities and the attacker's knowledge to model the impact of cyber events as geometrical prisms", *Conference on Trust, Security and Privacy in Computing and Communications*, 2016.

[5] G. Dini and M. Tiloca, *A simulation tool for evaluating attack impact in cyber physical systems.* Int. Workshop MESAS, pp. 77-94, 2014.

[6] A. Teixeira, I. Shames, H. Sandberg, and K. Johansson, *A secure control framework for resource-limited adversaries.* Automatica, vol. 51, pp. 135–148, 2015.

[7] L. Krautsevich, F. Martinelli, and A. Yautsiukhin, *Towards modelling adaptive attacker's behaviour.* FPS Symposium, pp. 357-364, 2013.

[8] C. Sarraute, O. Buffet, J. Hoffmann, *POMDPs make better hackers: Accounting for uncertainty in penetration testing.* arXiv preprint arXiv:1307.8182, 2013.

[9] F. Pasqualetti, F. Dorfler and F. Bullo, *Control-Theoretic Methods for Cyberphysical Security: Geometric Principles for Optimal Cross-Layer Resilient Control Systems* in IEEE Control Systems, vol. 35, no. 1, pp. 110-127, Feb. 2015.

[10] J. Rubio-Hernan, L. De Cicco and J. Garcia-Alfaro, *Adaptive Control-Theoretic Detection of Integrity Attacks against Cyber-Physical Industrial Systems.* Transactions on Emerging Telecommunications Technologies, ISSN: 2161-3915, August 2017.

[11] P. Brown, M. Evans, D. Hunt, J. Mclntosh, B. Pender, J. Ramagge, "Cones, Pyramids and Spheres (Measurement and Geometry: Module 12)", *Australian Mathematical Sciences Institute (AMSI)*, 2011.

[12] F. Cuppens and N. Cuppens-Boulahia, *Modeling contextual security policies.* International Journal of Information Security, 7(4):285-305, 2008.

[13] T. L. Norman, *Risk Analysis and Security Countermeasure Selection.* CRC Press, Taylor & Francis Group, 2010.

[14] G. Gonzalez-Granadillo, J. Garcia-Alfaro, H. Debar, "A Polytope-based approach to measure the impact of events against critical infrastructures", *Journal of Computer and System Sciences*, Vol. 83(1),pp. 3–21, 2016.

[15] Forum of Incident Response and Security Teams, *Common Vulnerability Scoring System v3.0: Specification Document.* Technical Paper, 2015.

[16] M. Bielecki and G. Quirchmayr, *A prototype for support of computer forensic analysis combined with the expected knowledge level of an attacker to more efficiently achieve investigation results.* International Conference on Availability, Reliability and Security, pp. 696-701, 2010.

[17] D. Shinder, *Scenes of the cybercrime. computer forensics handbook.* Syngress Publishing Inc., 2002.

[18] Usenix, "On User Choice in Graphical Password Schemes", available at:https://www.usenix.org/legacy/publications/library/proceedings/sec04/tech/full_papers/davis/davis_html/node8.html

[19] J. L. Massey, "Guessing and entropy", *In Proceedings of the International Symposium on Information Theory*, 1994.

[20] W. F. Kern, J. R. Bland, "Solid Mensuration", *John Wiley and sons Inc., London: Chapman and Hall, Limited*, 1934.

[21] M. J. Wenninger, "Polyhedron Models", *Cambridge University Press*, 1974.

[22] G. Gonzalez-Granadillo, J. Garcia-Alfaro, and H. Debar, "An n-sided polygonal model to calculate the impact of cyber security events", *11th Conference on Risks and Security of Internet and Systems*, 2016.