A security metric for assessing the Security Level of Critical Infrastructures

Andrea Tortorelli[1], Andrea Fiaschetti[1], Alessandro Giuseppi[1], Vincenzo Suraci[1,2], Roberto Germanà[1], and Francesco Delli Priscoli[1]

Abstract The deep integration between the cyber and physical domains in complex systems make very challenging the security evaluation process, as security itself is more of a concept (i.e. a subjective property) than a quantifiable characteristic. Traditional security assessing mostly relies on the personal skills of security experts, often based on best practices and personal experience. The present work is aimed at defining a security metric allowing evaluators to assess the security level of complex Cyber-Physical Systems (CPSs), as Critical Infrastructures, in a holistic, consistent and repeatable way. To achieve this result, the mathematical framework provided by the Open Source Security Testing Methodology Manual (OSSTMM) is used as the backbone of the new security metric, since it allows to provide security indicators capturing, in a non-biased way, the security level of a system. Several concepts, as component Lifecycle, Vulnerability criticality and Damage Potential – Effort Ratio are embedded in the new security metric framework, developed in the scope of the H2020 project ATENA.

Keywords: *Security Metrics; Critical Infrastructures; Cyber-Physical Systems; Cyber-Physical Security;*

1. INTRODUCTION

Critical Infrastructures (CIs) and, in general, modern complex systems, give rise to new and complex challenges from the security point of view (Alcaraz & Zeadally, 2015; Di Mase, Collier, Heffner, & Linkov, 2015; Mo et al., 2012). The deep integration between the cyber and physical domains, indeed, requires protecting heterogeneous cyber/physical resources in different environments (e.g. (Adamsky et al., 2018; Di Giorgio, Liberati, Lanna, Pietrabissa, & Priscoli, 2017; Frezzetti & Manfredi, 2019; Kourtis et al., 2017) for the energy and communication environments, respectively); moreover, vulnerabilities in the cyber domain can increase the attack surface thus introducing vulnerabilities also in the physical domain, and vice-versa (Wells, Camelio, Williams, & White, 2014). Furthermore, security evaluators must have a strong knowledge of both domains. In addition, complex systems are composed of a huge number of components which may be built in different countries thus developed under different regulations. This aspect renders difficult the evaluation of the security level of the whole system since capturing vulnerabilities introduced by components with unsecured production chains is not an easy task. This problem occurs also with legacy components whose presence is common in CIs. One of the main challenges when it comes to security is how to quantify the security level of a system. In most cases, the evaluation of security is performed by checking the compliance with guidelines and rules specified by security standards. In most cases, these standards provide qualitative measures or discrete levels for characterizing security (e.g. the ISO/IEC 15408 and the ISA/IEC 62443 series). However, the lack of quantitative measures impairs the repeatability of the security evaluation process which renders difficult the creation of a common ground which can be used to compare the security testing results and to understand how the evaluated system scores with respect to others. Guaranteeing consistency and repeatability of a security test is indeed one of the biggest challenges in security. Consistency requires to identify those aspects concurring to the security characterization which is intrinsically difficult. Repeatability requires that such aspects can be evaluated in a rigorous and non-ambiguous way. The purpose of the present work[3], indeed, is to describe a security metric, suitable for the CI

[1] The authors are the with the Department of Computer, Control and Management Engineering (DIAG) "Antonio Ruberti" of the University of Rome "La Sapienza", Via Ariosto 25, 00185, Rome, Italy, tortorelli@diag.uniroma1.it.

[2] The author is with the "Università degli Studi eCampus", Via Isimbardi 10, 22060, Novedrate (CO), Italy.

[3] This work has been carried out in the framework of the ATENA project which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700581. This work reflects only the author's view. The EU Commission and the Research Executive Agency are not responsible for any use that may be made of the information it contains.

domain, and able to capture relevant security aspects in a consistent and repeatable way.

The reminder of the paper is organised as follows: Section 2 reports a review on current security standards highlighting the current limitations; Section 3 summarises the main concepts behind OSSTMM underlying how it allows to overcome the mentioned limitations; Section 4 reports the proposed security metrics and extensions; Section 5 reports a use case example for qualitative validation, while Section 6 draws the conclusions and highlights possible future works.

2. SECURITY STANDARDS AND METHODOLOGIES

Many security standards, tools and frameworks have been developed for providing a measure of the security level of complex systems.

Security standards, such as ((ISA), n.d.; ISO/IEC, 2020; National Institute of Standards and Technologies (NIST), 2008), provide a collection of guidelines and best practices identifying the security aspects that should be addressed. More specifically, security standards also provide a description of the actions that should be implemented or checked to be compliant to given security levels. In this context, several tools, such as (Open Web Application Security Project (OWASP), 2020b, 2020a) and (Center for Internet Security (CIS), 2020; OpenSCAP, 2019), have been developed to check the compliance with one or more security standards. In other words, the security level is characterized in terms of the compliancy degree to given sets of guidelines and best practices. Based on such checks, these tools are able to suggest, or even prioritize, the countermeasures to implement. The same approach is also implemented by frameworks such as (ISACA, 2019). Although useful in practical applications, these approaches have the following drawbacks. First, security is characterized in terms of compliance with given sets of guidelines (different for each tool) which, in turn, renders difficult to compare results and to create a common ground which can be used by organizations to understand their relative positioning. Second, such description of security is qualitative (or at most discretized) which renders impossible to discriminate between similar security configurations. This aspect may lead to overspending in countermeasures which, in turn, may lead to a higher attack surface.

Risk-based approaches, such as (Ahmed, Al-Shaer, Taibah, & Khan, 2011; “ISO/IEC 27005:2018, Information Technology - Security Techniques - Information Security Risk Management,” 2018; “ISO 31000:2018 - Risk Management,” 2018; Saripalli & Walters, 2010), measure security as a function of the probability that given threats can actually affect assets. Other methodologies, such as (Gadyatskaya et al., 2016; Mauw & Oostdijk, 2006; Schneier, 2015), measure security based on the system response in given attack scenarios. These two approaches, although commonly adopted, are prone to consistency and repeatability issues because risk depends on features difficult to compute (e.g. impact, probability of occurrence) and it is not possible to consider all the possible attack scenarios that could effectively occur.

Mathematical approaches , such as (Herzog, 2016; Morgagni, Fiaschetti, Noll, Arenaza-Nuño, & Del Ser, 2017; Rehak, Senovsky, Hromada, & Lovecek, 2019), on the other hand, provide formal frameworks guaranteeing consistency and repeatability of security tests. The problem is their description capability since it is difficult to model given features concurring to the security level. In other words, the choice is between qualitative instruments specifying actions that should be implemented or verified for achieving a given security level and quantitative instruments defining rules to compute numerical values characterizing relevant security features and thus the overall security level.

The approach described in the present work can be cast into the above-mentioned mathematical approaches and addresses the highlighted criticalities. In particular, the aim is to define a security metric (i) with enhanced description capabilities and (ii) allowing to capture relevant security features. To achieve this, the mathematical framework set up by the Open Source Security Testing Methodology Manual (OSSTMM) (Herzog, 2016) is retained and

extended by considering the guidelines of other recognized security standards in order to extend its description capabilities. This can be achieved due to the flexibility of the OSSTMM framework which, together with its formal means to assess security (peculiar of mathematical approaches), lead the authors' choice to retain it as the reference framework. As detailed in Section 4, to overcome the mentioned drawbacks of mathematical approaches, the guidelines defined in recognized security standards have been considered and translated into quantitative properties. In other words, the proposed security metric is able to characterize security in a broader way meaning that the security features that widely recognized standards defined as relevant to characterize security are captured.

3. THE OSSTMM METHODOLOGY

The Open Source Security Testing Methodology Manual (OSSTMM) constitutes the backbone of the proposed security metric. The framework provided by such methodology, indeed, from the one hand guarantees the repeatability of security tests and, from the other hand, is very keen to be extended in order to take in consideration the security aspects mentioned in the previous sections. More specifically, the proposed extensions, described in detail in the next chapter, allow embedding in the OSSTMM framework cost-benefit considerations from the attacker point of view, lifecycle aspects and the different severity of vulnerabilities.

Although the terminology in the literature is very variegate, it is possible to define the main concepts at the basis of security: *assets* are those valuable elements that should be protected, *threats* are what jeopardize the system, and *countermeasures* are protection mechanism that can be put in place to protect the assets and lower the effect, or eliminate, threats. However, when it comes to defining what security actually is, there is no consensus in the literature.

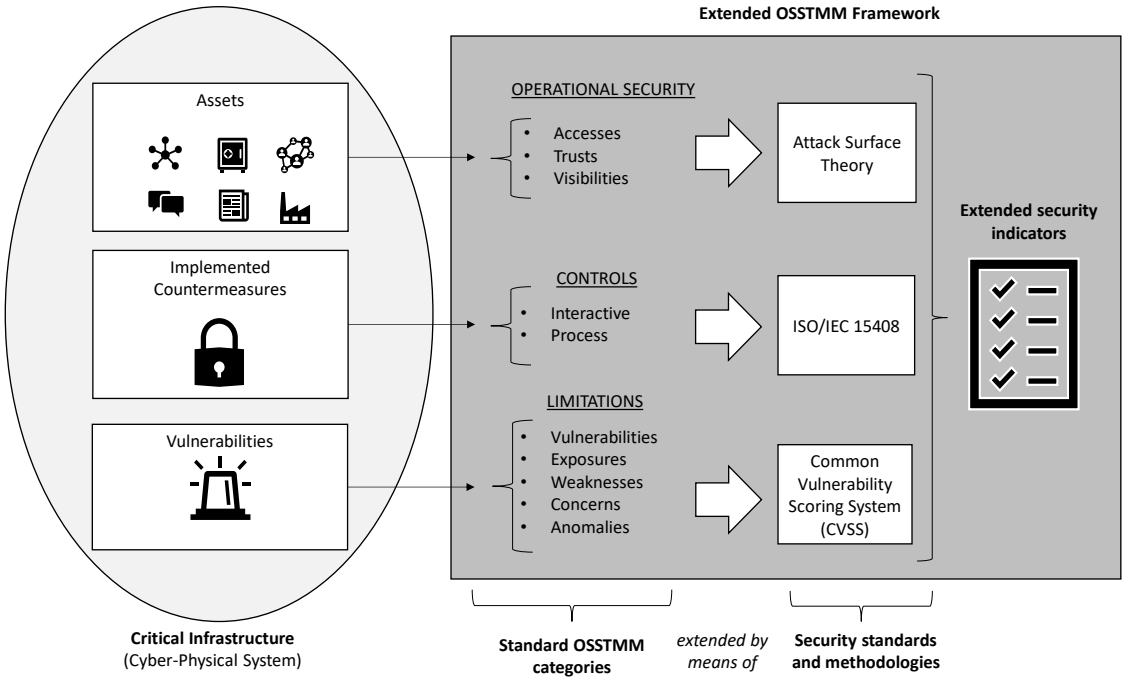
The OSSTMM methodology is based on the idea that security is a function of the separation between the identified assets and existing threats. That is, to reach perfect security a complete (physical or logical) separation between assets and threats has to be guaranteed. In real-world scenarios, a complete separation cannot be enforced because, otherwise, the system functionalities would be impaired. In other words, for the system to be accessible and to work in a proper and useful way, a lack of separation between assets and threats is required. For example, a building without doors and windows cannot be accessed by unauthorized users but, at the same time, it would be useless since neither authorized users could access it.

Following this concept, the OSSTMM focuses on the so-called *Operational Security* which represents the lack of separation between assets and threats that do exist in operational environments. In the OSSTMM, such loss of separation is also referred to as *Porosity* which is a function of three elements: the points from which an interaction with the external world can occur (*Access*) which have to be there for operational reasons, the interactions occurring within the system itself (*Trusts*) which are needed to implement system functionalities and the number of known assets within the scope (*Visibility*). The latter element allows to characterize opportunity: unknown assets are not in danger of being targeted but only of being discovered.

Porosity represents one of the three categories that the OSSTMM considers in order to characterize the security level of a given system. The second category is *Controls* which represents the protection mechanisms that can be put in place for lessening the system exposure (described by Porosity). In the OSSTMM methodology, countermeasures are categorized in ten classes covering all the possible ways in which it is possible to protect interactions (*Interactive Controls*) or processes (*Process Controls*).

The deficiencies of controls protection mechanisms and the problems in maintaining the separation between assets and threats are referred to as *Limitations*; five classes of Limitations are defined in the OSSTMM: *Vulnerabilities* and *Exposures* (impacting system operations), *Concerns* and *Weaknesses* (reducing controls effectiveness) and *Anomalies* (unknown

Figure 1: Extended OSSTMM framework



and not controlled events which cannot be accounted for in normal operations).

Based on the three above-mentioned categories (Porosity, Controls and Limitations), the OSSTMM methodology provides a set of security indicators which, when considered collectively, allow to characterize the security level of a given system. These indicators are reported in a spreadsheet, referred to as RAV (Risk Assessment Values), which provide a snapshot of the security level of the system in terms of the system exposure, the implemented controls and the vulnerabilities.

In the next chapter, all the mentioned elements will be detailed along with the proposed extensions (see **Errore. L'origine riferimento non è stata trovata.**).

4. EXTENDED OSSTMM

The proposed security metric extends the OSSTMM and in particular defines an extended version of Porosity, by means of the Attack Surface theory (Manadhata & Wing, 2011), Controls, by means of Common Criteria (ISO/IEC, 2020), and Limitations, by means of the Common Vulnerability Scoring System (CVSS) (FIRST, 2015). This is achieved by parameterizing the OSSTMM categories (Porosity, Controls and Limitations) by means of recognized security standards and methodologies (namely the Attack Surface theory, Common Criteria and CVSS).

The security indicators obtained by combining the extended versions of these three categories allow capturing new and innovative security aspects not covered by the OSSTMM and specified as relevant by many security standards and methodologies (e.g. IEC 62443 and ISO/IEC/IEEE 15288).

In the following, with *Standard OSSTMM* it is meant the OSSTMM as defined in (Herzog, 2016) while with *Extended OSSTMM* it is meant the OSSTMM with the modifications proposed in this paper.

4.1. EXTENDED POROSITY

The Porosity, also referred to as Operational Security ($OpSec_{\Sigma}$), characterize the system exposure to threats, i.e. the lack of separation between assets and threats and is computed based on three elements:

- Visibility (P_V) is a mean to characterize opportunity (which is a crucial aspect encouraging attacks) and consists in the number of known assets (which can be targeted) within the scope

- Access (P_A) is the number of points of the system from which interaction with the external world can occur
- Trust (P_T) is the number of interactions occurring between the elements of the system

In Standard OSSTMM, the $OpSec_\Sigma$ is computed as the sum of these three elements:

$$OpSec_\Sigma = P_V + P_A + P_T \qquad (1)$$

For taking into consideration the fact that not all pores (i.e. elements concurring to the system exposure) have the same impact, it is provided with an extended version of Porosity. To achieve this, the notion of Damage potential – Effort Ratio (DER) introduced in (Manadhata & Wing, 2011) is embedded in the OSSTMM framework. The DER is a mean to characterize the fact that gaining access to a resource may require different effort from the attacker point of view and, once the resource has been impaired, may provide different privileges to the attacker. In other terms, the DER provides a cost-benefit description of resources from the attacker point of view. This aspect (i.e. taking into consideration attacker resources) has been highlighted in several standards as a relevant aspect that the security evaluator should take into account (ISO/IEC, 2020; Mauw & Oostdijk, 2006). Indeed, an attacker may find more appealing to target a resource with high Damage potential even if for gaining access to it requires a high Effort. Although the Damage potential (DP) and the Effort (E) can be computed separately, in the proposed extension they are jointly accounted for since from the attacker point of view, these parameters are linked as already discussed. This approach is in line with what done in (Manadhata & Wing, 2011). By exploiting the notion of DER, it is thus possible to redefine Access (i.e. P_A) as

$$P'_A \coloneqq \sum_{i=1}^{P_A} \frac{DP_i}{E_i} = \sum_{i=1}^{P_A} DER_i \qquad (2)$$

where DER_i is the DER computed for the i-th Access point.

A further proposed extension derives from the consideration that the proposed security metric should be tailored to the specific context of CIs. For this reason, it has been envisaged the possibility of assigning different weights to the three elements concurring to the system exposure, i.e. to Porosity. Indeed, the impact of each of the three Porosity elements on the security level of a given CI can be different. For example, it may happen that in a specific domain the visibility of assets is not very relevant since the opportunity of actually targeting them is extremely remote. Following on these considerations, it is possible to define an extended version of Porosity (referred to as *Weighted Porosity*) as

$$OpSec_\Sigma^W \coloneqq \alpha P_V + \beta P'_A + \gamma P_T \qquad (3)$$

where α, β and γ are static weights reflecting environmental peculiarities of a given CI.

4.2. EXTENDED CONTROLS

Controls constitute the second element concurring to the characterization of the system security level. Several indicators can be associated with Controls such as their optimal distribution or coverage with respect to the system exposure. To define such indicators, it is necessary to introduce some parameters. First, recall that in the OSSTMM ten types of Controls have been identified and cast in two classes:

- Interactive Controls including: Authentication, Indemnification, Resilience, Subjugation, Continuity
- Process Controls including: Non-repudiation, Confidentiality, Privacy, Integrity, Alarm

Loss Controls (LC_i) are defined as those measures put in place to protect the system functionalities and can be particularized for each one of the ten control categories just recalled. The total number of Loss Controls is stored in the LC_{sum} parameter defined as

$$LC_{sum} = \sum_{i=1}^{10} LC_i \qquad (4)$$

In the proposed security metric, the definition of Loss Controls is extended for taking into account

the component lifecycle. Indeed, components (and their security functionalities) with an unsecure value chain introduce additional entry and exit points for attackers, with an effect that can be hidden in nominal operations or postponed in time. This implies a potential mismatch in the evaluation of the system exposure, between the expected reliability of the resources deployed for security purposes and its actual reliability. In the literature, quality or security standards for lifecycle certification are typically based on the verification that given properties or procedures are satisfied, i.e. it is reduced to the evaluation of a "check-list" of security requirements that assess, on a static, self-standing way, whether the lifecycle is secure or not. An approach particularly suitable to be embedded in the OSSTMM framework is the one adopted by the ISO/IEC 15408, used for certifying products and systems (the focus is actually on the former). Note that, given the generality of the approach, any standard or methodology based on check-lists of security requirements or procedure can be used.

The ISO/IEC 15408 certification is based on an analysis aimed at assuring that the requirements of several *assurance classes* are met; the specific requirements of each class assure a different level of security and thus it is possible to discriminate between the products. Such security levels, in the ISO/IEC 15408, have been identified with the *Evaluation Assurance Levels* (EALs). The EALs are seven hierarchically ordered levels which are used for rating the security level that a product or system can guarantee. The Lifecycle support class (one of the mentioned assurance classes), denoted with "ALC", allows the identification of many checks (i.e. security requirements to be satisfied during the development and maintenance phases) which have to be verified by the product for being certified. The ALC class, considered as baseline implementation in this section, consists of seven families (ISO/IEC, 2020) each focusing on a specific security aspect associated with the lifecycle.

Within each family, several assurance components are defined. Assurance components, which are organized in a hierarchically way, describe a set of security requirements that if are all satisfied renders the assurance component satisfied as well. In general, higher EALs require, for each assurance family, a higher number of satisfied assurance components (and consequently of verified security requirements): in a nutshell, the more controls are satisfied, the more assurance components are verified, and the higher lifecycle assurance level is reached, family by family. However, such an approach does not allow to discriminate between cross-border situations. Indeed, it is possible that two components have the same EAL but one satisfies more security requirements (even if not enough to satisfy a higher assurance component and thus achieving a higher EAL). In other words, considering only 7 discrete values of EALs does not provide the adequate granularity to measure the overall lifecycle assurance level, and, consequently, the assurance components should be combined, or interpolated, so as to quantify cross-border situations. In addition, one of the goals was to move from a look-up table-based approach into a more "mathematical" and quantitative approach.

In order to do so, discrete values of EAL ([1 2 3 4 5 6 7]) can be translated into numerical values directly related to the verified security requirements and assurance components. To do so, the first step of the adopted strategy consists in computing, within each assurance family, the number of verified security requirements (r_i) versus the total number of security requirements for this family (R_i). Then, it is necessary to compute the weight (l_i) of each assurance family of the ALC assurance class to the product lifecycle. In the adopted strategy, all the families have been considered equally important for the evaluation of the overall lifecycle, and thus all the families have been assigned the same weight; nevertheless, in other contexts, these weights can be different. Finally, the weighted contributions of all the families are summed thus providing a numerical quantification of the so-called *lifecycle value* (γ_{LC}) able to also quantify cross-border situations (i.e. partial verification of EALs). Practically speaking, it is sufficient to:

1. count the number of security requirements verified for each family (r_i),
2. assess the weight of each family (l_i) and

3. sum all the individual contributions to obtain the overall lifecycle value (γ_{LC}).

Translated into a formula:

$$\gamma_{LC} = \sum_{i=1}^{N} l_i \left(\frac{r_i}{R_i}\right) \quad (5)$$

where r_i is the number of verified requirements within the $i-th$ assurance family, R_i is the total number of requirements within the the $i-th$ assurance family, N is the number of considered assurance families and l_i is the weight of each family. In the current modelling, $N = 6$ and $l_i = 1/N$.

Following these considerations, for embedding lifecycle aspects into the RAV computation, Loss Controls are weighted based on the lifecycle values (i.e. γ_{LC}). In other words, for each one of the ten Controls categories, it is possible to define the *Actual Controls* (AC_i) accounting for the background lifecycle of components. Taking in consideration, for example, the Authentication class, the relevant Actual Control parameter (AC_{Au}) is computed accordingly to the following formula

$$AC_{Au} = \sum_{j=1}^{LC_{Au}} \sum_{i=1}^{N} \frac{r_i}{R_i} \cdot l_i \quad (6)$$

The total number of Actual Controls is stored in the AC_{sum} parameter.

In the Standard OSSTMM, key parameters are the *Missing Controls* (MC_i) which account for the portion of the system exposure which is not protected. In the Standard OSSTMM, Missing Controls are calculated separately for each Loss Control category in the following way:

$$MC_i = \begin{cases} 0 \text{ , if } OpSec_{\Sigma} - LC_i \leq 0 \\ OpSec_{\Sigma} - LC_i \text{ , else} \end{cases} \quad (7)$$

The sum of the values of all the missing Controls is denoted as MC_{sum}.

By substituting in equation (7) Actual Controls (i.e. the parameters AC_i as defined in equation (6)) in place of Loss Controls, it is possible to define a modified version of the Missing Controls which allows to embed lifecycle aspects. For each Controls category, it is possible to define the *Actual Missing Controls* (AMC_i) in the following way:

$$AMC_i = \begin{cases} 0 \text{ , if } OpSec_{\Sigma} - AC_i \leq 0 \\ OpSec_{\Sigma} - AC_i \text{ , else} \end{cases} \quad (8)$$

The sum of the values of all the Actual Missing Controls is denoted with AMC_{sum}.

For taking into consideration the different contributions of pores to the security level, it is convenient to substitute in equation (7) the Weighted Operational Security in place of $OpSec_{\Sigma}^{W}$ computed as in equation (3); the resulting parameters are referred to as *Weighted Missing Controls* (MC_i^W). This means that equation (7) becomes

$$MC_i^W = \begin{cases} 0 \text{ , if } OpSec_{\Sigma}^{W} - LC_i \leq 0 \\ OpSec_{\Sigma}^{W} - LC_i \text{ , else} \end{cases} \quad (9)$$

Then, for taking into consideration, at the same time, even lifecycle aspects and the different contributions of pores, the computation of Missing Controls can be modified as follows:

$$MMC_i = \begin{cases} 0 \text{ , if } OpSec_{\Sigma}^{W} - AC_i \leq 0 \\ OpSec_{\Sigma}^{W} - AC_i \text{ , else} \end{cases} \quad (10)$$

where MMC_i, with $i = 1, \dots, 10$, are just the *Modified Missing Controls* for each of the ten control categories mentioned at the beginning of this section. The parameter MMC_{sum}, defined as the sum of the ten MMC_i, is used to compute the total number of the Modified Missing Controls.

In the Standard OSSTMM, *Missing coverage* ($MCvg$) is an indicator allowing to characterize the number of not protected system operations (i.e. the Missing Controls) with respect to the Operational Security and is computed as

$$MCvg = \begin{cases} 0 \text{ , if } OpSec_{\Sigma} \leq 0 \\ \dfrac{MC_{sum} \times 0.1}{OpSec_{sum}} \text{ , else} \end{cases} \quad (11)$$

This parameter can be also modified taking into account the three above-described extensions (i.e. the Actual, Weighted and Modified Missing Controls). By doing so, the modified parameters (indicated as *Actual* ($AMCvg$), *Weighted*

($MCvg^W$) and *Modified Missing Coverage* ($MMCvg$), respectively) can be obtained following the same logic adopted for extending Missing Controls.

Starting from the Loss and Missing Controls, it is possible to define three important parameters: the *True Controls* (TC), the *True Coverage* ($TCvg$) and the *Full Controls* (FC). The former, which can be determined for each one of the ten categories of Missing Controls (i.e. $TC_i, \ i = 1, \dots, 10$), provides a measure not only of the amount of implemented Controls, but also on their placement and is computed as

$$TC_i = OpSec_{\Sigma} - MC_i \qquad (12)$$

The sum of all the true Controls is stored in the parameter TC_{sum}. As already done for the other parameters, True Controls can be extended for taking in consideration the components lifecycle (by so doing, it is possible to define the *Actual True Controls*, ATC_i) and the extended porosity (by so doing, it is possible to define the *Weighted True Controls*, WTC_i). The *Modified True Controls* (MTC_i) allows to simultaneously take into account both aspects and it is computed substituting in equation (12) $OpSec_{\Sigma}^{W}$ (as defined in equation (3)) and MMC_i (as defined in equation (10)). The parameters ATC_{sum}, WTC_{sum} and MTC_{sum} allow to compute the total number of Actual, Weighted and Modified True Controls, respectively, and are defined as the sum of ATC_i, WTC_i and MTC_i, respectively.

In the Standard OSSTMM, the *True Coverage* ($TCvg$) parameter is based on the same idea of True Controls but it is expressed as a percentage (%) and is computed as:

$$TCvg = \begin{cases} 0 \text{ , if } OpSec_{\Sigma} \leq 0 \\ 1 - \dfrac{TC_{sum}}{10 \times OpSec_{sum}} \text{ , else} \end{cases} \qquad (13)$$

The extended versions of this parameter, referred to as *Actual True Coverage* ($ATCvg$), *Weighted True Coverage* ($WTCvg$) and *Modified True Coverage* ($MTCvg$), can be computed following the same logic as above; they allow to take in consideration lifecycle aspects, different impact of impaired pores and both aspects at the same time, respectively.

Full Controls (FC), unlike True Controls, account for the countermeasures put in place regardless from their category. This means that, in order to compute this parameter, it is not necessary to separately consider each Controls category. Full Controls can be computed as

$$FC_{base} = \log^2(1 + 10 \times LC_{sum}) \qquad (14)$$

This parameter can be extended for taking in consideration lifecycle aspects by defining the *Actual Full Controls* (AFC):

$$AFC_{base} = \log^2(1 + 10 \times AC_{sum}) \qquad (15)$$

4.3. EXTENDED LIMITATIONS

The third OSSTMM category having an impact on the final security assessment consists in the Limitations (Vulnerabilities, Weaknesses, Concerns, Exposures and Anomalies) accounting for the presence of flaws or errors increasing the system exposure. In the Standard OSSTMM, each Limitation is individually weighted according to the following table

	Weights
Vulnerabilities	$W_V = \dfrac{(OpSec_{\Sigma} + MC_{sum})}{OpSec_{\Sigma}}$
Weaknesses	$W_W = \dfrac{(OpSec_{\Sigma} + MC_A)}{OpSec_{sum}}$
Concerns	$W_C = \dfrac{(OpSec_{\Sigma} + MC_B)}{OpSec_{\Sigma}}$
Exposures	$W_E = \dfrac{\left((P_V + P_A) \times MCvg + L_V + L_W\right.}{OpSec_{\Sigma}}$
Anomalies	$W_A = \dfrac{(P_T \times MCvg + L_V + L_W + L_C)}{OpSec_{\Sigma}}$

Table 1. Limitations weights

That is, Limitations are accounted for in the following way. First, it is computed the number of instances of each type of Limitations, resulting in the computation of the parameters L_V, L_W, L_C, L_E and L_A; thus these parameters can be defined as the number of Vulnerabilities, Weaknesses, Concerns, Exposures and Anomalies, respectively.

Then, the weights W_j reported in Table 1 are computed. Finally, the Security Limitations parameter ($SecLim_{\mathrm{sum}}$) is defined as a collection of all the information regarding the Limitations as

$$SecLim_{sum} = L_V \times W_V + L_W \times W_W + L_C \times W_C + L_E \times W_E + L_A \times W_A \quad (16)$$

The Security Limitations parameter can be easily extended for considering lifecycle aspects and the different contribution of pores. For this purpose, it is sufficient to extend the weights reported in Table 1. Indeed, by considering the Actual Missing Controls in place of Missing Controls, the extended Porosity defined in equation (2) in place of the one defined in equation (1) and both extensions at the same time, it is possible to obtain the so-called *Actual Security Limitations* ($ASecLim_{sum}$), *Weighted Security Limitations* ($WSecLim_{sum}$) and *Modified Security Limitations* ($MSecLim_{sum}$), respectively.

The extended security parameters derived so far have been developed starting from two modifications: The first one based on the Attack Surface theory and impacting Porosity, the second one based on Common Criteria and impacting Controls. In the following, a third modification, impacting the last OSSTMM category (namely Limitations) will be described. This extension aims at capturing the fact that not all vulnerabilities have the same impact on the system exposure. For this purpose, the vulnerability scoring system developed by the National Institute of Standards and Technology (NIST), namely the Common Vulnerability Scoring System (CVSS) (FIRST, 2015), is used to weight Limitations accordingly to their severity. These severity scores are retained from the CVE database (MITRE, 2005) and computed according to the procedure defined by the CVSS methodology.

That is, when computing the contribution of Limitations, instead of considering the number of instances of Limitations for each class (i.e. the parameters L_V, L_W, L_C, L_E and L_A) their CVSS base scores are used as shown below

$$L'_j = 0.1 * \sum_{i=1}^{L_j} L_{j_{base_i}} \quad (17)$$

where j discriminates among the five classes of Limitations L_j, $L_{base,i}$ is the CVSS base score as defined in (FIRST, 2015). The CVSS scores (which take values in the range $[0; 10]$) are bounded to the interval $[0; 1]$ so that the balance among the three OSSTMM categories (Porosity, Controls and Limitations) is not impaired. It is worth noticing the fact that the 0 value for CVSS scores (referring to the latest version of the CVSS methodology, i.e. v3.0) is never assumed in practice[1]; therefore, existing vulnerabilities will not be neglected due to a low score.

For taking into account these extensions, the parameters L_V, L_W, L_C, L_E and L_A appearing in equation (16) are replaced by the parameters L'_j (with $j = 1, \dots, 5$) defined in equation (17); the resulting parameter, referred to as Adjusted Security Limitations ($AdjSecLim_{sum}$), is computed as

$$AdjSecLim_{sum} = L'_V \times W_V + L'_W \times W_W + L'_C \times W_C + L'_E \times W_E + L'_A \times W_A. \quad (18)$$

In order to jointly consider lifecycle aspects, the extended porosity and the severity scores of Limitations, it is sufficient to replace the weights in equation (18) with their modified version already discussed, thus obtaining the so-called *Adjusted Modified Security Limitations* parameter ($AdjMSecLim_{sum}$).

4.4. EXTENDED ADDITIONAL SECURITY INDICATORS

As a preliminary observation, note that each one of the three extensions described in Sections 4.1-4.3 directly impacts one of the three OSSTMM categories: the Attack Surface-based extension

[1] A vulnerability can assume a 0 value in the CVSS v3 only if the impact score is 0 i.e. if there are no components affected by it. This means, in practice, that the 0 value is never assumed. Indeed, in the NVD there are no vulnerabilities with a score of 0 (there are vulnerabilities with a score of 0 but ranked with CVSS v2). That is a vulnerability will be always accounted for despite its low score.

modifies Porosity, the Common Criteria-based extension modifies Controls while the CVSS-based extension modifies Limitations.

By considering these extensions described, it is possible to derive *extended security indicators* enhancing the security indicators defined in the Standard OSSTMM.

The first indicator which can be extended is the *Actual Security Delta* ($ActSec\Delta$): this indicator accounts for the balance between the Controls put in place, their Limitations and the system exposure. The Actual Security Delta can be used to estimate the impact that a product or solution would cause for the evaluated system and, in the Standard OSSTMM, is computed as

$$ActSec\Delta = FC_{base} - OpSec_{base} - SecLim_{base} \quad (19)$$

where the subscript *base* specifies that the values have been reported on a logarithmic scale. The elements concurring to the computation of this indicator are the Full Controls, the Operational Security and the Security Limitations.

A second indicator that can be extended is *True Protection* ($TruPro$) which allows to characterize the optimal coverage of the system vulnerabilities. Indeed, this indicator provides an insight on the optimal balance between Porosity, (True) Controls and Limitations and, in the Standard OSSTMM, it is computed as

$$TruPro = 100 + TC_{base} - OpSec_{base} - SecLim_{base} \quad (20)$$

A value of 100 corresponds to a perfect balance. Also, in this case, the indicator considers Controls, Operational Security and Limitations.

A third indicator which can be extended is *Actual Security* ($ActSec$) which allows to measure the Operational Security taking in consideration the applied countermeasures and the discovered Limitations (i.e. this indicator considers all the three OSSTMM categories). In the Standard OSSTMM it is defined as

$$ActSec = 100 + ActSec\Delta - \frac{1}{100}(OpSec_{base} \times FC_{base} - OpSec_{base} \times SecLim_{base} + FC_{base} \times SecLim_{base}) \quad (21)$$

where a value of 100 corresponds to a perfect balance between the OSSTMM categories, while values lower than 100 indicate that there are some not addressed security aspects. It should be noted that values higher than 100 are also possible and corresponds to a situation in which there are more Controls than necessary.

The three indicators just described (namely Actual Security Delta, True Protection and Actual Security) can be extended based on the extended versions of Porosity (Section 4.1), Controls (Section 4.2) and Limitations (Section 4.3). In particular, it is possible to define the

- *Modified Actual Security Delta* ($MActSec\Delta$) as

$$MActSec\Delta = AFC_{base} - OpSec_{ba}^{W} - AdjMSecLim_{ba} \quad (22)$$

- *Modified True Protection* ($MTruPro$) as

$$MTruPro = 100 + MTC_{base} - OpSec_{base}^{W} - AdjMSecLim_{base} \quad (23)$$

- *Modified Actual Security* ($MActSec$) as

$$MActSec = 100 + MActSec\Delta - \frac{1}{100}\left(OpSec_{sum}^{W} \times AFC_{base} - OpSec_{base}^{W} \times AdjMSecLim_{base} + AFC_{base} \times AdjMSecLim_{base}\right) \quad (24)$$

4.5. HANDLING THE CYBER AND PHYSICAL DOMAINS

The OSSTMM methodology has been developed for the ICT domain and can be successfully applied to cyber and cyber-physical systems. The Extended OSSTMM described so far inherits this property. However, in the context of CI, it is of paramount importance to extend the applicability of such security metric even to the physical domain.

The resources that are included in the attack surfaces (i.e. those elements characterizing the Operational Security) are typically homogenous

and oriented to the cyber world. Extending this concept to the physical world means to include physical resources in the surface available to the attacker: interfaces become physical entry/exit points and the Damage Potential and Effort can be quantified as well.

So, by analyzing the definition of *controls* and *limitations*, it is clear that they consider a generic process that implies *interactions*, regardless of the cyber or physical nature of this interaction. That is, physical threats and vulnerabilities can be addressed in the same way of cyber vulnerabilities, i.e. by applying the same formulas. However, particular attention should be devoted to the problem of merging in a single number the cyber and the physical information obtained by the computation.

The cyber and physical attack surfaces are in fact computed by considering threats and controls that could affect only the cyber world (e.g., undesired access via internet) or the physical world (e.g., unauthorized physical access to a facility). Control categories defined in the Standard OSSTMM, even if developed for ICT purposes, already include some suggestions on their applicability to the physical domain; hence, the problem of extending the metric to the physical domain translates into the one of correctly classifying the instances of physical controls in the OSSTMM categories.

For what concerns the composition of the attack surfaces coming from the two different domains, empirical results showed that the system security level is not a simple average of the different surfaces. This can be explained with the fact that entry and exit points in a given domain, may introduce additional vulnerabilities in other domains. Following on these considerations, cyber and physical resources are dealt with in a homogeneous way following a careful analysis focused on deriving a mapping of physical resources into the proper OSSTMM categories.

Note that, since the proposed security metric has been developed as a support to CI operators, this metric should allow the CI operators, on the one hand, to have a snapshot of the security level in a holistic way and, on the other hand, to have a mean to realise which domain contributes the most to the system exposure. For this reason, i.e. Actual Security Delta, True Protection and Actual Security indicators are presented to the CI operators in three different versions, i.e. jointly considering the two domains and by distinctly considering each of the two domains.

5. VALIDATION STRATEGY AND EXAMPLE OF USAGE

The security metric proposed in this paper consists of a set of security indicators which, when considered collectively, provide an insight of the system security level. These indicators include both the ones deriving from the Standard OSSTMM, hereinafter referred to as *Standard Security Indicators,* and the ones deriving from the Extended OSSTMM, hereinafter referred to as *Extended Security Indicators*.

The proposed metric has been tested in realistic operational scenarios (e.g. thermo- and hydro-electric power plants) in order to prove, on the one hand, the consistency of the Extended security indicators with the Standard ones and with the security evaluations performed by a certified security evaluator (namely, Andrea Morgagni); on the other hand, the differences between the Standard and Extended security indicators have been in the expected directions, thus providing that the Extended OSSTMM allows to properly capture security features, detailed in Section 4, not covered by the Standard OSSTMM. Given these considerations, the authors understand that such validation strategy, although successfully applied to real-world operation scenarios, is not enough to translate the approach into a standard and that the standardization process itself requires more rigorous tests and long acceptance phase. However, the authors actively interacted with the Institute of Security and Open Methodologies (ISECOM) – i.e. the organization which developed the OSSTMM – community and some of the achieved results are expected to be embedded, in short term, in the Standard OSSTMM methodology.

5.1. AUGMENTED RAV

The security metric proposed in this paper is presented to the CI operators by a spreadsheet referred to *Augmented RAV* (see **Errore. L'origine**

riferimento non è stata trovata.). The Augmented RAV computes the Extended Security Indicators, according to the formulas introduced in this paper, on the basis of the data about the considered CI; these data are the ones to be inserted in the RAV white cells. For a detailed description of the data collection phase, the reader can refer to (Adamsky et al., 2018). It should be noted that the data that must be inserted in the spreadsheets can be obtained in a simple way by just counting the instances of each category and class; as these data are inserted, the Augmented RAV spreadsheet automatically computes the Extended security indicators.

The example reported in **Errore. L'origine riferimento non è stata trovata.** relates to a simplified version of a CI (namely, a hydroelectric power plant) considered during the validation phase. The difference between the Actual Security (i.e. a key Standard security indicator equal to 69.85, and the Modified Actual Security (i.e. the corresponding Extended security indicator) equal to 68.80, is due to the fact that, in the Extended OSSTMM, the Modified Actual Security is able to capture

- the impact of unsecure components lifecycles which derives from the difference between Loss Controls and Actual Controls as defined in equations (4) and (6), respectively and captured in the Augmented RAV by the data inserted in the Loss Controls and Actual Loss columns in white;
- the different contributions of pores to the Operational Security which derives from equation (3);
- the different severity of Limitations which derives from the concepts introduced in Section 4.3 and captured in the Augmented RAV by the data inserted in the Limitations section.

These aspects result in a difference even between the other Standard and Extended security indicators. For example, the difference between the values of the Standard (8.17%) and Extended (4.43%) True Protection indicators – providing a measure of the protection mechanisms put in place with respect to the system exposure and their optimal placement – captures the fact that, due to unsecure value chains of the CI components

Figure 2. Example of Augmented RAV

Extended OSSTMM ATENA metrics

OPSEC		DER	Weights
Visibility	40,0	40,0	1,0
Access	15,0	25,0	1,0
Trust	38,0	38,0	1,0
Total (Porosity)	93,0	103,0	

CONTROLS

Class A	Loss Controls	Actual Loss	Missing	Actual Missing	Weighted Missing	Modified Missing
Authentication	37,00	23,00	56,00	70,00	66,00	80,00
Indemnification	0,00	0,00	93,00	93,00	103,00	103,00
Resilience	5,00	3,20	88,00	89,80	98,00	99,80
Subjugation	0,00	0,00	93,00	93,00	103,00	103,00
Continuity	9,00	4,50	84,00	88,50	94,00	98,50
Total Class A	51,00	30,70	414,00	434,30	464,00	484,30

Class B	Loss Controls	Actual Loss	Missing	Actual Missing	Weighted Missing	Modified Missing
Non-Repudiation	0,00	0,00	93,00	93,00	103,00	103,00
Confidentiality	0,00	0,00	93,00	93,00	103,00	103,00
Privacy	0,00	0,00	93,00	93,00	103,00	103,00
Integrity	0,00	0,00	93,00	93,00	103,00	103,00
Alarm	25,00	14,90	68,00	78,10	78,00	88,10
Total Class B	25,00	14,90	440,00	450,10	490,00	500,10
All Controls	76,00	45,60	854,00	884,40	954,00	984,40
Whole Coverage	8,17%	4,90%				
Missing Coverage			91,83%	95,10%	92,62%	95,57%

LIMITATIONS	Number	CVSS scores
Vulnerabilities	0,00	0,00
Weaknesses	75,00	72,93
Concerns	35,00	34,36
Exposures	8,00	4,44
Anomalies	13,00	13,00
Total # Limitations	131,00	124,73

OpSec	15,74923
Weighted value	16,10320
True Controls	8,30238
Actual value	7,07515
Full Controls	8,30238
Actual value	7,07515
Limitations	23,1224
Adjusted Value	23,1155
True Coverage	8,17%
Modified value	4,43%
True Protection	69,43
Modified value	67,86
Security Δ	-30,57
Modified value	-32,14
ACTUAL SECURITY:	69,85
MODIFIED ACTUAL SECURITY:	68,80

the actual level of protection provided by countermeasures is lower. In other words, the Extended True Protection security indicator is able to capture additional security features not covered by the Standard OSSTMM which, in this case, are a higher attack surface, a lower effectiveness of Controls and a lower impact of vulnerabilities. The same aspect is also captured by the Standard (-30.57) and Extended (-32.14) Security Delta indicator. The difference between the OpSec (15.75) and the Weighted OpSec (16.10) indicators derives from the fact that pores have different exploitability and impact degrees.

The difference between the values of the Standard (23.122) and Extended (23.116) Limitations security indicators captures the fact that, in the given scenario, some Limitations are known, as well as their impact; such a priori information allows to lower their influence on the overall security of the considered CI.

In other words, these results prove that the Extended Security Indicators are able to capture meaningful information on security features not covered by the Standard OSSTMM (e.g. lifecycle); such information can be used by CI operators to improve their understanding of the security environment.

Finally, it is worth noting that the fact that the value of the Modified Actual Security is close to the value of the Standard OSSTMM Actual Security (which is the most comprehensive parameter of the security metric) is a confirmation that the proposed security metric can provide a fine-tuning over a consolidated standard. In this respect, take into account that the indicators outside the gey box (OpSec, Limitations, True Controls, Full Controls, True Protection, Security Delta and Actual Security) are expressed on a logarithmic scale. This means that small differences between the indicators reflect meaningful deviations from a security point of view.

A relevant contribution of the presented framework consists in having proved, in an empirical way, the convergence of different security standards. The Extended OSSTMM Framework, indeed, is able to capture, in a coherent and holistic way, variations of the security level expected from other standards which could not be captured by the Standard OSSTMM.

6. CONCLUSIONS AND FUTURE WORKS

This paper has described an innovative security metric obtained as an extension of the OSSTMM methodology. Two main reasons drove the development of such an extension.

First, the requirement to enrich the description capabilities of the OSSTMM security evaluation procedure in order to capture meaningful security features not covered by the Standard OSSTMM. Such features are the components lifecycle, the different contributions to the system exposure pores and the different severity of vulnerabilities. In this respect, it is worth stressing that the possibility of discriminating between the Limitations associated with the system interactions and operations also allows discriminating among the possible countermeasures to be implemented.

Second, the requirement of extending the applicability of the OSSTMM methodology to the CI domain. This has been accomplished by identifying the counterparts of the OSSTMM Controls classes in the physical domain.

The effectiveness of the description capabilities of the proposed security metric have been tested in realistic operational scenarios and have been validated by a certified security evaluator.

7. ACKNOWLEDGEMENTS

The authors would like to thank the partners involved in the H2020 ATENA project, in particular the researchers working in the CRAT Team and eng. Andrea Morgagni from Leonardo s.p.a., for their precious help in the development of the present and the previous related works (Giuseppi, Tortorelli, Germana, Liberati, & Fiaschetti, 2019; Panfili et al., 2018).

A special acknowledgment goes to Pete Herzog, the creator of the OSSTMM methodology, which provided useful comments that have been taken in consideration in the development of the present work and in the rest of the related activities of the H2020 ATENA project.

Finally, the authors would like to thank the reviewers for their precious comments which helped improving the quality of the present work.

REFERENCES

(ISA), I. S. of A. (n.d.). ISA/IEC 62443 - Industrial Network and System Security. Retrieved from www.isa.org/

Adamsky, F., Aubigny, M., Battisti, F., Carli, M., Cimorelli, F., Cruz, T., … Soua, R. (2018). Integrated protection of industrial control systems from cyber-attacks: the ATENA approach. *International Journal of Critical Infrastructure Protection*, *21*, 72–82. https://doi.org/10.1016/J.IJCIP.2018.04.004

Ahmed, M. S., Al-Shaer, E., Taibah, M., & Khan, L. (2011). Objective risk evaluation for automated security management. *Journal of Network and Systems Management*, *19*(3), 343–366. https://doi.org/10.1007/s10922-

010-9177-6

Alcaraz, C., & Zeadally, S. (2015). Critical infrastructure protection: Requirements and challenges for the 21st century. *International Journal of Critical Infrastructure Protection*, *8*, 53–66. https://doi.org/10.1016/j.ijcip.2014.12.002

Center for Internet Security (CIS). (2020). CIS-CAT.

Di Giorgio, A., Liberati, F., Lanna, A., Pietrabissa, A., & Priscoli, F. D. (2017). Model Predictive Control of Energy Storage Systems for Power Tracking and Shaving in Distribution Grids. *IEEE Transactions on Sustainable Energy*, *8*(2), 496–504. https://doi.org/10.1109/TSTE.2016.2608279

Di Mase, D., Collier, Z. A., Heffner, K., & Linkov, I. (2015). Systems engineering framework for cyber physical security and resilience. *Environment Systems and Decisions*, *35*(2), 291–300. https://doi.org/10.1007/s10669-015-9540-y

FIRST. (2015). Common Vulnerability Scoring System v3.0: Specification Document. *Forum of Incident Response and Security Teams (FIRST)*. https://doi.org/10.1109/msp.2006.145

Frezzetti, A., & Manfredi, S. (2019). Design and experimental testing of an optimization-based flow control algorithm for Energy Harvesting Wireless Sensor Networks. *Control Engineering Practice*, *92*, 104075. https://doi.org/10.1016/j.conengprac.2019.06.014

Gadyatskaya, O., Jhawar, R., Kordy, P., Lounis, K., Mauw, S., & Trujillo-Rasua, R. (2016). Attack trees for practical security assessment: Ranking of attack scenarios with ADTool 2.0. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9826 LNCS, pp. 159–162). Springer Verlag. https://doi.org/10.1007/978-3-319-43425-4_10

Giuseppi, A., Tortorelli, A., Germana, R., Liberati, F., & Fiaschetti, A. (2019). Securing Cyber-Physical Systems: An Optimization Framework based on {OSSTMM} and Genetic Algorithms. In *2019 27th Mediterranean Conference on Control and Automation ({MED})*. IEEE. https://doi.org/10.1109/med.2019.8798506

Herzog, P. (2016). OSSTMM: The Open Source Security Testing Methodology Manual: v3. *Isecom*, 213. Retrieved from http://www.isecom.org/mirror/OSSTMM.3.pdf

ISACA. (2019). COBIT 5 - Control OBjectives for Information and related Technology.

ISO/IEC. (2020). ISO/IEC IS 15408 - Common Criteria.

ISO/IEC 27005:2018, Information Technology - Security Techniques - Information Security Risk Management. (2018).

ISO 31000:2018 - Risk Management. (2018).

Kourtis, M.-A., McGrath, M. J., Gardikis, G., Xilouris, G., Riccobene, V., Papadimitriou, P., … Petrini, A. (2017). T-NOVA: An Open-Source MANO Stack for NFV Infrastructures. *IEEE Transactions on Network and Service Management*, *14*(3), 586–602. https://doi.org/10.1109/tnsm.2017.2733620

Manadhata, P. K., & Wing, J. M. (2011). An attack surface metric. *IEEE Transactions on Software Engineering*, *37*(3), 371–386. https://doi.org/10.1109/TSE.2010.60

Mauw, S., & Oostdijk, M. (2006). Foundations of attack trees. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 3935 LNCS, pp. 186–198). https://doi.org/10.1007/11734727_17

MITRE. (2005). Common vulnerabilities and exposures.

Mo, Y., Kim, T. H. J., Brancik, K., Dickinson, D., Lee, H., Perrig, A., & Sinopoli, B. (2012). Cyber-physical security of a smart grid infrastructure. *Proceedings of the IEEE*, *100*(1), 195–209. https://doi.org/10.1109/JPROC.2011.2161428

Morgagni, A., Fiaschetti, A., Noll, J., Arenaza-Nuño, I., & Del Ser, J. (2017). Security,

privacy, and dependability metrics. In *Measurable and Composable Security, Privacy, and Dependability for Cyberphysical Systems: The SHIELD Methodology* (pp. 159–191). https://doi.org/10.1201/9781138042858

National Institute of Standards and Technologies (NIST). (2008). Special Pubblication, 800-115.

Open Web Application Security Project (OWASP). (2020a). OWASP SAMM v2.0.

Open Web Application Security Project (OWASP). (2020b). OWASP Security Knowledge Framework.

OpenSCAP. (2019). OpenSCAP User Manual.

Panfili, M., Giuseppi, A., Fiaschetti, A., Al-Jibreen, H. B., Pietrabissa, A., & Delli Priscoli, F. (2018). A Game-Theoretical Approach to Cyber-Security of Critical Infrastructures Based on Multi-Agent Reinforcement Learning. In *2018 26th Mediterranean Conference on Control and Automation (MED)* (pp. 460–465). IEEE. https://doi.org/10.1109/MED.2018.8442695

Rehak, D., Senovsky, P., Hromada, M., & Lovecek, T. (2019). Complex approach to assessing resilience of critical infrastructure elements. *International Journal of Critical Infrastructure Protection*, *25*, 125–138. https://doi.org/10.1016/j.ijcip.2019.03.003

Saripalli, P., & Walters, B. (2010). QUIRC: A Quantitative Impact and Risk Assessment Framework for Cloud Security. *Ieeexplore.Ieee.Org*. https://doi.org/10.1109/CLOUD.2010.22

Schneier, B. (2015). Attack Trees. In *Secrets and Lies* (pp. 318–333). https://doi.org/10.1002/9781119183631.ch21

Wells, L. J., Camelio, J. A., Williams, C. B., & White, J. (2014). Cyber-physical security challenges in manufacturing systems. *Manufacturing Letters*, *2*(2), 74–77. https://doi.org/10.1016/j.mfglet.2014.01.005