On Cybersecurity Science and Engineering*

Francesco Beltramini[1] and Marco Rocchetto[1]

[1] *V-Research, Verona, Italy*

Abstract

The objective of this research is to develop of a theory that defines (all and only) the possible insecurity and security configurations of any abstract system. The theory is structured upon other theories that defines how a component of a system can be abstracted into an agent, defining how agents can be formalized (both syntactically and semantically) to describe an abstract system, such as a graph. Some of these theories (e.g. used for the semantic definition of the abstract system) are the epistemological definition of knowledge, the Belief-Desire-Intent and the Assertion-Belief-Fact framework of reference, mereology, and topological structure. We argue that a mereology is the most appropriate abstract underlying structure, do to its generality, for defining the expressiveness of the system abstraction. Furthermore, a mereology allows us to define an ontology rather than a taxonomy. We also correlate different abstractions of the system to the TRL and the engineering V-model.

We implemented a formal theory (of axioms) of a mereotopology, and of the Region Connection Calculus (RCC3 and RCC5) in a Python program that uses the Z3 SMT solver. The results show that a single component (i.e. agent) of an abstract system has a definite number of different insecurity configurations (e.g. 53 using RCC5 over a topological structure) and only 1 secure (i.e. expected) configurations. The configurations are reported as models satisfying the abstract system semantics.

We considered the philosophical definition of truth behind our approach, rejecting "proof" by induction from partial empirical evidences. Our theory can be applied to system engineering and we show a concrete application of our theory to the risk assessment of an ad-hoc system. Finally, we provide a number of ideas to support the engineering of secure systems (e.g. purely cyber or cyber-physical).

1 Introduction

Humanum est errare

Seneca the Elder

The European Commission states in[23] that: “Cybersecurity is one of the priority areas [. . .] of the Commission initiative on ICT Standards, which is part of the Digitising European Industry[24] strategy launched on 19 April 2016. The aim is to identify the essential ICT standards and present measures to accelerate their development in support of digital innovations across the economy”. The same document (i.e.[23]) states that “The EU will invest up to €450 million [. . .], under its research and innovation programme Horizon 2020”. The EU, in 2016 published a press release[25] in which they present a strategy to invest €1.8 *billion* to “increase measures to address cyber threats”. The EU is not the only investor in cybersecurity, most of the developed countries and several companies are investing enormous amount of money towards various aspects of cybersecurity (e.g. The US vulnerability databases[69] maintained by the National Institute of Standards and Technologies, i.e. NIST, of the US Department of Commerce).

The cybersecurity industry is growing fast, e.g. as reported in[21]. For example, in[76], published by the Forbes, is stated that €5.3 billion of funding where poured by venture capitalist into cybersecurity companies in 2018. The Forbes, in the same article, also highlights another peculiar (as seemly contradictory) trend: “[. . .] during the same time period, the number of cybersecurity breaches increased exponentially”. The data reported by the NIST through the official CPE (Common Platform Enumeration) Dictionary Statistics on the NVD websites in [70], show that in 2016 the number of reported vulnerabilities reported where around 6000 while in 2019 the number of vulnerabilities was above 16000. The scientific community also reports similar findings. In fact, in[37], Cormac Herley (Microsoft Research) shows how basic cybersecurity principles (such as the confidentiality benefit over the clear text for passwords typed into forms, e.g. for logins in websites) are not fully understood or shared between the cybersecurity research community[54]. The lack of understanding of basic security principle, the inverse proportionality between investments in cybersecurity and the number of reported vulnerabilities year after year, can be linked to the lack of a foundational theory on cybersecurity, as already highlighted by Cormac Herley in[39].

Another way of looking at the same problem is by analyzing the different definitions of security with respect to the scientific method of enquiry applied to get to the definition itself. In the following we categorize the related work into three categories, as defined by Sextus Empiricus in [27]. To the best of our knowledge, there is no evidence that this categorization isn’t complete. “The natural result of any investigation is that the investigators either discover the object of search or deny that it is discoverable and confess it to be inapprehensible or persist in their search. [. . .] This is probably why”[27]:

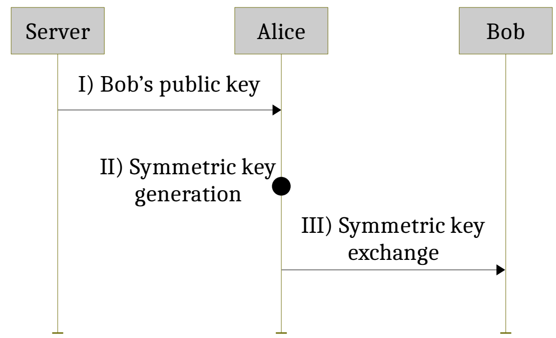


Figure 1: Abstraction of an ad-hoc esemplificative protocol execution

- The *dogmatists* "have claimed to have discovered the truth" on what cybersecurity is
  - Wikipedia defines cybersecurity in [47] as the protection of computer systems and networks from the theft of or damage to their hardware, software, or electronic data, as well as from the disruption or misdirection of the services they provide
- The *academics* "have asserted that it cannot be apprehended"
  - Eugene H. Spafford, Professor at Purdue University, defines cybersecurity as follow. "The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards — and even then I have my doubts." [65]
- The *skeptics* "go on inquiring"
  - Cormac Herley reaches the conclusion that cybersecurity has no definition. "There is an inherent asymmetry in computer security: things can be declared insecure by observation, but not the reverse. There is no test that allows us to declare an arbitrary system or technique secure. This implies that claims of necessary conditions for security are unfalsifiable. " [39]

In this article, we give the first scientific theory (to the best of our knowledge) on security.

Structure. In Section 2 we define and formalize the problem statement. In Section 4 we outline our security theory, and in Section 6 we describe the implementation of the theory and some empirical tests of the theory. Finally, in Section 7 we conclude the paper with an overview of the related work.

2 Problem Statement

In[39], Cormac Herley explores what he calls "an asymmetry in computer security", which he defines as follows: "Things can be declared insecure by observation, but not the reverse. There is no observation that allows us to declare an arbitrary system or technique secure". Herley then uses this argument to show that "claims that any measure is necessary for security are empirically unfalsifiable". Given that, any theory which is not falsifiable by an empirical experiment is well known[1] to be nonscientific (i.e. unfalsifiability is a fallacy of a theory), Herley concludes that there is no scientific theory on cybersecurity; which means that cybersecurity lays in the realm of pseudo-sciences[38]. Herley, e.g. in[36], discusses the implications of a nonscientific approach to cybersecurity, and highlights the tremendous impact on all the scientific research and engineering of systems; leading often to terrorism and wars, and wasting of resources in useless protections or overspending. While the criticism is investigated in[39], no solution is provided. On the contrary, the goal of this work *is* to lay the foundations of a scientific cybersecurity theory. Furthermore, in Section 2.1, we consider the problem raised by Herley not confined to "computer security" but to any abstract system (so that our theory may hold for any sound implementation such as networks, mechanical, cyber, or cyber-physical system, or even a single computer or a single device such as an hard-drive). There is also an apparent inconsistency in[39] that we seek to clarify before following (as we agree) the scientific path draw by Herley: cybersecurity is defined as an abstract property in many formal approaches to the investigation of the security of systems, and the security of the design of a formally verified protocol is indeed falsifiable. For example, in the protocol verification community, security is often defined as a formalization of the high-level properties confidentiality, integrity, and availability. The problem in such approaches is not the definition of what cybersecurity is, but the use of theories (such as the Dolev-Yao attacker model[2][26]) that only applies to specific instances (often called scenarios) and abstraction of the protocol. This, in turn, creates a false sense of security since requires assumptions on the abstraction of the system of which security is verified. As an example, for the formal security verification of the system in Figure 1, a formalized scenario needs to be defined by a modeler who chooses (among others): (i) a scope of the formalization (e.g. excluding the server that distributes the public key is often done when verifying the security of authentication protocols), (ii) the number of sessions (even tough some approaches do reason on an infinite number of sessions such as[29]), (iii) honesty/dishonesty of the peers (e.g. in the ASLan++ language[77]), and (iv) the abstraction of the cryptographic primitives (e.g. ProVerif vs CryptoVerif[5]). Some of the choices will completely change the results of the formal verification

[1] "A theory which is not refutable by any conceivable event is nonscientific. Irrefutability is not a virtue of a theory (as people often think) but a vice." – Karl Popper, Conjectures and Refutations[56]

[2] For the sake of simplicity, the Dolev-Yao attacker can be considered as an abstraction of an active attacker who controls the network but cannot break cryptography.

of the system. For example, under the perfect cryptography assumption[3] and assuming that no violation to any security property is done after message I); in Figure 1, the freedom of choosing the scope determines that the flaws related to the dishonest impersonation of the Server may or may not be considered in the verification process. This choice has tremendous impact on the focus and findings of the verification of the security of the protocol. While this may seem to turn upon minutiae and foreseeable, this highlights the false sense of security that may derive from a non-scientific theory of system security[4].

2.1 Sicurezza: Safety and Security

In most of the natural languages, and in Italian too, the concepts of safety and security are not syntactically differentiated and both terms (safety and security) are expressed by the same word, e.g. sicurezza in Italian. A semantic distinction between safety and security is correlated to a belief[5] that safety deals with *accidents* (i.e. an unfortunate incident) posed by the natural environment (e.g. natural events such as wearing of hardware components) while security deals with *incidents* posed by mankind (e.g. attackers and bugs). The fundamental difference between nature and mankind (and, in turn, between safety and cybersecurity) is believed to be on the different intents[6] (accidents are unfortunate while incidents are not) of the causes that generates the threat; namely, nature is believed not to have malicious intents (but unfortunate causes-effects) while threats generated by mankind are believed to be malicious[7]. An overview on the aforementioned aspects of safety and security is depicted in Figure 2 and is used as a baseline for a definition of the terms that structure our current understanding of safety and security.

- *Mankind* "refers collectively to humans" [49], while the concept of *Nature* is related "to the intrinsic characteristics that plants, animals, and other features of the world develop of their own accord" (e.g. the physical universe)[50].

[3]As defined in[61]: "In the so called perfect cryptography assumption, the security encryption scheme is suppose to be perfect, without any exploitable flaw, and so the only way for the attacker to decrypt a message is by using the proper key. That assumption is widely accepted in the security protocol community, and most of the formal reasoning tools for the analysis of security protocols abstract away the mathematical and implementation details of the encryption scheme[74, 3, 2, 62]"

[4]"To the superficial observer, the analysis of these forms seems to turn upon minutiae. It does in fact deal with minutiae, but they are of the same order as those dealt with in microscopic anatomy." – Karl Marx, Capital Volume 1, 1867

[5]A belief has to be intended as a proposition which is supposed to be true by the majority of humans in our society without a scientific underlying theory but based on partial empirical evidences or inductive proofs.

[6]"The belief–desire–intention software model (BDI) is a software model developed for programming intelligent agents."[45]. In the BDI model, the intents represents the deliberative state of an agent which determines the choice of that agent on what to do.

[7]Of course, logical flaws or bugs may be introduced by other means (e.g. ignorance) without explicit malicious intents, but the exploitation of those flaws is considered (for now, and detailed afterwards in the article) malicious, and then we consider any vulnerability to be malicious (without loss of generality) even if due to the lack of skills.

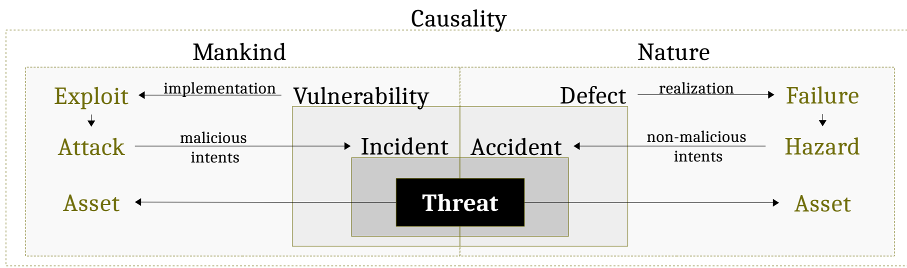


Figure 2: Overview security and safety keywords

  - So far, we have used several terms to refer to an *attacker*, i.e. threat agent or threat source, considering those terms to be semantically equivalent. This "shallowness" raise form the necessity of properly citing the different sources, but, in the reminder of this paper, we consider the Causality principle to be the *threat source*, Nature or Mankind to be the *threat agents* and an *attacker* as a specific malicious threat agent which materialize a threat.

- *Vulnerability*[8], as defined in[53] (and adopted in[4]), is "weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source". On the one hand, the definition is broad to enclose as much causes (that generates a vulnerability) as possible; on the other hand, it derives from empirical evidences (which should be considered beliefs[9] since they are partial results in nature) while a vulnerability should be defined in a way that is empirically falsifiable. This means that the term vulnerability should have a complete and sound definition, so that no other causes (e.g. other sources) but the ones in the definition are responsible for a vulnerability. Furthermore, the term "threat sources" used in the definition in[53] may be identified with both Nature and Mankind, not differentiating between safety and security. In Definition **??**, we provide a formal theory of vulnerability (so that the scientific community can identify tests for the completeness and soundness of the definition itself).

Most of the safety-preserving principles in the field of engineering of safety-critical cyber-physical systems (such as elevators and aircraft), upon which safety requirements (e.g. in standards such as the IEC 61508 or 61511[1]) are

[8]The term vulnerability is not present in the Encyclopedia of Cryptography and Security, while it is used in 12 entries (such as in the definition of "penetration testing"[6]) highlighting how commonly this word is used without a proper supporting semantics

[9]"For this view, that *That Which Is Not* exists, can never predominate. You must debar your thought from this way of search, nor let ordinary experience in its variety force you along this way, (namely, that of allowing) the eye, sightless as it is, and the ear, full of sound, and the tongue, to rule; but (you must) judge by means of the Reason (Logos) the much-contested proof which is expounded by me." – Parmenides of Elea, On Nature (circa 500 B.C.), fragments B7.1–8.2 [35]

defined, have been defined following empirical tests and measurements. While reasoning by induction based on the empirical observation should be avoided, since it may easily lead to false beliefs instead of scientific theories, this approach is often justified by the supposed impossibility of defining a theory that correctly predicts failures which, in turn, pose hazards to a system. To the best of our knowledge, and supported by[39], the correlation between predictability of environment and believed unpredictability of attackers (i.e. a malicious environment) has not been correlated to a theory on cybersecurity. Therefore, inductive research efforts in predicting malicious effects are accepted (and published) in scientific conferences (e.g. [60]). A failure of a wire due to environment (e.g. due to humidity, dust, heat &c) is defined from empirical evidences and processes have been standardized to test qualities of hardware components This process completely breaks down when a malicious environment (i.e. an attacker) is considered instead of the (supposedly honest and predictable) natural environment. Therefore, the same approach that is in use for safety, seems not to be applicable to test security.

Going back to Figure 2, a vulnerability does not necessarily become a threat for the system, unless exploited "through a channel that allows the violation of the security policy [...]"[53] (e.g. a software or procedure) that takes advantage of the vulnerability causing an *attack* to the system, which may result in several correlated incidents and threats. The process of exploitation of a defect as a vulnerability is reported in Figure 2 such that the difference between exploit and failure, and attack and accident is to be found just in the maliciousness of the intents that causes this process (i.e. excluding the intent, the terms are just syntactic transformation from a vulnerability to defect, from accident to incident). In the following, we conclude the informal definition of the terms that we used in this section and in Figure 2.

- *Causality* refers to the causality principle; defined in[22] as "Causality is a genetic connection of phenomena through which one thing (the cause) under certain conditions gives rise to, causes something else (the effect). The essence of causality is the generation and determination of one phenomenon by another. In this respect causality differs from various other kinds of connection, for example, the simple temporal sequence of phenomena, of the regularities of accompanying processes".

- An *Exploit*[10] is "An exploit (from the English verb to exploit, meaning to use something to one's own advantage) is a piece of software, a chunk of data, or a sequence of commands that takes advantage of a bug or vulnerability to cause unintended or unanticipated behavior to occur on computer software, hardware, or something electronic (usually computerized)."[48].

- An *Attack*, as defined by the International Standard ISO/IEC 27000 is an "attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of an asset"; where an *Asset* is "anything

[10]We note that the term exploit is only used as a verb in[68]

that has value to the organization". We note that for the purpose of this article, we do not want to focus on a specific organization or business to define asset but, in general, on any abstract organization (e.g. a company or a society). We do not consider ethical hackers as attacking a system. In fact, we consider the term *hack* as non-malicious (see Hacker[66]).

- A *Threat*, as defined in[53], is "Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service".
- *Defect*, "anything that renders the product not reasonably safe"[59] (i.e. a characteristic of an object which hinders its proper usability).
- *Failure*, as defined in[43] as "a state of inability to perform a normal function". The term is structured and detailed in [53, 28] but relying on an abstract notion of failure without a specific definition.
- *Hazard*, "a potential source of harm"[28].

3 Glossary – A Formalization

We now define a formalizations of the concepts described in Section 2 and depicted in Figure 2. We base our formalization on first order logic (FOL) with the standard truth-value semantics. The choice of this logic is required by the semantics of the concepts formalized afterwards in this section.

We consider a first-order language $\mathcal{L}$ over a signature Σ_P where $P, P', \ldots, P^n$ represent terms, and $\varphi, \varphi', \ldots, \varphi^n$ represent formulas. The syntax is defined as follows.

$$\varphi := P \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \Rightarrow \varphi' \mid \forall P.\ \varphi \mid \exists P.\ \varphi$$

where $\wedge$, $\neg$, $\vee$, and $\Rightarrow$ are connectives representing conjunction, negation, disjunction, and (material) implication respectively; while $\forall$ and $\exists$ represent the standard universal and existential quantifiers (resp.). Finally, the symbol "." is just syntactic sugar. We consider $\sigma \subset \Phi \times \{\top, \bot\}$ as the interpretation function, where Φ is any collection of sentences in $\mathcal{L}$ and $\top$ and $\bot$ represent the concepts of "Tautology/True" and "Contradiction/False" respectively.

Following Figure 2, we start by formalizing the outermost term: Causality.

3.0.1 Causality Principle

We formalize the Causality Principle starting from a K modal logic[31] (i.e. without restrictions on the causality relation between worlds). The standard definition of K modal logic is given in Definition 3.1 in terms of an interpretation function (which we named σ with a slight abuse of notation) defined in Definition 3.2.

Definition 3.1. K Modal Logic – A K-frame is a frame $\mathbf{K} = \langle G, R \rangle$ in the K modal logic where R is the binary relation (i.e. a set of ordered pairs) between possible worlds $R \subseteq G \times G$, where G represents the possible worlds, and $G \neq \emptyset$. An actual world $\omega* \in G$ is assumed. For any proposition P, an interpretation function $\sigma(\omega, P)$ returns the truth value of P; e.g. $\sigma(\omega, P) = \top$ means that P holds in ω. A model is defined as the tuple $M = \langle G, R, \sigma \rangle$.

Definition 3.2. Causality as K Modal Interpretation Function – Causality is (recursively) defined as the modal interpretation function σ, as follows.

($\sigma 0$) if $\sigma(\omega, P) = \top$ then $\omega \models P$
($\sigma 1$) $\omega \models \neg P$ iff $\omega \not\models P$
($\sigma 2$) $\omega \models P \wedge Q$ iff $\omega \models P$ and $\omega \models Q$
($\sigma 3$) $\omega \models \Box P$ iff for any world $\omega' \in G$ if $\omega R \omega'$ then $\omega' \models P$
($\sigma 4$) $\omega \models \Diamond P$ iff there exists a set of worlds $\Omega' \subset G$ such that for any $\omega' \in \Omega'$, if $\omega R \omega'$ then $\omega' \models P$
($\sigma 5$) $\models P$ iff $\omega* \models P$

where truth is defined as necessary with $\Box$ and possible with $\Diamond$.

The causality principle has been defined in its generic form. In fact, the accessibility relation R is free from any axiomatic restriction (e.g. it's not reflexive nor anti-reflexive). We will focus in Section 5 (for our tests) on the application of our theory to CPS system engineering. More detailed case studies (i.e. **tofix:** a CPS as a smart power-grid) will be defined in Section 6, where the accessibility relation defines in details how the system itself evolves, due to causal relation (i.e. by restricting the causality principle only to those cause-effect that defines the system). Therefore, the definition of R will be specialized in a more strict way based on the application domains and case study.

We note that we have defined the Causality principle without considering it a *threat source* since we lack the concept of intent and maliciousness. Similarly, in the next sections we won't discriminate between Nature and Mankind until we introduce the concepts of maliciousness in Section 4 and then formally define a threat agent in the same section.

3.0.2 Agents: Mankind and Nature

Mankind and Nature are defined in Section 2 as two abstract agents, both as collections (i.e. an abstract type that does not imply a specific implementation) of their sub-agents (i.e. humans for Mankind and plants, animals, &c. for Nature). Similarly to[63], we define Mankind and Nature, and any other agent in the reminder of this article, as a meronomy (an hierarchy of Part-Whole relations) based on a standard definition of mereology, i.e. based on the definition of Parthood relation between *Parts*. However, shall consider different types of Part; so, we extend the mereology to a mereo-topology[64, 75, 57], to increase the number of different types considered and to generalize the relations between Parts (as in Table 1). For the sake of readability, we use the term *Region* both to refer to a mereological Part and to a topological Region. The

Table 1: RCC3, RCC5, and RCC8 relations between Regions X, Y and Z

RCC3	RCC5	RCC8	Name	Notation	Definition
			Connects with	$C(X,Y)$	$X \subseteq Y$
			Disconnected from	$\neg C(X,Y)$	$X \nsubseteq Y$
			Part of	$P(X,Y)$	$\forall Z\ C(Z,X) \rightarrow C(Z,Y)$
			Overlaps	$O(X,Y)$	$\exists Z\ P(Z,X) \wedge P(Z,Y)$
●			**Overlaps Not Equal**	$ONE(X,Y)$	$O(X,Y) \wedge \neg EQ(X,Y)$
●	●	●	**Equal to**	$EQ(X,Y)$	$P(X,Y) \wedge P(Y,X)$
●	●	●	**DiscRete from**	$DR(X,Y)$	$\neg O(X,Y)$
	●	●	**Partial-Overlap**	$PO(X,Y)$	$O(X,Y) \wedge \neg P(X,Y) \wedge \neg P(Y,X)$
	●		**Proper-Part-of**	$PP(X,Y)$	$P(X,Y) \wedge \neg P(Y,X)$
	●		**Proper-Part-of-*i*nverse**	$PPi(X,Y)$	$P(Y,X) \wedge \neg P(X,Y)$
		●	**Externally Connected**	$EC(X,Y)$	$C(X,Y) \wedge \neg O(X,Y)$
		●	**Tangential PP**	$TPP(X,Y)$	$PP(X,Y) \wedge \exists Z\ [EC(Z,X), EC(Z,Y)]$
		●	**Tangential PPi**	$TPPi(X,Y)$	$TPP(Y,X)$
		●	**Non-Tangential PP**	$NTPP(X,Y)$	$PP(X,Y) \wedge \neg\exists Z\ [EC(Z,X), EC(Z,Y)]$
		●	**Non-Tangential PPi**	$NTPPi(X,Y)$	$NTPP(Y,X)$

choice of mereotopology is also correlated to the objective of defining a formal ontology, which we use to define the (formal) semantics of the terms (Parts) in Section 2, and of the concepts of safety and security (whole). We aim at creating a meronomy instead of the taxonomies such as the one provided in[69, 58] or instead of the poorly justified CVSS[52] scoring system.

A mereotopology, as defined e.g. in[57], is an ordered mathematical structure where the basic relation between Regions is the reflexive and symmetric**FIX**[11] *Parthood* relation $\subseteq$.

Definition 3.3. Parthood – Given any pair of mereotopological Regions X and Y,

1. Reflexivity: $\forall X. \ (X \subseteq X)$
2. Symmetry: $\forall X, Y. \ (X \subseteq Y \Rightarrow Y \subseteq X)$

As later defined in Definition 3.5, the Parthood relation orders a universe of agents $\mathcal{A}$ by defining the so called *Connects with* (see in Table 1) relation between Regions. We want this universe $\mathcal{A}$ to be expressible in FOL. In this way, we can reason both on the constituent of security (i.e. its terms and agents defining a system where security needs to be considered), and on the evolution of those constituent w.r.t. cause-effects relations according to the modal structure of causality we defined in Section 3.0.1. This will allow us (in Section 5 and Section 6) a better**FIX**[12] positioning w.r.t. risk assessment technologies (which most often reason on the constituent of a system design), and protocol verification tools (which requires some formalization of a cause-effect relation, e.g. in Linear Temporal Logic).

As we argued in Section 2, we must correlate the definition of agent (i.e. Mankind and Nature) in Definition **??** to the mathematical structure of the

[11] **mr:** I guess it must be monotonic as defined in[57] but I don't find it consistently in other papers.

[12] **mr:** Can we prove this by construction?

logic that defines them. We express Mankind and Nature as formulas over the theory of mereology and then in terms of mereotopological Regions, extending the interpretation function σ, to include a formal theory of mereotopology. We use the Region Connection Calculus (RCC), as defined in [51, 33], to provide an axiomatization of the spatial concepts and relations in first-order logic to correlate the algebraic structure to mereology. In its broader definition, the RCC theory is composed by eight axioms, and is known as RCC8. In the text, for brevity, we will often focus only on RCC5 (without loss of generality) by not considering tangential connections between spatial Regions. We discuss the choice of RCC5 in more detail in Appendix **??**. In Table 1, we summarize the axioms of the Region Connection Calculus (see, e.g., [34]).

Definition 3.4. RCC axiomatization – For any X, Y pair of Regions in a mereotopology:

(σ6) $\sigma(X \subseteq Y)$ iff $[\sigma(X \subseteq X) = \top$, and $\sigma(X \subseteq Y) = \bot$ or $\sigma(Y \subseteq X) = \top]$ **FIX**[13]
(σ7) $\omega \models C(X, Y)$ iff $\omega \models X \subseteq Y$
(σ8) $\omega \models \neg C(X, Y)$ iff $\omega \not\models X \subseteq Y$
(σ9) $\omega \models P(X, Y)$ iff $\omega \models \forall Z.\ C(Z, X) \Rightarrow C(Z, Y)$
(σ10) $\omega \models O(X, Y)$ iff $\omega \models \exists Z.\ P(Z, X) \wedge P(Z, Y)$
(σ11) $\omega \models EQ(X, Y)$ iff $\omega \models P(X, Y) \wedge P(Y, X)$
(σ12) $\omega \models DR(X, Y)$ iff $\omega \models \neg O(X, Y)$
(σ13) $\omega \models PO(X, Y)$ iff $\omega \models O(X, Y) \wedge \neg P(X, Y) \wedge \neg P(Y, X)$
(σ14) $\omega \models PP(X, Y)$ iff $\omega \models P(X, Y) \wedge \neg P(Y, X)$
(σ15) $\omega \models PPi(X, Y)$ iff $\omega \models P(Y, X) \wedge \neg P(X, Y)$

where Z is a mereotopological Region.

Definition 3.5. Agent: Mankind or Nature – An agent $a \in \mathcal{A}$ is a tuple $\langle rcc(\chi', \chi''), \ldots, rcc(\chi^{n-1}, \chi^n)\rangle$ of RCC relations rcc over mereotopological Regions $\chi', \ldots, \chi^n \subseteq \mathrm{X}$.

Currently, we do not distinguish between Mankind and Nature (since we still lack of the definition of "malicious intent", which is defined in Section 4) and we have defined them as two generic agents.

As depicted in Figure 2, Causality, Mankind, and Nature have a dashed border representing their correlation in terms of cause-effect and then in terms of formal structure which defines them: Modal Logic. Vulnerability, Defect, Incident, Accident, and Threat, similarly, are correlated (depicted as a solid border) in terms of underlying formal structure: Mereotopology.

3.0.3 Regions: Vulnerability and Defect (and Weakness)

As defined in Section 2, a Vulnerability or a Defect is a *Weakness*. As an example, a categorization of weaknesses is given in[20] with 808 weaknesses categorized as "Research Concepts", distributed as follows:

[13] **mr:** would it be better/clearer or just correct to write $\sigma(X \subseteq Y)$ iff $\sigma(X \subseteq X \wedge [X \subseteq Y \vee Y \subseteq X]) = \top$

- Incorrect Calculation - (682)
- Incorrect Access of Indexable Resource (“Range Error”) - (118)
- Use of Insufficiently Random Values - (330)
- Improper Interaction Between Multiple Correctly-Behaving Entities - (435)
- Improper Control of a Resource Through its Lifetime - (664)
- Insufficient Control Flow Management - (691)
- Protection Mechanism Failure - (693)
- Incorrect Comparison - (697)
- Improper Check or Handling of Exceptional Conditions - (703)
- Improper Enforcement of Message or Data Structure - (707)
- Improper Adherence to Coding Standards - (710)

The definition given by the MITRE in[30] of weakness is: “Software weaknesses are errors that can lead to software vulnerabilities. A software vulnerability, such as those enumerated on the Common Vulnerabilities and Exposures (CVE) List, is a mistake in software that can be directly used by a hacker to gain access to a system or network”. The definition is circular if we interpret the word “error” and “mistake” with the same semantics: a weakness is an error that leads to a vulnerability and a vulnerability is a mistake which, in turn, is a weakness. The only difference (between weakness and vulnerability) seems to be that one can consider weakness as a ground term and state that a vulnerability is caused by a weakness, i.e. $\Omega, W \models \Diamond V \wedge W$ where W, V are Regions of Weaknesses and Vulnerabilities (resp.); accepting the hierarchy in the CWE[12] as ground truth. Similarly, we consider the CVE[13] (a database of Vulnerability) or the CVE reported in the NVD as a ground truth.

Definition 3.6. Region: Weakness and Vulnerability – A Region $\chi \subseteq a$ of an agent $a \in \mathcal{A}$, is defined as Weakness W (i.e. representing a weakness introduced by the agent into any phase of production, e.g. of the secure process development life-cycle, of any system or subsystem) iff there exists $\omega \in G$ such that $\omega, W \models \Diamond \exists W', V'.\ rcc(W', V') \wedge \neg DR(W', V')$, where $W' \subseteq W$, and $V' \subseteq V$ is a Region of an agent $a' \in \mathcal{A}$.

Example 1. *CWE-116: Improper Encoding or Escaping of Output[42] –*

- *Description: The software prepares a structured message for communication with another component, but encoding or escaping of the data is either missing or done incorrectly. As a result, the intended structure of the message is not preserved.*
- *Example: This code displays an email address that was submitted as part of a form. Example language JSP.*

  ```
  <% String email = request.getParameter("email"); %>
  ...
  Email Address: <%= email %>
  ```

  The value read from the form parameter is reflected back to the client browser without having been encoded prior to output, allowing various XSS attacks (CWE-79).

- *Observed Examples*
  - *CVE-2008-4636[18]: OS command injection in backup software using shell metacharacters in a filename; correct behavior would require that this filename could not be changed.*
  - *CVE-2008-0769[16]: Web application does not set the charset when sending a page to a browser, allowing for XSS exploitation when a browser chooses an unexpected encoding.*
  - *CVE-2008-0005[14]: Program does not set the charset when sending a page to a browser, allowing for XSS exploitation when a browser chooses an unexpected encoding.*
  - *CVE-2008-5573[19]: SQL injection via password parameter; a strong password might contain &*
  - *CVE-2008-3773[17]: Cross-site scripting in chat application via a message subject, which normally might contain & and other XSS-related characters.*
  - *CVE-2008-0757[15]: Cross-site scripting in chat application via a message, which normally might be allowed to contain arbitrary content.*

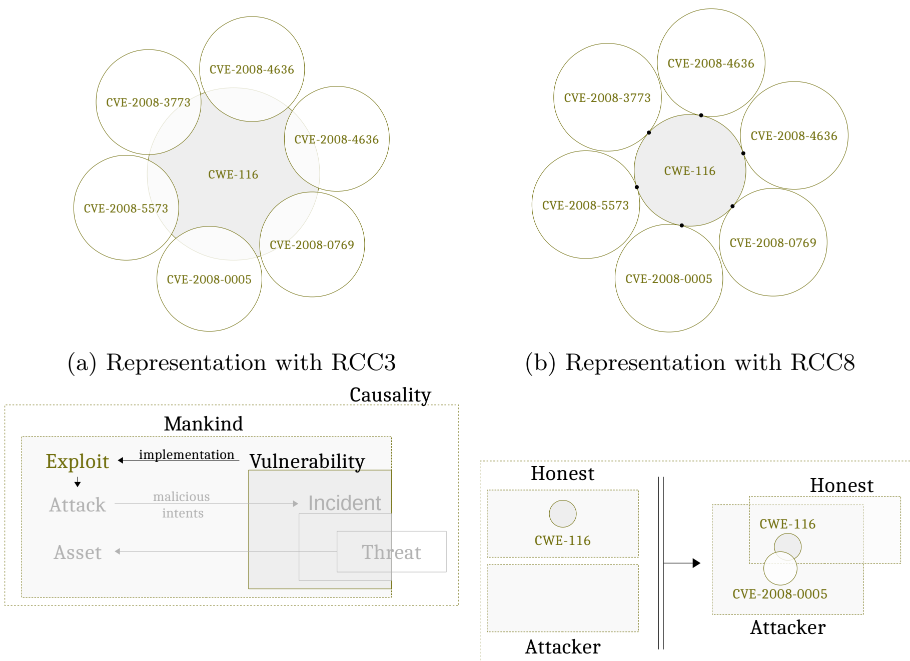
Definition 3.6 states that CWE-116 is a weakness iff there exists a world in K Modal Logic, representing the system in which this weakness exists, such that the natural evolution of this system (i.e. formalized by the causality principle) make it possible to reach another state of the system (i.e. another, accessible world) where there exist a vulnerability that can be implemented from CWE-116 (and this relation is not DR). The CWE website proposes the connection between CWE-116 and, for example, CVE-2008-5573; a vulnerability of the login sub-system of the Poll Pro v2.0[72] system. The formal relation between the two is given as a link (i.e. URI) between the CWE and the CVE, the description of the relation is not defined but it is supposed to be inferred from the descriptions of the CWE-CVE. We can formally represent this link as the ONE connection in RCC3, depicted in Figure 3a, or EC connection in RCC8 (Figure 3b).

FIX[14]

It is interesting to note that the CWE-CVE relation expresses the correlation between weaknesses and vulnerabilities in the most simple form, such that we can formalize all the relation as ONE in RCC3 or EC in RCC8. To express a more complex relation between the two we shall analyze the definition of CWE-116. This is related to the Weakness-Vulnerability-Incident process (i.e. to the details on the implementation/realization in Figure 2) that we analyze in the next section, Section 3.0.4.

Definition 3.7. Region: Vulnerability and Defect – A region $\chi \subseteq a$ of an agent $a \in \mathcal{A}$ is called *Vulnerability* if the agent a is referred to as Mankind, *Defect* if the agent is referred to as Nature.

[14]**mr:** shouldn't we consider the agent who introduces the weakness as dishonest? don't we say this in part 1?



(a) Representation with RCC3

(b) Representation with RCC8

(c) Representation in K-Modal Logic

Figure 3: Relations between CWE-116 and correlated CVEs

3.0.4 Process: Incidents and Accidents

In our informal definition depicted in Figure 2, an Incident is generated through a process caused by a Vulnerability (which, in turn, is caused by a Weakness). Symmetrically, Nature has a process from Defect to Accidents. We start by analyzing the process generated by Mankind, which is divided into the following phases (as shown in Figure 5):

1. An agent (Mankind or Nature) defines a system, e.g. by following an engineering process for creating a new system, or for redesigning or improving an existing system (e.g. a component or a collection of components)

2. The dogmatic definition of a system contains a Weakness (i.e. the agent, Mankind or Nature, introduces an unintended behavior), for example as an un-intended behavior or as an architectural fallacy. The Weakness can be applied to the abstract definition of the system.

3. The Weakness can be defined as a malicious process for a system, called Vulnerability (or Defect for Nature)

4. The process expressed by the Vulnerability can be made operational for a specific implementation of the system (i.e. implemented by Mankind or realized by Nature)

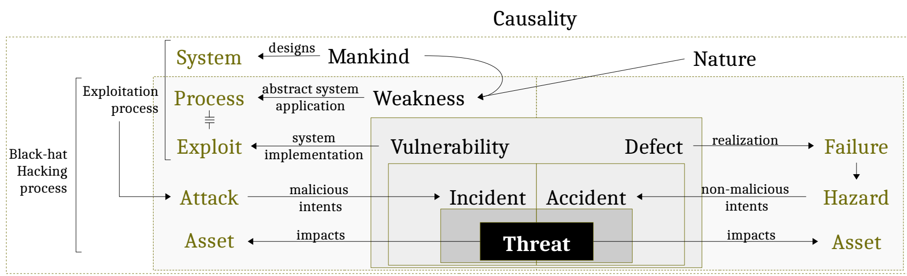


Figure 4: Updated overview security and safety keywords, where Process has to be intended as "Abstract Exploitation Process"

Those additional details are depicted in Figure 4, an update of Figure 2.

Definition 3.8. Exploitation Process – An *Exploit* is an *implementation* of a Vulnerability, where a Vulnerability describes how to transform a Weakness into an abstract malicious process, which can be applied[15] to one or many systems. So, the act of an agent (Mankind) of "exploiting a Vulnerability" is the process of making the Vulnerability operational, by implementing the Vulnerability w.r.t. a specific target system. Whenever the agent is Nature, the implementation is more broadly considered a realization[16].

The Common Attack Pattern Enumeration and Classification[11] (CAPEC) online service provides a "community resource" which aims at "identifying and understanding attacks". The MITRE writes on the CAPEC homepage that "Understanding how the adversary operates is essential to effective cyber security. CAPEC helps by providing a comprehensive dictionary of known patterns of attack employed by adversaries to exploit known weaknesses in cyber-enabled capabilities. It can be used by analysts, developers, testers, and educators to advance community understanding and enhance defenses.". We can extrapolate two major concepts:

- While understanding the exploitation process is the aim of the CWE and CVE initiatives, understanding the hacking process is the aim of the CAPEC initiative.
- CAPEC provides a *dictionary* of *known* attack patterns. As stated in details in Section 2, this dictionary contains empirical evidences and should not be used to induce a cybersecurity theory; but it can be used to validate a theory.

Example 2. *The CWE-116 doesn't just relate the Weakness to a number of Vulnerability in the CVE but also relates the Weakness to a number (4 for the CWE-116) of abstract attack path in the CAPEC:*

[15] Meaning that the malicious process, i.e. any structured procedure such as a protocol, can be somehow made operational, e.g. implemented

[16] "The state of being realized"[44]

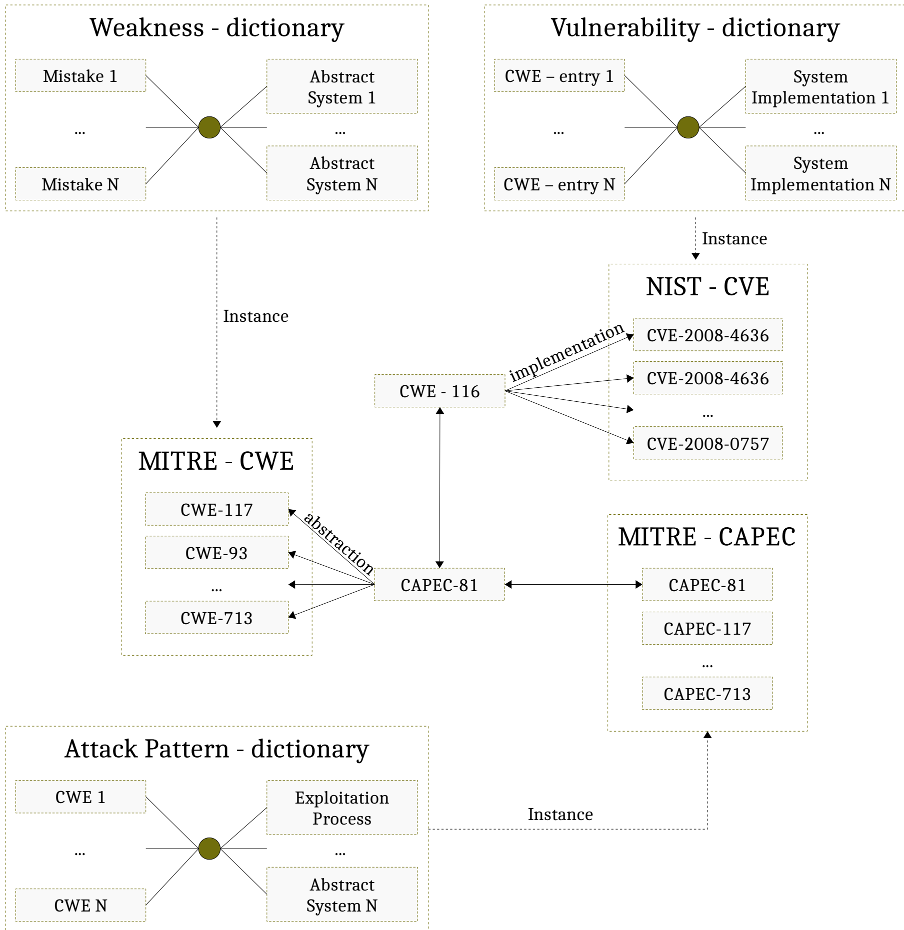


Figure 5: Cybersecurity online dictionaries and connections

- *CAPEC-104[7] Cross Zone Scripting*
- *CAPEC-73[8] User-Controlled Filename*
- *CAPEC-81[9] Web Logs Tampering*
- *CAPEC-85[10] AJAX Fingerprinting*

Each CAPEC entry has a reference to one or many CWE, including CWE-116.

After the exploitation process, as depicted in Figure 5, another process starts and transforms a Vulnerability in an Incident by Attacking the system with a malicious Hack, as follows.

5. An agent (Mankind[17]) the exploitation process can be used to hack a system. We use the term hack (as hacking) to stress that nothing prevents this process from being honest, however, for the sake of simplicity, we focus on Attacks which are malicious by definition.

6. An Attack is applied to a system with malicious intent by an agent

7. The application of the attack results in an Incident which impacts, i.e. poses a Threat to, an Asset

Definition 3.9. Black-hat Hacking Process An *Attack* is the act of making operational an *Exploitation Process* causing an Incident.

Example 3. *CAPEC-81[9] Web Logs Tampering –*

- *Description: An attacker is able to cause a victim to load content into their web-browser that bypasses security zone controls and gain access to increased privileges to execute scripting code or other web objects such as unsigned ActiveX controls or applets. This is a privilege elevation attack targeted at zone-based web-browser security. In a zone-based model, pages belong to one of a set of zones corresponding to the level of privilege assigned to that page. Pages in an untrusted zone would have a lesser level of access to the system and/or be restricted in the types of executable content it was allowed to invoke. In a cross-zone scripting attack, a page that should be assigned to a less privileged zone is granted the privileges of a more trusted zone. This can be accomplished by exploiting bugs in the browser, exploiting incorrect configuration in the zone controls, through a cross-site scripting attack that causes the attackers' content to be treated as coming from a more trusted page, or by leveraging some piece of system functionality that is accessible from both the trusted and less trusted zone. This attack differs from "Restful Privilege Escalation" in that the latter correlates to the inadequate securing of RESTful access methods (such as HTTP DELETE) on the server, while cross-zone scripting attacks the concept of security zones as implemented by a browser.*

- *Execution Flow:*

  1. *Explore: Find systems susceptible to the attack: Find systems that contain functionality that is accessed from both the internet zone and the local zone. There needs to be a way to supply input to that functionality from the internet zone and that original input needs to be used later on a page from a local zone.*

  2. *Experiment: Find the insertion point for the payload: The attacker first needs to find some system functionality or possibly another weakness in the system (e.g. susceptibility to cross site scripting) that would provide the attacker with a mechanism to deliver the payload*

[17] For the sake of readability we focus only on Mankind as an agent

(i.e. the code to be executed) to the user. The location from which this code is executed in the user's browser needs to be within the local machine zone.

3. *Exploit: Craft and inject the payload: Develop the payload to be executed in the higher privileged zone in the user's browser. Inject the payload and attempt to lure the victim (if possible) into executing the functionality which unleashes the payload.*

- *Prerequisite: The target must be using a zone-aware browser.*
- *Consequences:*
  - *Integrity: Modify Data*
  - *Confidentiality: Read Data*
  - *Confidentiality, Access Control, Authorization: Gain Privileges*
  - *Confidentiality, Integrity, Availability: Execute Unauthorized Commands*

The formalization of the Exploitation and Black-hat Hacking processes is the formalization of two foundational cybersecurity concepts which lay the basis for the definition (and formalization) of the cybersecurity theory in the next Section (Section 4). However, the formal definition of those processes shall (i) correlate the formal definition of agents (i.e. the Definition 3.5) with the Causality principle (i.e. the formalization of the Kripke structure in Definition 3.1), and, most importantly, (ii) formally define a system as a structure of agents. For the sake of clarity, we then postpone the formalization of the two processes to the next Section.

4 $\mathcal{ABF}$ – A Security Theory

In order to address the problem raised by Herley, we shall define how to distinguish between a secure and an insecure system. While most of the literature (and this paper so far) have correlated the problem of in-security to the maliciousness of agents interacting with the system, we show that security doesn't seem to have a malicious nature but, rather, insecurity raises from the lack of well-defined security requirements for the design process of a system. We argue that the high number of security vulnerability reported today are simply the realization of potential system configurations, which deviate from the nominal behavior just because the intended behavior of the system is not precisely defined in the specification of the design process.

Given that we apply our theory to *systems engineering* and that the two foundational processes (exploitation and hacking) are based on the concept of a system, we shall begin by defining what a system is, in its abstract form.

4.1 Abstract System State

The ISO/IEC/IEEE 15288:2015 (System Life Cycle Processes) provides a definition of system as "A combination of interacting elements organized to achieve one or more stated purposes."[67]. Therefore, a system can be considered as a single agent where its interacting elements are the constituents of the agent itself. For the sake of simplicity we first define a system as an agent and then extend the definition to a "combination of interacting" agents. We use the same process of [63] but over slightly different terms. The differences and commonalities with [63] are described in Appendix **??**.

There is no agreement between the research communities (e.g. Multi-Agent-System, Epistemic Logic) on which are the constituent of an agent as a system. However, the same ideas revolved around for thousands of years. Some examples follows.

- In [40], Hintikka describes the difference between Knowledge and Belief (as epistemological concepts), and the whole Doxastic logic defines in details how Beliefs can be formalized.
- In [41], Hintikka describes the concept of Information and the difference with Knowledge and Belief.
- In [27], the author states: "The logical criterion also may be used in three senses – of the agent, or the instrument, or the "according to what"; the agent, for instance, may be a man, the instrument either sense perception or intelligence, and the "according to what" the application of the impression "according to" which the man proceeds to judge by means of one of the aforesaid instruments."
- In [63], the authors defines an agent as a tuple of Assertions, Beliefs, and Facts.

If we consider [27], the author applies a "logical criterion" to an agent, for instance a man, for instance a device in a system (or a system) as in [63]. The instrument identified is sense perception, because a man is considered as example, but we can abstract the concept of instrument as any received information, where information is considered as defined in [41]; which can be summarized by the following axiom: "Information is specified by specifying which alternatives concerning the reality it admits and which alternatives excludes". This means, for our argument, that considering a propositional variable, which admits the two alternatives True/False, its information is defined as Believed to be True/False and not Believed to be the opposite. Finally, in [27] the author states that the logical criterium can be judged according to a frame of reference. For our argument, as in [63], we define this concept as Knowledge (or Facts); where knowledge is defined, as in [71], as a set of proposition known by an agent, such that: (i) knowledge requires belief, (ii) knowledge require truth, (iii) knowledge must be *properly justified*.

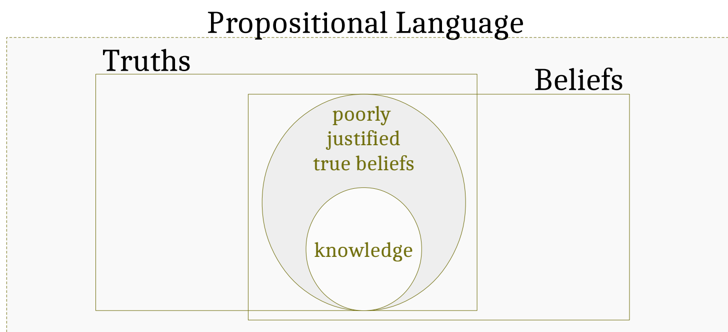


Figure 6: Informal representation of Knowledge and Belief

Before detailing what "properly justified" means, we discuss the mentioned relation between agent and belief and we introduce a more formal definition of knowledge and belief. For our argument, the only objective of Information is to exchange beliefs between agents. Due to the definition of Information and then with its relation to the probabilistic correlation to truth/reality, we consider information in relation with an agent's beliefs. Similarly, we consider beliefs to define the actual behavior of an agent or a system. On the other hand, Knowledge drives the nominal behavior of an agent or a system. This view is not considered in most (if not all) the approaches to protocol verification and the Dolev-Yao theory is usually applied to a representation/abstraction of a protocol that only considers knowledge and transfer of knowledge (i.e. if an agent sends a message, the recipient knows that message). However, the security of a system is tightly related to the difference between the nominal behavior and the implemented one.

Definition 4.1. System State and State-Space – The state of a system (or a sub-system) s is identified with an instance of an agent state (see Definition 3.5) such that $s = \langle rcc(\mathbb{K}, \mathbb{B}), rcc(\mathbb{K}, \mathbb{I}), rcc(\mathbb{B}, \mathbb{I})\rangle$, where $\mathbb{K}, \mathbb{B}, \mathbb{I}$ are Regions of Knowledge, Beliefs, and Information respectively, and rcc is a specific relation between the Regions it predicates on. The state-space of a system is then defined by the different rcc relations between the three pairs of Regions defining the system and all the sub-systems.

A system state space can be seen as a superposition of multiple agent states where knowledge, beliefs, and information are related between each other (in a mereotopological space) through the all the admissible RCC relations. A *system state* (as an instance of the system state space) is a specific configuration of the rcc relations between the three regions. Therefore, a collection of system states defines a collection of choices for those rcc relations.

The difference between Knowledge and Belief is depicted in Figure 8 (see [46]). However, according to[73], Knowledge[18] as an epistemological concept is

[18] "*Theaetetus*: [...] He said that knowledge was true opinion accompanied by reason, but

difficult to formally define. Similarly for the concept of information, Hintikka in [41] states that "A purely logical definition of information is impossible". In this work, however, we are not interested in how knowledge, information, or belief can be precisely formalized from an epistemic standpoint. We assume that a semantic of a correct definition of epistemic knowledge exists, for example the one given in[40] by Hintikka, and we then define knowledge in terms of the Kripke structure defined in Definition 3.1; similarly for Belief.

Definition 4.2. Knowledge – Given an abstract collection of Agents Ag, and the modal operator K_a (where $a \in Ag$), Knowledge is defined as a region of predicates known by an agent $\mathbb{K}_a = \bigcup_\Phi K_a\varphi$ (where Φ is the collection of all the propositions known by a). Given a proposition P, we extend the semantics of the Causality structure with:

(σ16) $\omega \models K_a P$ iff $\omega' \models P$ for all ω' such that $\omega R \omega'$.

Definition 4.3. Belief – Given an abstract collection of Agents Ag, and the modal operator B_a (where $a \in Ag$), Belief is defined as a region of predicates believed by an agent $\mathbb{B}_a = \bigcup_\Phi B_a\varphi$ (where Φ is the collection of all the propositions believed by a). Given a proposition P, we extend the semantics of the Causality structure with:

(σ17) $\omega \models B_a P$ iff $\omega \models \neg K_a \neg P$ (i.e. the agent a considers P possible) and $\omega' \models P$ for all ω' such that $\omega R \omega'$.

Definition 4.4. Information – Given an abstract collection of Agents Ag, and the modal operator I_a (where $a \in Ag$), Information is defined as a region of beliefs asserted by an agents a. $\mathbb{I}_a = \bigcup_\Phi I_a\varphi$ (where Φ is the region of all the propositions believed and asserted by a). Given a proposition P, we extend the semantics of the Causality structure with:

(σ18) $\omega \models I_a P$ iff $\omega \models \mathbb{B}_a P$ (i.e. the agent a considers P possible), $\omega \models \neg\mathbb{B}_a \neg P$

4.2 Engineering Operational System States

When dealing with a specification of a system (i.e. the design if the system for a specific operation (as a generic operating/operational system), considering abstract and general concepts such as knowledge, belief, or information may result to be too abstract for the objectives of the overall engineering process (e.g. verification and validation). As an example, we are usually not interested in information in abstract but to the transfer of that information, which we call Assertions, made by a subset of agents. Therefore, we now define how we consider an agent for the engineering of CPS.

that unreasoning true opinion was outside of the sphere of knowledge; and matters of which there is not a rational explanation are unknowable – yes, that is what he called them – and those of which there is are knowable. [...] *Socrates*: [...] the primary elements of which we and all else are composed admit of no rational explanation; for each alone by itself can only be named, and no qualification can be added, neither that it is nor that it is not, for that would at once be adding to it existence or non-existence, whereas we must add nothing to it, if we are to speak of that itself alone. [...]" Plato – Theaetetus 201 [55]

1. We consider Information only when the intention of exchanging that Information is from a sender to recipient is defined; and we call it *Assertion*
2. Similarly, the portion of Beliefs we consider for system engineering is the one that builds (input) or describes (output) the behavior of an agent (strategy rules [19]), and
3. We consider a set of axiomatic *Facts* (definitory rules) instead of considering the more general epistemic definition of Knowledge. Specifically, Facts describes:
   - the functional architecture of each agents
   - the structural (HW/SW) architecture of each subsystem of agents and agent within a subsystem
   - the data flow as a transfer of Beliefs through Assertions (i.e. the Beliefs flow)

We give a graphical representation of (sub-)system and agent in Figure 7.

Definition 4.5. Assertion – An assertion is an intended transfer of beliefs between two agents a and b such that

(σ19) $\omega \models A_{a\to b}P$ iff $\omega \models \mathbb{B}_a P$, $\omega' \models rcc(\mathbb{B}_a P, \mathbb{B}_b P)$, $\omega' \models \mathbb{B}_a\mathbb{B}_b P$ and $\omega' \models P$ for all ω' such that $\omega R \omega'$.

Definition 4.6. Behavior – The behavior of an agent is defined as the transformation process (e.g. defined as a protocol, or a functional architecture) that determined the output-beliefs based on input-beliefs (where input-beliefs and output-beliefs are beliefs taken as input or output respectively).

(σ21) $\omega \models \bigcup_{s\in S} B_s P_s$ and $\omega \models B_a[\bigcup_{s\in S} B_s P_s]$ iff $\omega \models \mathbb{B}_a P'$,
where a is an agent, S is a set of agents such that $a\neg \in S$, and the calculation of P' is deteministic.

Definition 4.7. Fact – Facts are a sub-region of Knowledge $\mathcal{F} = \bigcup_\Phi K_a\varphi$ that predicate in a factive way, e.g. if a knows that P then P, and the region of facts is monotone (no revision).

(σ20) $\omega \models F_a P$ iff $\omega \models \Box P$.

Definition 4.8. Operational System State – An operational system (or a sub-system) is represented as an agent (see Definition 3.5) and then defined by the resulting state space, such that $s = \langle rcc(\mathcal{F},\mathcal{B}), rcc(\mathcal{F},\mathcal{A}), rcc(\mathcal{B},\mathcal{A})\rangle$, where $\mathcal{A},\mathcal{B},\mathcal{F}$, are regions of assertions, behavior, and facts respectively.

[19] "The logical structure of information is one of the most basic and one of the most basic and one of the simplest thing in the wide and wonderful world of logical analysis. This point can be put in a deeper perspective. A distinction [...] ought to be made [–] between two kinds of rules (or principles) in any strategic activity like knowledge seeking. On the one hand you have the rules that define the game, e.g. how chessmen are moved on a board. The can be called *definitory* rules. They must be distinguished from rules [...] that deal with what is better and what is worse in the game in question. Definitory rules do not say anything about this subject. Rules which do can be called *strategic rules*" – Hintikka in [41]

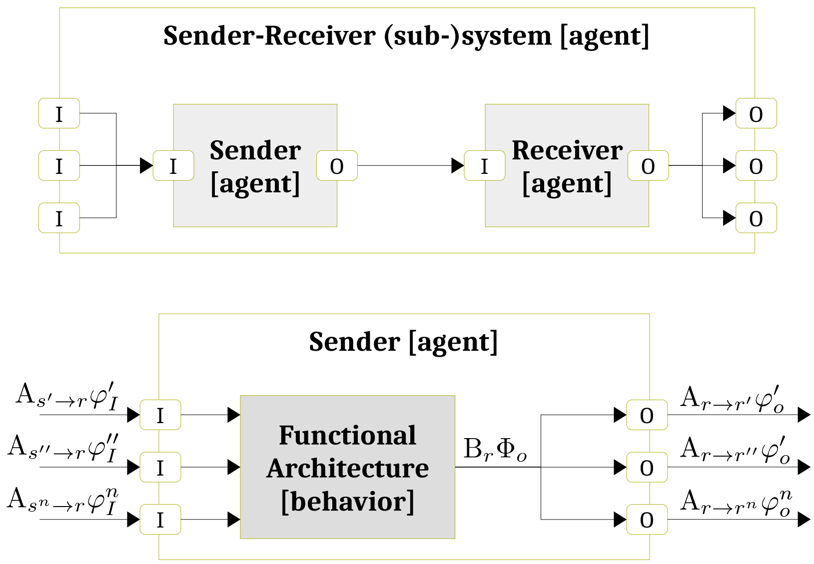


Figure 7: Example of system and agent structure

As already presented in [63] it follows that, defining a system (or an operational system) with a fixed number of regions, there exist an upper-bound to the number of possible configuration of a system, defined by the possible relations between the different regions. For completeness, we report in the next paragraph the calculation done in [63].

Number of different configurations of a system. The general formula to calculate the number of different types of agents is $r^{\binom{n}{k}}$, where r is the number of relations with arity k, between n different sets, where r^e is the number of permutation of r relations over e elements with repetitions, with e being the number of k-ary combinations of n sets, $\binom{n}{k}$. In our case, $\binom{n}{k} = 3$ since we consider 3 sets $(\mathcal{A}, \mathcal{B}, \mathcal{F})$, and all the relations considered in the RCC are binary. Hence, using RCC5 (with five different spatial relations) over three sets, we can theoretically define up to 125 different type of agents. However, only 54 of the 125 (as showed in [33]) combinations are topologically correct with respect to the definition of the relations of RCC5. Generalizing to all the RCCs:

- *RCC3* — theoretical: $3^3 = 27$, correct: 15
- *RCC5* — theoretical: $5^3 = 125$, correct: 54
- *RCC8* — theoretical: $8^3 = 512$, correct: 193

Hence, even if considering a different number of sets than the three $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{F}$ exponentially affects the number of theoretical agents, the application of RCC downscales that number of a factor that ranges from 1.8 to 2.5. In addition, using RCC5 we consider 3.6 times more (different) types of agents than RCC3, but using RCC8 would allow us to consider 3.5 times more different agents.

4.3 System Security Evolution

tofix: Facts should be correlated to the causality principle. Therefore a specification in opnet should be translated into vmt and and by bounded model checking with nuxmv identify a path towards an undesired state, if that fails use abf to mutate the first state into one of the possible according to abf and re-run nuxmv.

Knowledge is the theory itself, an instance is the facts of a design of it. A testing process tests facts against knowledge as much as we test a cps design chiices against its specification. Information carries the knowledge of the negation of the opposite of the information. Therefore an assertion carries the belief (as a result of the behavior execution) of not the opposite. But, as Hintikka States in "popper" our previous statement is true if the probability distribution of the information is representative wrt the causality structure (i.e. knowledge and facts, see email before). So, based on our theory, the assertions should have a number between 3 to 5 possible negated meaning, one per each rcc if reference and not just a negation of the information. Those negation representative of an insecure configuration on the communication of the system. In turn, those configurations should categorize all the CWE. The fact that weaknesses are generated by mankind but not exploited makes malicious attacks confined to the hackers (and black-hat hackers). Fix: move black-hat hackers as subregion of mankind with malicious intent. Describe the intent as a desire of reaching a breach of a cybersecurity property over an asset (agent of the system), instead of searching for a specific attack. The tool would then translate the system from opnet to abf formula. The abf tool generates all the sat state of the system fix: add state to all def before this text. And we shall define the causality relation between the different states generated by the abf tool as chosen nondeterministically from the possible sat states by a model checker such as nuxmv.

By calculating sat instead of validity with abf tool and then adding it to the causality structure of a system fact description, e.g. in nuxmv; we can calculate a path for reachability of an insecurity property violated on an agent which represents an asset. Whenever nuxmv fails, we can add a new state and re-run the process. Those guesses should be better managed by a trained v-ai since the inference by induction from the partial information of the system in which we represent the assertions as negation of a beliefs and without rcc do not carry enough information for a Reasoner that doesn't implemented rcc.

4.4 Security Properties and Assets

In [32], the BDI (Belief, Desire, Intent) paradigm is explained; where an agent can be defined over those three constituents.

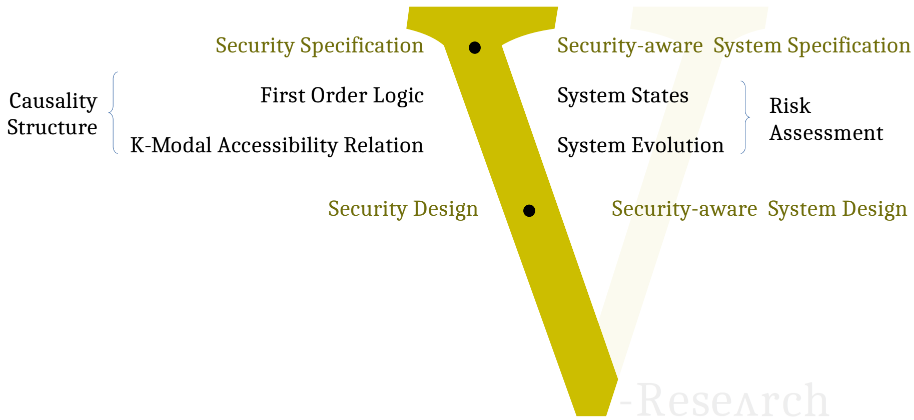


Figure 8: Cybersecurity Engineering Life-cycle

5 Cybersecurity Engineering Process

> The difference between ideas and reality is the difference between philosophy and engineering. The work to transform one into the other is scientific research
>
> *V-Research*

5.0.1 Relevant Standards

1. DO-326A

1. specification – definition of the desired design of a system. The specification is verified w.r.t. the $\mathcal{A}, \mathcal{B}, \mathcal{F}$ theory. Checking the controls of the whole cybersecurity life-cycle and the relation with CWE.

2. design – the mitigations identified in the specification stage are implemented into the design. The security of the assets w.r.t. the design are verified with a, so called, cybersecurity risk assessment.

6 Prototype and Empirical Tests

6.1 Prototype Implementation

6.2 Empirical Tests

7 Conclusion and Related Work

References

[1] International Electrotechnical Commission (IEC). *IEC 61511-1:2016+AMD1:2017 CSV Consolidated version*. Aug. 16, 2017. URL: https://webstore.iec.ch/publication/61289 (visited on 01/21/2020).

[2] Alessandro Armando, Roberto Carbone, and Luca Compagna. "SATMC: a SAT-based model checker for security protocols, business processes, and security APIs". In: *International Journal on Software Tools for Technology Transfer* 18.2 (2016), pp. 187–204.

[3] David Basin, Sebastian Mödersheim, and Luca Vigano. "OFMC: A symbolic model checker for security protocols". In: *International Journal of Information Security* 4.3 (2005), pp. 181–208.

[4] Rebecca M. Blank and Patrick D. Gallagher. "NIST Special Publication 800-53 Revision 4 - Security and Privacy Controls for Federal Information Systems and Organizations". In: *National Institute of Standards and Technology Special Publication* (Apr. 2013). URL: http://dx.doi.org/10.6028/NIST.SP.800-53r4.

[5] Blanchet Bruno. "Symbolic and Computational Mechanized Verification of the ARINC823 Avionic Protocols". In: *Computer Security Foundations Symposium (CSF)*. IEEE, Aug. 2017, pp. 68–82.

[6] Tom Caddy. "Penetration Testing". In: *Encyclopedia of Cryptography and Security*. Ed. by Henk C. A. van Tilborg. Boston, MA: Springer US, 2005, pp. 456–457. ISBN: 978-0-387-23483-0. DOI: 10.1007/0-387-23483-7_297. URL: https://doi.org/10.1007/0-387-23483-7_297.

[7] *CAPEC-104*. Sept. 30, 2019. URL: https://capec.mitre.org/data/definitions/104.html (visited on 02/19/2020).

[8] *CAPEC-73*. Sept. 30, 2019. URL: https://capec.mitre.org/data/definitions/73.html (visited on 02/19/2020).

[9] *CAPEC-81*. Sept. 30, 2019. URL: https://capec.mitre.org/data/definitions/81.html (visited on 02/19/2020).

[10] *CAPEC-85*. Sept. 30, 2019. URL: https://capec.mitre.org/data/definitions/85.html (visited on 02/19/2020).

[11] *Common Attack Pattern Enumeration and Classification*. URL: https://capec.mitre.org/ (visited on 02/19/2020).

[12] *Common Weakness Enumeration (CWE)*. Jan. 9, 2019. URL: https://cwe.mitre.org/ (visited on 02/04/2020).

[13] The MITRE Corporation. *CVE – Common Vulnerabilities and Exposures*. Jan. 16, 2020. URL: https://cve.mitre.org/ (visited on 01/22/2020).

[14] *CVE-2008-0005*. Dec. 3, 2007. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0005 (visited on 02/04/2020).

[15] *CVE-2008-0757*. Feb. 13, 2008. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0757 (visited on 02/04/2020).

[16] *CVE-2008-0769*. Feb. 13, 2008. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0769 (visited on 02/04/2020).

[17] *CVE-2008-3773*. Aug. 22, 2008. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-3773 (visited on 02/04/2020).

[18] *CVE-2008-4636*. Oct. 21, 2018. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4636 (visited on 02/04/2020).

[19] *CVE-2008-5573*. Dec. 15, 2008. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-5573 (visited on 02/04/2020).

[20] *CWE VIEW: Research Concepts*. 2020. URL: https://cwe.mitre.org/data/definitions/1000.html (visited on 02/03/2020).

[21] *Cybersecurity: Industry Report & Investment Case*. June 25, 2018. URL: https://www.nasdaq.com/articles/cybersecurity-industry-report-investment-case-2018-06-25 (visited on 01/22/2020).

[22] *Dialectical Materialism*. Progress Publishers, 1983. URL: https://www.marxists.org/reference/archive/spirkin/works/dialectical-materialism/index.html.

[23] *Digital Single Market – Policy – Cybersecurity industry*. Sept. 30, 2019. URL: https://ec.europa.eu/digital-single-market/en/cybersecurity-industry (visited on 01/22/2020).

[24] *Digital Single Market – Policy – Standards*. Dec. 6, 2019. URL: https://ec.europa.eu/digital-single-market/en/standards-digitising-european-industry (visited on 01/22/2020).

[25] *Digital Single Market – Press Release – Commission signs agreement with cybersecurity industry to increase measures to address cyber threats*. July 5, 2016. URL: https://ec.europa.eu/digital-single-market/en/news/commission-signs-agreement-cybersecurity-industry-increase-measures-address-cyber-threats (visited on 01/22/2020).

[26] Danny Dolev and Andrew Yao. "On the security of public key protocols". In: *IEEE Transactions on information theory* 29.2 (1983), pp. 198–208.

[27] Sextus Empiricus. *Outline of Pyrrhonism*. Prometheus Book, 1990. ISBN: 978-0-87975-597-3.

[28] Institution of Engineering and Technology (IET). *Glossary of safety terminology.* Jan. 2017. URL: `https://www.theiet.org/media/1435/hsb00.pdf`.

[29] Santiago Escobar, Catherine A. Meadows, and José Meseguer. "Maude-NPA: Cryptographic Protocol Analysis Modulo Equational Properties". In: *Foundations of Security Analysis and Design (FOSAD) - Tutorial Lectures.* 2007, pp. 1–50.

[30] *FAQ – What is the difference between a software vulnerability and software weakness?* Sept. 17, 2019. URL: `https://cwe.mitre.org/about/faq.html#A.2` (visited on 02/03/2020).

[31] James Garson. *Modal Logic.* Ed. by Edward N. Zalta. 2018.

[32] Michael P. Georgeoff and Anand S. Rao. "The Semantics of Intention Maintenance for Rational Agents". In: *International Joint Conference on Artificial Intelligence (IJCAI).* 1995.

[33] Rolf Grütter, Thomas Scharrenbach, and Bettina Bauer-Messmer. "Improving an RCC-Derived Geospatial Approximation by OWL Axioms". In: *ISWC.* 2008.

[34] Rolf Grütter, Thomas Scharrenbach, and Bettina Bauer-Messmer. "Improving an RCC-Derived Geospatial Approximation by OWL Axioms". In: *The Semantic Web (ISWC).* Ed. by Amit Sheth et al. Springer Berlin Heidelberg, 2008, pp. 293–306.

[35] Albert B. Hakim. *Historical Introduction to Philosophy.* Routledge, 2016. ISBN: 978-0-13-190005-9.

[36] Cormac Herley. "Justifying Security Measures—a Position Paper". In: *European Symposium on Research in Computer Security.* Springer. 2017, pp. 11–17.

[37] Cormac Herley. "So long, and no thanks for the externalities: the rational rejection of security advice by users". In: *Proceedings of the 2009 workshop on New security paradigms workshop.* 2009, pp. 133–144.

[38] Cormac Herley. *The Unfalsifiability of Security Claims - Invited Talk USENIX Security.* Aug. 2016. URL: `https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/herley` (visited on 01/15/2020).

[39] Cormac Herley. "Unfalsifiability of security claims". In: *Proceedings of the National Academy of Sciences (PNAS)* 113.23 (2016), pp. 6415–6420.

[40] Jaakko Hintikka. "Knowledge and Belief: An Introduction to the Logic of the Two Notions". In: *Studia Logica* 16 (1962), pp. 119–122.

[41] Jakko Hintikka. "On proper (popper?) and improper uses of information in epistemology". In: *Theoria.* Wiley Online Library, 1993, pp. 158–165. URL: `https://doi.org/10.1111/j.1755-2567.1993.tb00867.x`.

[42] *Improper Encoding or Escaping of Output.* June 20, 2019. URL: `https://cwe.mitre.org/data/definitions/116.html` (visited on 02/04/2020).

[43] Merriam-Webster Inc. *failure*. 2020. URL: https://www.merriam-webster.com/dictionary/failure (visited on 01/18/2020).

[44] Merriam-Webster Inc. *Realization*. 2020. URL: https://www.merriam-webster.com/dictionary/realization (visited on 02/19/2020).

[45] Wikipedia Foundation Inc. *Belief–desire–intention software model*. Dec. 22, 2019. URL: https://en.wikipedia.org/wiki/Belief%E2%80%93desire%E2%80%93intention_software_model (visited on 01/21/2020).

[46] Wikipedia Foundation Inc. *Classical Definition of Knowledge – Epistemology*. Aug. 26, 2014. URL: https://en.wikipedia.org/wiki/Epistemology#/media/File:Classical_definition_of_Kno.svg (visited on 01/27/2020).

[47] Wikipedia Foundation Inc. *Computer Security*. Feb. 17, 2020. URL: https://en.wikipedia.org/wiki/Computer_security (visited on 02/19/2020).

[48] Wikipedia Foundation Inc. *Exploit (computer security)*. Nov. 23, 2019. URL: https://en.wikipedia.org/wiki/Exploit_(computer_security) (visited on 01/21/2020).

[49] Wikipedia Foundation Inc. *Mankind*. Dec. 19, 2019. URL: https://en.wikipedia.org/wiki/Mankind (visited on 01/15/2020).

[50] Wikipedia Foundation Inc. *Nature*. Jan. 14, 2020. URL: https://en.wikipedia.org/wiki/Nature (visited on 01/15/2020).

[51] Tsau Young Lin, Qing Liu, and Y. Y. Yao. "Logics Systems for Approximate Reasoning: Approximation via Rough Sets and Topological Spaces". In: *ISMIS*. 1994.

[52] Peter Mell, Karen Scarfone, and Sasha Romanosky. "A complete guide to the common vulnerability scoring system version 2.0". In: *Published by FIRST-forum of incident response and security teams*. Vol. 1. 2007, p. 23.

[53] Committee on National Security Systems (CNSS). "Glossary No 4009". In: *National Information Assurance (IA) Glossary* (Apr. 6, 2015). URL: https://rmf.org/wp-content/uploads/2017/10/CNSSI-4009.pdf.

[54] Jakob Nielsen. *Stop password masking*. June 22, 2009. URL: https://www.nngroup.com/articles/stop-password-masking/ (visited on 01/22/2020).

[55] Plato et al. *Plato in Twelve Volumes: With an English Translation*. Ed. by Heinemann H. Harvard University Press, 1914. URL: http://www.perseus.tufts.edu/hopper/text?doc=Perseus%3Atext%3A1999.01.0172%3Atext%3DTheaet.%3Apage%3D201#note1.

[56] R. Karl Popper. *Conjectures and refutations: The growth of scientific knowledge*. New York London, 1962.

[57] Saikeerthi Rachavelpula. "The Category of Mereotopology and Its Ontological Consequences". In: *University of Chicago Mathematics Research Program*. Ed. by Michael Neaton and Peter Peter. 2017.

[58] MIT Research and Engineering (MITRE). *Common Vulnerabilities and Exposures (CVE)*. URL: https://cve.mitre.org/ (visited on 01/27/2020).

[59] Patricia A Robinson. *Writing and designing manuals and warnings*. CRC Press, 2019.

[60] Marco Rocchetto, Martín Ochoa, and Mohammad Torabi Dashti. "Model-Based Detection of CSRF". In: *Proceedings of the ICT Systems Security and Privacy Protection – IFIP TC International Conference, SEC*. 2014, pp. 30–43.

[61] Marco Rocchetto and Nils Ole Tippenhauer. "CPDY: extending the Dolev-Yao attacker with physical-layer interactions". In: *International Conference on Formal Engineering Methods*. Springer. 2016, pp. 175–192.

[62] Marco Rocchetto, Luca Viganò, and Marco Volpe. "An interpolation-based method for the verification of security protocols". In: *Journal of Computer Security* 25.6 (2017), pp. 463–510.

[63] Katia Santacà et al. "A Topological Categorization of Agents for the Definition of Attack States in Multi-agent Systems". In: *Proceedings of the European Conference on Multi-Agent Systems and Agreement Technologies (EUMAS)*. 2016, pp. 261–276.

[64] Barry Smith. "Mereotopology: A theory of parts and boundaries". In: *Data & Knowledge Engineering* 20.3 (1996). Modeling Parts and Wholes, pp. 287–303.

[65] Eugene H Spafford. *Quatable Spaf*. Dec. 4, 2019. URL: https://spaf.cerias.purdue.edu/quotes.html (visited on 02/19/2020).

[66] Richard Stallman. *The Hacker Community and Ethics: An Interview with Richard M. Stallman*. 2002. URL: https://www.gnu.org/philosophy/rms-hack.html (visited on 01/22/2020).

[67] International Organization for Standardization (ISO). *ISO/IEC/IEEE 15288:2015, Systems and Software Engineering – System Life Cycle Processes*. May 2015. URL: https://www.iso.org/standard/63711.html (visited on 02/20/2020).

[68] International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). "Information technology – Security techniques – Information security management systems – Overview and vocabulary". In: (2009). URL: http://standards.iso.org/ittf/PubliclyAvailableStandards/c041933_ISO_IEC_27000_2009.zip (visited on 01/22/2020).

[69] National Institute of Standards and Technologies (NIST). *National Vulnerability Database*. URL: https://nvd.nist.gov/ (visited on 01/22/2020).

[70] National Institute of Standards and Technologies (NIST). *Official Common Platform Enumeration (CPE) Dictionary Statistics*. URL: https://nvd.nist.gov/products/cpe/statistics (visited on 01/27/2020).

[71] Matthias Steup and Ram Neta. “Epistemology”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2020. Metaphysics Research Lab, Stanford University, 2020.

[72] *The Youtube Poll Expert*. 2019. URL: `https://www.pollpro.com/` (visited on 02/17/2020).

[73] John Turri. “Is knowledge justified true belief?” In: *Synthese* 184.3 (2012), pp. 247–259.

[74] Mathieu Turuani. “The CL-Atse protocol analyser”. In: *International Conference on Rewriting Techniques and Applications*. Springer. 2006, pp. 277–286.

[75] Achille C. Varzi. “On the Boundary Between Mereology and Topology”. In: *Proceedings of the International Wittgenstein Symposium*. 1994, pp. 261–276.

[76] Ricardo Villadiego. *The Need For A Breakthrough In Cybersecurity*. Oct. 9, 2019. URL: `https://www.forbes.com/sites/forbestechcouncil/2019/10/09/the-need-for-a-breakthrough-in-cybersecurity/#520e08839f1f` (visited on 01/22/2020).

[77] David Von Oheimb and Sebastian Mödersheim. “ASLan++—a formal security specification language for distributed systems”. In: *International Symposium on Formal Methods for Components and Objects*. Springer. 2010, pp. 1–22.