L. Cazorla, C. Alcaraz, and J. Lopez, "A Three-Stage Analysis of IDS for Critical Infrastructures", *Computers & Security*, vol. 55, pp. 235-250, 2015.
NICS Lab. Publications: https://www.nics.uma.es/publications

A Three-Stage Analysis of IDS for Critical Infrastructures

Lorena Cazorla, Cristina Alcaraz, Javier Lopez

Computer Science Department,
University of Malaga, Spain
{lorena,alcaraz,jlm}@lcc.uma.es

Dated: November, 2015

Abstract

The correct operation of Critical Infrastructures (CIs) is vital for the well being of society, however these complex systems are subject to multiple faults and threats every day. International organizations around the world are alerting the scientific community to the need for protection of CIs, especially through preparedness and prevention mechanisms. One of the main tools available in this area is the use of Intrusion Detection Systems (IDSs). However, in order to deploy this type of component within a CI, especially within its Control System (CS), it is necessary to verify whether the characteristics of a given IDS solution are compatible with the special requirements and constraints of a critical environment. In this paper, we carry out an extensive study to determine the requirements imposed by the CS on the IDS solutions using the Non-Functional Requirements (NFR) Framework. The outcome of this process are the abstract properties that the IDS needs to satisfy in order to be deployed within a CS, which are refined through the identification of satisficing techniques for the NFRs. To provide quantifiable measurable evidence on the suitability of the IDS component for a CI, we broaden our study using the Goal Question Metric (GQM) approach to select a representative set of metrics. A requirements model, refined with satisficing techniques and sets of metrics which help assess, in the most quantifiable way possible, the suitability and performance of a given IDS solution for a critical scenario, constitute the results of our analysis.

Keywords: Control Systems, Critical Infrastructures, Requirements, Satisficing Techniques, Metrics.

1 Introduction

Practically all our *Critical Infrastructures* (CIs) are today under the supervision and are dependent on other additional systems whose underlying infrastructures in turn, rely heavily on the new Information and Communication Technologies (ICTs) for control. Indeed, ICTs have now become essential elements in our society because they offer significant benefits to improve efficiency, cost reduction and enhance quality of life. Mobile computing technologies, distributed systems, smart devices, wireless communication or cloud-computing are becoming the major driving forces behind the management of diverse information, allowing a quicker operation of the great majority of today's competitors' infrastructures and their services.

In fact, most of these physical facilities are highly interconnected to other national (and international) infrastructures through communication systems, and are managed through ICTs [10]. This new way of monitoring makes the present *Control Systems* (CSs) critical in themselves, where the notion of critical-

ity is intertwined with the nature of the system and its sensitivity to adverse events caused by unforeseen faults or intentional threats. This also means that CIs and their minimum services (e.g., water, energy or transport) are also dependent on the effectiveness of the ICTs integrated inside CSs in charge of collecting, distributing and processing the correct functional performance of resources and the provision of services.

Examples of CSs are *Distributed Control Systems* (DCSs) or *Supervisory Control and Data Acquisition* (SCADA) systems, and both belong to the category of Industrial Control Systems (ICS) [8]. SCADA systems, in particular, are composed of hybrid integral systems in which a set of control processes is widely distributed over large geographical areas and the information is centralized at a single point, the SCADA Center. Generally speaking, SCADA systems are commonly sensitive to a number of threatening factors: (deliberate/unintentional) faults and existing vulnerabilities related to access control, communication or control. All of them may imply not only the degradation of the minimum monitoring services but also the neglect of other essential services for society, which could even result in the well-known *cascading effect* between infrastructures [48].

The proof of this is found in annual reports published by different governments through specific organizations such as the *European Union Network and Information Security Agency* (ENISA) [25] and the *Industrial Control System Cyber Emergency Response Team* (ICS-CERT) [52, 53], respectively. Both reflect the current situation and the severity of potential threats, where the number of specific incidents apparently continues to grow. This requires an immense effort to design protection measures without infringing the five basic control principles defined in [11]: *real-time operational performance, dependability, survivability, sustainability* and *safety critical*.

These five requirements are basic because they encompass a further set of important conditions, such as availability, integrity, access to component, component lifetime, change management and reliability, among others [27]. Many of these requirements are also included in the guidelines published by the National Institute of Standards and Technology (NIST) in [4] for ICSs and for Smart Grids in [7], and even in the guidelines published by the North American Electric Reliability Corporation (NERC) through its NERC-CIP series (002-009) [5].

On the other hand, we have so far identified three chief vulnerabilities in CSs: (i) weaknesses in the configurations of the CIs (network configurations, security policies, defense mechanisms, etc.); (ii) architectural design of the system; and (iii) errors in the development, which are generally related to software (SW) and hardware (HW) elements. Many of these vulnerabilities come from modernizing the TCP/IP-based technologies, increasing, on the one hand, the complexities of the CS and, on the other hand, adding new vulnerabilities to the control.

Indeed, existing SCADA architectures, their devices and their protocols have to adjust to the new technological changes to offer network architectures which are distributed, secure and autonomous for data management in real time, interoperability between protocols (e.g., Modbus or DNP3 with ISA100.11a, WirelessHART or ZigBee PRO) and scalability for the cooperation and collaboration with the new control technologies. A clear example of this new change is found in the new electrical distribution generation, i.e, in the Smart Grid generation.

In the Smart Grid, CSs serve as the central edge of supervision and data acquisition from thousands to millions of smart devices with direct connection to diverse networks: backhaul, *Wide Area*, *Field Area*, *Neighborhood Area*, and *Local Area Networks*. In this context, backhaul and the Internet are the chief sources that connect the different sub-domains with the rest of the networks, including *Advanced Metering Infrastructures* that characterize the bidirectional interfaces between the real world, and the acquisition and control world.

Through these interfaces it is possible to manage and interact with smart meters and utility business systems, substituting the traditional one-way advanced meters. This technological evolution also shows how CSs are becoming more complex at the different levels (functionally, architecturally), and hence also require special care when adapting new technologies. In other words, finding a perfect connection between systems and guaranteeing secure monitoring at

all times requires, for each adapted technology, a set of minimum requirements which should not interfere with the basic control principles.

Specifically, this paper looks at those requirements that *Intrusion Detection Systems* (IDSs) should consider since they act as the main way of filtering and defense in CSs. We also explore the set of metrics that help determine the compliance level of the requirements of the IDSs in relation to the five control requirements. It is necessary to focus on detection technologies due to the current problems in finding a suitable IDS capable of offering sufficient means to not only understand any SCADA vulnerability and all SCADA traffic (i.e., property protocols), but also to guarantee sustainability, coexistence and scalability when other types of TCP/IP protocols (e.g., 6LowPAN for Internet of Things) have to be integrated within the system.

To the best of our knowledge, the current SCADA literature does not contain any extensive theoretical study on identifying those IDS requirements that have to be considered when implementing IDS solutions for ICSs. There are only specific-purpose approaches restricted to a set of factors [58],1805, e.g., the type of observed protocol (e.g., the SCADA IDS defined within the SPARKS European project can identify anomalies associated with IEC-61850 and IEC-60870-5 networks [32]), the acquisition architecture (centralized, distributed), the level of scalability and detection granularity, the type of response (passive, active), or the degree of interoperability.

Given this, the main contribution in our paper is the provision of an implementation guideline for IDS solutions for CSs based on the intrinsic characteristics of IDSs and in relation to the context features. We also try to assist in the elaboration of future adaptable intrusion detection applications for those ICSs that are becoming increasingly complex and dynamic.

This paper is organized as follows, Section 2 revises the special requirements and constraints of CSs. Based on this analysis, in Section 3 we explore the requirements imposed by this critical scenario on a given IDS solution that would be deployed within this environment. In Section 4, we further develop our study by determining satisficing techniques that would help fulfill the previously identified requirements. In Section 5 we outline useful metrics which can help quantitatively evaluate the suitability of the given IDS solution for the CSs. Finally, Section 6 concludes the paper and highlights future work.

2 The Special Requirements and Constraints of Critical Control Systems

As stated, CSs are systems that manage other critical systems (also known as "systems of systems"). Specifically, ICSs are a type of control system, deployed to aid in the operation of industrial infrastructures, the services of which are also essential to social and economic welfare. This feature means that CSs can also be considered as *Critical Control Systems* (CCSs), where the correct operation of their control services against unforeseen and/or dynamic changes is of paramount importance.

2.1 CCS Requirements

CCS are complex systems, and this complexity is due to several factors: (i) ICSs are composed of multiple networks with thousands of nodes in them; (ii) the heterogeneity of the network is high, integrating multiple types of nodes and technologies, i.e., legacy equipment (e.g., old *Remote Terminal Units* (RTUs) and industrial sensors) using protocols such as Modbus/TCP [40] running alongside modern devices (e.g., cloud servers, wireless sensors, mobile devices, etc.) which use technologies such as Bluetooth [14] or ISA100.11a [45]; (iii) ICS components and subsystems have many dependencies between them, and CCSs also have interdependencies with other CIs [48].

These factors make CCSs challenging systems to manage, moreover, they also make the ICSs targets vulnerable to attacks [25]. In recent years, cyber attacks targeting CIs have increased exponentially, as shown in the cases of Stuxnet [38], Duqu [34], the Nitro attacks [18] and others [22, 51].

Due to their complexity, CCSs need to be analyzed in order to better understand their requirements, to

provide them with adequate protection against the multiple threats they face because of their characteristics. In [11], the authors compile the requirements that a CCS must comply with to achieve the right levels of security and performance. We describe them here, as these requirements form the basis of our study:

- *Real-time performance*: CCSs have hard real-time constraints regarding communications, execution processes and system upgrading, as none of them should cause delays in the system. The communications' response time is heavily constrained, sometimes tightened to a maximum of one millisecond [8]. Additionally, naturally occurring faults in CIs, or malicious activities can introduce delays in the system, something that needs to be palliated and reduced as soon as possible.

- *Dependability*: is "*the ability of a system to properly offer its services on time, avoiding frequent and severe internal faults*" [11], thus a control system must provide its service despite fault occurring. Dependability comprises five attributes that absolutely have to be observed: a*vailability, reliability, maintainability, safety* and *security* [11].

- *Sustainability*: as defined in [11], sustainability is "*the ability of a system to meet the needs of the present without compromising its ability to meet future needs*", i.e., the system must continue to function like the day it was deployed despite any later updates, upgrades or modification of its components (hardware and software).

- *Survivability*: is "*the capability of a system to fulfill its mission and thus to face malicious, deliberate or accidental faults in a timely manner*" [11]. Survivability is composed of three main elements: *unsusceptibility*, *resilience* and *recoverability* (defined below).

- *Safety critical*: this is safety related to critical environments; its implementation makes it possible to prevent unplanned effects that the failure of a critical system could inflict on society. It also relates to the protection against faults cascading from critical infrastructure to another, the so-called "*cascading effect*" [11, 48].

These requirements are common to all critical environments. These scenarios are highly complex and heterogeneous and, as we have mentioned, they combine multiple technologies and are subject to many restrictions and constraints. This makes it difficult to use the same state-of-the-art in ICTs as in regular, non-critical information systems. In Figure 1, we represent the aforementioned CCS requirements, graphically. In this figure, not only do we consider the hierarchies of the requirements and how they are composed, we also consider that all these requirements have the same level of importance or criticality.

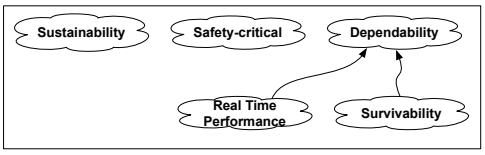

Figure 1: SCADA Requirements

The reason for the hierarchization of these requirements is purely semantic, according to the definition and concepts of safety, security and survivability engineering [26]. We understand that if one of the SCADA requirements is not met, the CCS fails to perform its tasks and is incapable of providing the required service. This failure to meet the SCADA requirements causes failures in the affected CI which could also possibly cause a cascading effect, affecting those related and interdependent infrastructures that rely on the service affected.

2.2 CCS Constraints

SCADA systems are one of the main types of CCSs used for large, geographically-dispersed distribution operations, such as electrical power grids, petroleum and gas pipelines, water or waste-water systems [58]. The special characteristics of CCSs, especially SCADA, as seen in [11], result in special constraints that control HW and SW elements deployed within

them have to comply with. Fleury et al. [27] identify these constraints and classify them in 5 different categories:

- *Performance and availability*: critical data must be available at all times, without delays or jitter in data delivery, and it must be reliable and have high integrity [27]. In addition to these constraints, any (cyber) security mechanism implemented must be fail-safe so that the failures of such mechanisms do not result in the failure of the CI.

- *Deployment and management*: CCSs need to be highly stable with respect to failures before they can be deployed because they govern physical systems with equipment deployed to last decades. What is more, their operation cannot suffer down-time for system maintenance and upgrades in the way that is common to traditional *Information Communications Technology* (ICT) systems [27]. Thus, practices such as SW patching are not trivial in CSs, since it is not practical (and sometimes impossible) to take down CCSs to apply security patches [42].

- There are strong *computation, space and storage* constraints in CCSs because they were adopted in the 1960s and although their architecture has evolved, legacy equipment, SW and protocols are still working in today's networks and need to be taken into account [8].

- A common constraint found in control systems is the strict *application timing* requirements, some of which require a message delivery time of no more than 2 ms [47].

- The *extra costs* associated with security computations, i.e., the ones performed solely to achieve a device's security goals, do not scale well in critical environments, due to the diversity of its many embedded systems [47].

Due to the above-mentioned constraints, it is necessary to provide CCS with special security measures, in a way that is compatible with their requirements. A. Nicholson et al. [42] defend that Anti-Virus, Firewalls, *Intrusion Detection Systems* (IDSs) and *Intrusion Prevention Systems* (IPSs) solutions found in general information technology networks are equally effective when employed to protect SCADA networks, but they must be tailored to the types of data used in this environment.

Apart from the use of different communication protocols and data types, tailored security solutions are particularly critical wherever the resources are constrained and the security measures applied compete with the SCADA software to perform their tasks. These security measures must respect the *responsiveness* aspect of the system, (e.g., a command from the controller to actuator should be executed in real-time by the latter), and the *timeliness* of any related data being delivered within its designated time period, also meaning *freshness* of data, i.e., the data is only valid for its assigned time period [58].

3 Requirements for Protection Solutions Deployed within Critical Systems

Since CCSs, particularly SCADA systems, are considered critical assets with a great potential impact for the society, it is vital to protect their correct operation and integrity [11]. Traditional means of protection for CSs are usually based on the triad *firewalls*, *Demilitarized Zones* (DMZs) and *antivirus* [8, 58]. However, it is necessary to go a step further and deploy monitoring tools specifically designed and adapted to critical contexts [20, 24] which will aid prevention through automatic detection, creating an environment of awareness and alert [9, 8].

These mechanisms are the IDSs [8], a security layer (HW or SW), designed to detect ongoing intrusive activities in computer systems and networks [57]. In the context of CCSs, IDSs are designed to monitor network or system activities for suspicious activities and produce reports to a management station. The IDS is also used for other purposes, such as identifying problems with security policies, documentation of threats and to dissuade individuals from violating

security policies [36].

IDS solutions, as part of the SW deployed within the CCS, are subject to special requirements and constraints. To provide good IDS solutions for CCSs is a goal that a lot of studies try to approach from many different fields and areas of knowledge, but to the best of our knowledge, still there is no study that provides a holistic point of view of the suitability of these solutions to CIs, by contrasting them against the requirements of a CCS.

In an effort to deliver such point of view, we have modeled the requirements that any IDS solution has to fulfill to be deployed within a CI. This model takes into account the need to comply with the CCS requirements identified in [11], and the constraints of CCS as described in Section 2.2. These constraints on the CCS impose powerful restrictions on the IDS solutions which is deployed in the critical environment, then our model will help build a more secure control system and satisfy its requirements, through the identification and compliance with the identified requirements for the IDS.

3.1 NFR Model Framework

Since this work covers the requirements of a system at a very high level of abstraction, the modeling is carried out in terms of non-functional requirements. A *Non-Functional Requirement* (NFR) is defined as "*a SW requirement that describes not what the SW will do, but how the SW will do it*" [19]. Examples of these NFRs are SW performance requirements, SW external interface requirements, and SW quality attributes.

NFRs are difficult to test, therefore, they are usually evaluated subjectively [19]. It is our aim, however, to develop our study further, translating these high-level requirements into quantitative information with respect to the IDS solution and its suitability within the critical environment where it will be deployed. To this end, we base our analysis on different frameworks and guidelines to be able to model our scenario.

In the first stage of this modeling, we address the study of the requirements of the system. To this end, we use the NFR Framework, as described in [19]. This framework is used to model qualitative process-oriented goals, dividing them into *non-functional requirements*, *satisficing techniques* and *claims*. Since our proposed model implies high-level non-functional requirements, instead of goals, we will model softgoals.

A *softgoal* is defined as a goal with "*no clear-cut definition and or criteria as to whether it is satisfied or not, since NFRs are subjective, relative, and interdependent*" [19]. For example, Figure 1 represents the requirements or softgoals for the SCADA system; whereas Figure 2 illustrates two main areas. The top half contains the CCS softgoals as identified in Figure 1, while the bottom half of the figure shows the IDS softgoals, needed in a critical scenario.

In other words, Figure 2 shows the application of the NFR Framework to describe the non-functional requirements or softgoals that the IDS needs to satisfy in order to comply with the CCS requirements. All the requirements are organized hierarchically and connected with other softgoals according to the definition and concepts of safety, security and survivability engineering [26]. The identification of the softgoals is a first step towards modeling the requirements of a critical complex system. The next stage of our study is the definition of the NFR softgoals for those IDS to be configured within a CCS.

3.2 NFR Requirements for Protection Systems

In this section, we define the concepts involved in the NFRs identified in the model, in order to provide more information about the non-functional requirements selected for the IDS so as to comply with the requirements of the CCS (cf. Figure 2). These definitions determine the scope of the requirements and will be used as a reference to later identify the techniques capable of satisfying the requirements.

- *Responsiveness*: is the ability of a functional system to perform an assigned task within the required time interval [56]. For our purpose, we will consider that a system is responsive when it is capable of performing its functions in the

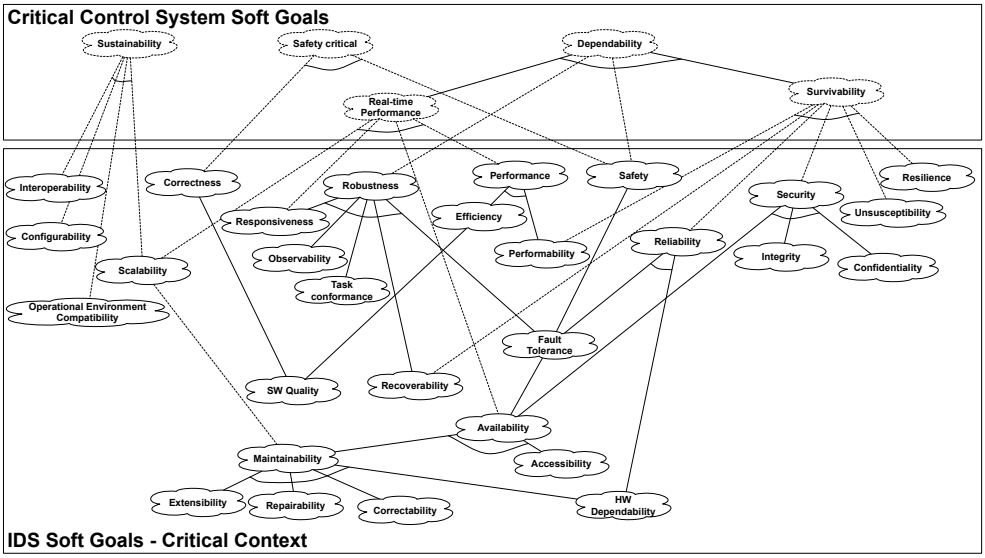

Figure 2: SCADA and IDS Softgoals

required time intervals, with no (or at least no significant) delays.

- *Safety*: From the point of view of reliability engineering, safety is defined as the "*degree to which accidental harm is prevented, reduced and properly reacted to*" [26]. A safety system has the mechanisms to prevent, reduce and react to accidental harm that could affect society in some way. A *safety critical* system is a particularization of a safety system, where the environment or the system itself is critical to social and economic welfare. Safety requirements in CIs must be always be complied with, otherwise malfunctions in the infrastructures operation could affect society and even endanger human life [11].

- *Security*: is the "*degree to which malicious harm is prevented, reduced and properly reacted to*" [26]. In CCSs it is assumed that this property must be a primary requirement for defense and protection of control assets, since the securement of CCSs is vital to maintain the normal operation of CIs [20], which in turn, is essential to society. It is necessary to consider the security variables of resources and data:

  - *Availability*: is "*the degree to which a system is in a specified operable and committable state at the start of a mission*" [50]. We also consider the *availability of information*, i.e., the information required by the system is always obtainable or accessible. Therefore, availability is defined as the *availability of a system* (asset), and the *availability of the information*. In our study, we refer to the availability of the IDS and the control resources, and to the availability of the information when it is required, regardless the filtering and processing of the IDS.
  - *Integrity*: is the quality of being honest and incorruptible [6], in computer science it relates to the consistency or lack of corruption in electronic data.
  - *Confidentiality*: describes something private, or secret [6]. In computer science, it is the way to prevent the disclosure of private or secret information.

- *Fault Tolerance*: the system that is fault tolerant is capable of continuing its operation despite the occurrence of failures; a desirable condition which guarantees the resilience or self-healing of the underlying CCS.

- *Scalability*: is the ability of something, especially an ICT system, to adapt to increased demands (e.g., control of Smart Grid environments) [6]. In computer science, it is the ability of a system, network, or process to handle a growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth [15].

- *Extensibility*: an extensible system "*is able to support new control components such as new technologies, protocols, HW and SW components, and security services*" [11]. Scalability and extensibility tend to be mixed up and confused, however they represent two very important challenges for complex heterogeneous systems like CCSs. Extensibility is a maintenance and updating characteristic which is especially desirable in a CI, where the arrival of SW applications and new technologies to CIs such as the *Internet of Things* (IoT) or *Wireless Sensor Networks* (WSN), greatly increases the need to incorporate new interoperable devices and protocols.

- *Interoperability*: is "*the ability of two or more systems or components to exchange information and to use the information that has been exchanged*" [28]. Thus we consider interoperable, a heterogeneous set of systems that are capable of exchanging useful information with each other.

- *Maintainability*: also *serviceability*, is the capability of the system to be maintained over time in order to continuously improve it and keep it free from defects and errors. Maintainability is influenced by three factors [26]: (1) *correctability*, the ease with which minor defects can be corrected between major changes, while the system is still in use; (2) *extensibility*, defined above; and (3) *repairability*, the ability of a damaged or failed system to be restored to acceptable operating conditions, within a specified period of time (*repair time*) [6].

- *Resilience*: is the capability of a system to maintain a proper service in the face of faults, as well as being able to return to normal operation as soon as possible. In terms of survivability, resilient is the antonym of vulnerable.

- *(Un)Susceptibility*: is the state of being likely or liable to be influenced or harmed by a particular thing [6]. In the context of CIs and the IDS deployed within them, unsusceptibility is a desirable requirement.

- *Recoverability*: refers to the ability to recover quickly from a system failure or disaster, to the point and (if it is possible) to the state of the system at which the failure occurred [26]. It is closely related to survivability, but it alludes to the capabilities of the system or deployment.

- *Efficiency*: is "*the degree to which something effectively uses (i.e., minimizes its consumption of) its resources (computing, machinery, facilities, and personnel)*" [26]. Related to efficiency is the *relative efficiency* of two procedures, defined as the ratio of their efficiencies. It is frequently calculated as the comparison made between a given procedure and a notional "best possible" one.

- *Correctness*: is "*the degree to which a work product and its outputs are free from defects once it is delivered*" [26]. Three main factors influence correctness, namely: *accuracy*, *currency* and *precision*, which are central in CSs in order to guarantee reliability of data and operations.

- *Operational Environment Compatibility*: is "*the degree to which a system functions correctly under specified conditions of the physical environment(s) in which it is intended to operate*" [26]. The IDS must be operationally compatible in order to be deployed within a CCS, where machinery and environmental radiation and noise might

interfere with the operation of unprepared systems.

- *Reliability*: is "*the degree to which a work product operates without failure under given conditions during a given time period*" [26]. It relates to the costs produced by hazards turning into incidents, and the level of loss of revenue for the company or the customer. It differs from safety, as safety deals with dangerous hazards which could lead to severe accidents with an impact on society.

- *Performance*: according to [26] is "*the degree to which timing characteristics are adequate*". Measurements of the quality of the performance can be *jitter*, *latency*, *response time*, *scheduleability* and *throughput*.

- *Performability*: is the characteristic of the performance of a system, viewed from the point of view of dependability, i.e., the performance of a system in the presence of faults over a specified period of time [39].

- *Robustness*: is "*the degree to which an executable work product continues to function properly under abnormal conditions or circumstances*" [26]. *Fault tolerance* is one of its main sub-quality factors. Robustness, from a usability point of view has four related criteria: *responsiveness* and *recoverability* (defined above), *observability* (consistency and inferability of the internal states of a system from the external outputs), and *task conformance* (support for the tasks established by design).

- *Configurability*: is "*the degree to which something can be configured into multiple forms (i.e., configurations)*" [26]. Configurability includes *internationalization*, *personalization*, *subsetability* and *variability*.

- *Accessibility*: is the degree to which the user interface enables users to perform their specified tasks [26]. IDSs for CCS must be accessible to the system administrators to receive updates and modifications.

- *Software Quality*: refers to the compliance of the SW with its design and established functional requirements [26]. SW quality is guided by good practices and guidelines [29], responsible for ensuring a quality SW development process and that the resulting SW complies with determined measurements and metrics of performance.

- *HW Dependability*: are the characteristics that make the HW of the system dependable (see definition above). Three characteristics indicate dependable HW [49]: *reliability* of the HW, *availability* of the HW and *maintainability* of the HW. All these characteristics are defined in a similar way to the same concepts applied to software.

The previous sections have described the scope and definitions of the NFRs identified for an IDS for CCSs. We have established and defined the characteristics present in a critical environment, and those requirements and constraints that restrict the inclusion of new components within such a scenario. In the next section, we will further develop our study in order to determine how to comply with the NFRs defined in this section.

4 NFR Satisficing Techniques for Protection Systems

In Section 3 we identified the requirements present in our scenario, taking into account both the requirements in CCSs and the ones that constrain the deployment of an IDS solution within a critical environment. The NFR Framework has provided us with tools to represent the NFRs, or softgoals, present in this scenario. However, we now need to go a step further in order to find specific ways to analyze whether these softgoals can be satisfied, and to find determined techniques or tools to help the IDS achieve this compliance.

Since we are working with NFRs, we have to address their characteristics to carry out our study. The problem is that these goals or properties lack a clear definition, as they are usually based on abstract terms

which is not very useful from a measurement perspective [43]. It is therefore difficult to assess whether or not the NFRs have been satisfied, because there is no clear-cut criteria for this evaluation.

Our approach to tackle this problem is inspired by the *Goal Question Metric* (GQM) approach [54]. The GQM approach is a goal-oriented methodology for the identification of measurements in SW engineering. It is built upon the idea of decomposing the problem into several goals, which are further refined by questions and metrics for answering them. We follow this idea to continue our study on the IDS softgoals for CIs, in order to bind these abstract characteristics to specific practices.

Instead of refining our analysis in terms of questions at the operational level of the GQM, and in line with the NFR Framework, we reflect on the operations taken for reaching the identified softgoals in terms of satisficing techniques. Thus, in this section, we aim to identify those techniques that can be implemented by the system (in our case, the IDS), capable of satisficing the established NFRs. Table 1 presents the simplified matching of the satisficing techniques found for the softgoals of the IDS, directly linked to the requirements of the CCS.

Table 1: Satisficing techniques

CCS Requirements		Techniques	
Dependability	Real-Time Performance	SW Optimization	HW Optimization
		Desegmentation	Prioritization
		Load Balancing	Testing
		Use of good practices	
	Survivability	Redundancy	Diversity
		Replication	Reaction
		Restoration	Intelligence
		Self-Healing	Self-Consciousness
Safety Critical		Isolation	Redundancy
		Replication	Prioritization
		Desegmentation	Restoration
		Use of good practices	
Sustainability		Standardization	Testing
		Modularization	Design for assurance
		Use of good practices	

This simplified matching to the NFRs of the CCS can be done because the use of these satisficing techniques for the IDS will, in turn, make the IDS comply with the requirements of the CCS. To better understand the scope of the satisficing techniques, we provide a brief description of each of them:

- *Isolation*: isolated systems are not connected to other systems or networks (e.g., the Internet). Currently, actual isolation is almost always impossible, since new technologies (e.g., the Internet, remote management, etc.) are incorporated to help manage CCS. Relative isolation is achieved through the deployment of protection layers (e.g., firewalls, DMZs) to only allow permitted communication traffic to the most critical systems of the CI.

- *Redundancy*: is the inclusion of extra components that are not strictly necessary for the normal operation, in case of failure. When the primary devices stop working due to a fault, the secondary components are activated to maintain the normal operation of the system, while the primary ones are under repair. Redundancy can be implemented by introducing exact copies of primary components in the system, or using components of different natures as redundant ones in order to maximize diversity.

- *Replication*: replication implies providing multiple identical instances of the same system (or task), all of them running in parallel. Replication benefits performance and availability (see Section 2.2), since replicated components help when there are peaks of activity. The use of replication implies that all the replicated systems are always running to balance their workload, in contrast to redundancy, where the additional components are put in place to ensure the continuation of the operation even if a system is brought down by a failure.

- *Prioritization*: is the establishment of priorities among processes in a system, to ensure that the most critical ones always have available the assets they need to operate properly. In CCSs, critical tasks need to be taken care of as soon as possible, this is made possible through the organization of tasks according to their priority.

- *Desegmentation*: implies uncoupling the processes in a system, so that they are not interdependent. This technique builds *robustness* into a

system, as independent processes are less likely to spread a cascading effect.

- *Restoration*: means to return a system to a former or original condition after the occurrence of a failure. Restoration is central to the need for mitigation and recovery techniques. International organizations provide guidelines to improve recovery capabilities in CIs [21, 41].

- *Development guided by good practices*: compliance with good practices [55] and standards [16] when designing and developing components (e.g., the IDS) for CCSs is of great importance, since the new system has to adapt to a complex environment without introducing new risks.

- *SW Optimization*: is the application of optimization techniques to the SW that is to be deployed in a critical context. Optimization should be guided by good practices.

- *HW Optimization*: implies that the deployed equipment has sufficient capabilities to perform the required tasks, and that they are used properly. All HW deployed in a critical context have to comply with the requirements of the environment, and be operationally compatible.

- *Load balancing*: is the distribution of the workload across multiple resources to maximize the throughput, minimize the response time and prevent overloads. The use of techniques such as replication can help balance the workload for CCSs' centralized systems and prevent any violations of their real-time performance requirements.

- *Testing*: provides the certainty that the tested system has determined capabilities. All SW components deployed within CIs must have been tested for errors to assess the compatibility with the environment.

- *Diversity*: implies providing different implementations of the same specification (HW or SW), and using them as replicated systems to cope with errors in a specific implementation. Diversity complements the techniques of replication and redundancy.

- *Reaction*: a system is reactive when it provides "*a response to some foregoing action or stimulus*" [6]. IPSs are specialized in reaction, and use the information offered by the IDS to decide the best response strategies. IPSs for CCSs are still subject to research, since automatic reaction in a critical environment could introduce new risks into the system and cause cascading effects.

- *Intelligence*: generates solutions capable of adapting to new circumstances and acting more efficiently against any anomalous occurrence. *Machine learning* can provide the IDS with intelligence, a vital capability to implement efficient and accurate IPSs.

- *Self-Consciousness*: allows the system to continuously monitor itself and its internal states, building security into the system. This characteristic helps provide early detection capabilities, even in the presence of sophisticated or stealthy attacks against the IDS.

- *Self-Healing*: self-healing systems are able to detect a malfunction and to react to it, returning to their normal status and operation. It complements self-consciousness, helping build robustness and preventing faults in the IDS that could affect the normal operation of the CCS.

- *Standardization*: makes the system or process compliant with standards, and is aligned with the development guided by good practices. The importance of assuring the deployment of good (certified) quality components within CIs is vital to guarantee that no IDS's failures will compromise the normal operation of the surveilled CCS.

- *Modularization*: is the design of a whole system as a set of different modules. It makes the system versatile, providing the means for modifying the structure and functionality of the system by adding or removing modules. A modularized system is also easier to maintain, given that the

complete functionality of the system is not compromised when a module needs modifications.

- *Design for Assurance*: refers to provisioning evidence for compliance to governing rules, and that the governing rules provide appropriate grounds for trustworthiness [17]. It is based on assurance cases, which make easier the accountability and the evaluation of the compliance to good practices and standards easier.

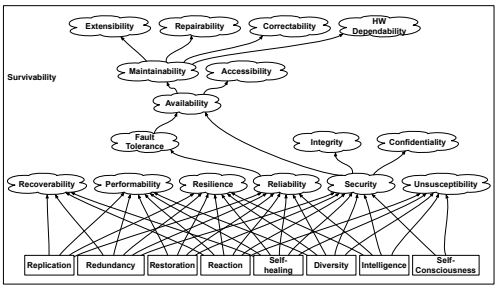

Figure 3: Survivability, softgoals and satisficing techniques

Table 1 describes the simplified matching of the satisficing techniques to CCS requirements, now we model how these techniques satisfice the IDS softgoals, which, in turn is related to the SCADA requirements. Figures 3 through 7 represent these relationships. We have divided the satisficing techniques and IDS softgoals taking into account the CCS requirements they are related to for the sake of clarity.

Figure 3 represents the relationship between the satisficing techniques and the NFRs related to the survivability of the CCS. The satisficing techniques which help the system to comply with these requirements are those that build resilience into the system, e.g., redundancy. Figure 4 presents the satisficing techniques linked to the real-time performance of the CCS. Some of the corresponding softgoals are related to the efficiency, performance and availability of the system, they try to optimize and balance the general performance of the system, to avoid peaks of demand that the system cannot respond to, e.g., SW and HW optimization and load balancing.

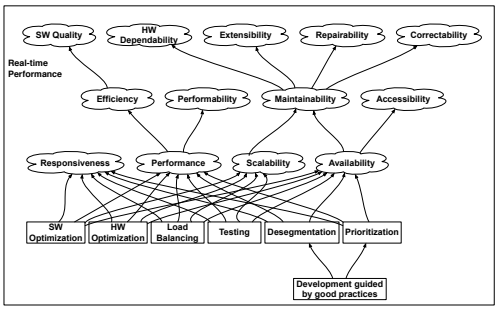

Figure 4: Real-time performance, softgoals and satisficing techniques

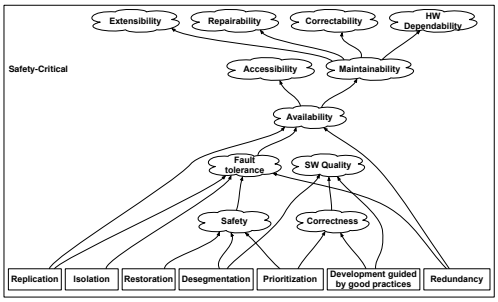

Figure 5: Safety critical, softgoals and satisficing techniques

In Figure 5 we represent those IDS requirements linked to the safety-critical of the CCS. IDS softgoals such as fault tolerance, safety or availability have a great impact on the safety-critical of the surveilled infrastructure. Thus, techniques focused on improving the IDS's safety and correct operation have a good influence on the general safety of the system, e.g., redundancy, replication, and development guided by good practices.

Figure 6 illustrates those IDS NFRs related to the sustainability of the CCS. Here, the IDS should comply with requirements such as maintainability, interoperability or scalability. Therefore the IDS should be designed in such a way as to make configuring and modifying its functionality easy. It should also make repairs or updates of their SW components easier so as to fix any problems of the system, as in modularized systems.

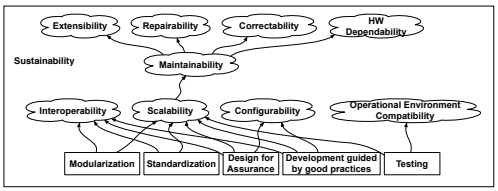

Figure 6: Sustainability, softgoals and satisficing techniques

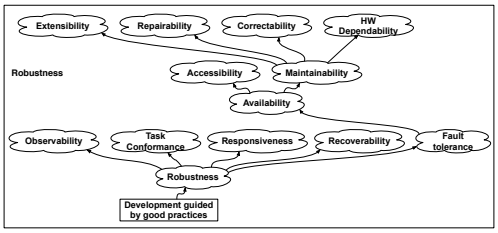

Figure 7: Robustness, softgoals and satisficing techniques

Finally, in Figure 7, we present those IDS satisficing techniques corresponding to the robustness of CCSs. Techniques that help achieve robustness are based on the design and development process of the IDS, especially the development guided by good practices. We can see in the figures that most of the aforementioned requirements and satisficing techniques overlap in the diagrams presented. This is due to the separation we previously introduced, to more clearly visualize our study.

From this study, it is possible to identify the great need for standardization and good practices when deploying new components within a critical scenario. These practices ensure that the CCSs are not affected by the addition of IDSs, and that new risks will not be introduced into the system as a source of unpredictable events or faults.

Additionally, as mentioned, there is an identified need for the CI to be protected. Some of the pillars that support this protection are the mitigation and response capabilities of the infrastructure [21, 41]. CCS are able to provide early detection and response to threats if their IDSs are equipped with sufficiently intelligent capabilities. The intelligence of the system will determine its adaptability to new circumstances and dynamics, providing the CCS with tools to react to unknown threatening events and restore the CI to its normal operation.

5 Metrics for Protection Systems

Throughout our approach, we have concentrated our efforts on studying the requirements present in CCS which influence and constrain the deployment of an IDS within the infrastructure (see Section 2). Once we have studied the NFRs and constraints of CCSs, we can identify the requirements an IDS solution has to comply with to be deployed in this environment (see Section 3). Our approach has followed the NFR Framework [19], which allows us to model the softgoals of the system.

The main problem of NFRs is that it is difficult to ascertain whether they have been satisfied or not. Their definitions have a high level of abstraction, so the constraints they impose on the system are not quantifiable. Since it is our aim to bring our study to a more specific and quantifiable area, as mentioned in Section 4, we have based our analysis on methods such as the NFR Framework [19] and the GQM approach [54].

In Section 4, we have outlined a set of satisficing techniques, inspiring the materialization of the operational level on the GQM approach [54]. Following this methodology, we now address the next level of the GQM, the quantitative level. This stage tries to identify different metrics to evaluate the suitability of the IDS for critical environments. Metrics, as denominated in the field of SW engineering, are quantitative measures of the degree to which a system or process has a given property.

For our purposes, we consider a metric as an evaluation method for assessing the level of satisfaction of certain non-functional properties in a quantitative or qualitative way, on the basis of evidence and contextual input, like, for example, stakeholders criteria [43]. In our analysis, we provide examples of sets of metrics for evaluating some NFRs of the IDS and

CCS. This connection between sets of metrics and sets of softgoals is a variation of the GQM approach, because in GQM a set of metrics is used to evaluate each operational question of the model.

We make, however, this high-level connection of sets of metrics and softgoals, in relation to the satisficing techniques. Otherwise, following the in-depth GQM analysis, the extension of this study would exceed the scope of a scientific paper. Our different sets of metrics are mere suggestions and examples, we understand that we have not been exhaustive, however we do refer to different standards and guides where it is possible to find extensive information and different implementations and formulae expressing detailed metrics within these sets.

5.1 Reliability and Availability Metrics

In this subsection, we present metrics related to the *reliability and availability* of a system. These metrics can help determine and quantify the availability and reliability parameters of given IDS solution. With these measurements, we try to provide quantifiable evidence for several parameters of the system that can help determine whether or not the IDS is sufficiently reliable and available to be deployed within a CCS.

- *Diversity*: measures the number of different implementations of the same specification, the more diverse a system is, the more resilient to failures it is. Examples are: number of platforms where the IDS can run, and number of communication protocols the IDS can understand.

- *Replication*: measures the number of replicated systems that are present in the system. It can also be applied to a component, in order to identify its level of replication, e.g., RAID systems: level 0 to level 7.

- *Uptime*: is the measure of the time a system is working and available, representing the time it can work non-stop and without maintenance. Examples are: the percentage of time the system is running and number of hours uptime versus number of hours of outage/downtime.

- *Downtime*: opposite to uptime, it represents the periods of time that the system is unavailable or off-line because of unplanned events or maintenance routines. An example is the percentage of time the system is down.

- *Availability*: is the time that the system is not failed and not under repairs [33]. In this set we usually find metrics related to maintainability, such as the mean time between failures, which are discussed below.

5.2 Performance and Responsiveness Metrics

Metrics related to *performance and responsiveness* are also useful for assessing IDS within CCSs. We have selected some of them as examples, describing the ones we think are representative of our scenario.

- *Jitter*: is "*the precision of the time when one or more events occur*" [26]. An example is the *absolute jitter*, that measures the difference between the instant one event occurs and the ideal instant when it should have happened.

- *Latency*: is the time it takes to provide a requested service or allow access to resources [26], which provides information about the delay of the system. Examples are the communication latency or operational latency.

- *Response time*: is the time it takes to initially respond to a request for a service or to access resources [26]. It is vital for the CCS surveilled by the IDS.

- *Schedulability*: is "*the degree to which events can be scheduled and then occur at their scheduled times*" [26]. An example is the acceptance ratio, i.e., number of accepted task with respect to the feasible ones.

- *Throughput*: is "*the number of times that a service can be provided within a specified unit of*

time" [26]. An example is the IDS's throughput, i.e., the number of items processed by the IDS over a defined period of time.

- *Compression Ratio*: refers to the reduction of the size of the data, performed by a compression algorithm. It is usually defined as the ratio between the uncompressed size and the compressed size.

- *Speedup*: speedup is the increase of performance (speed) of a parallel algorithm versus its sequential version. In CCSs, it can be useful to measure the performance in the presence of replication mechanisms, where identical instances of a component are running in parallel.

- *Service Time*: measures the time it takes for the system to deliver a service to the actor who requests it. It is useful to measure the time needed to upgrade the IDS, the time the IDS needs to respond to a query from the CCS, etc.

- *Instruction Path Length*: measures the number of machine code instructions required to execute a section of a computer program, providing information about the relative efficiency of a system. It helps assess the complexity versus the efficiency of the IDS modules.

- *Completion Time*: measures the amount of time required to perform and complete a given task. It is aligned with metrics such as response time and latency. It helps assess the suitability of the IDS for the CCS.

- *Channel Capacity*: refers to the maximum rate of information that can be transmitted reliably over a communications channel. It indicates whether the CCS's channels have sufficient capacity to include the IDS's traffic, or if additional dedicated resources are needed.

- *Performance per Watt*: is the rate of computation that can be delivered for each watt of energy consumed. A low ratio of performance per watt is an indicator of the suitability of a component for the CCS.

- *Relative Efficiency*: the relative efficiency of two procedures, known as the ratio of their efficiencies is frequently calculated as the comparison made between a given procedure and a notional "best possible" procedure.

- *Bandwidth*: is a measurement of the data communication resources available, expressed in bits per second or multiples of it. It is related to channel capacity metrics.

5.3 Correctness Metrics

This section discusses examples of the metrics we have identified related to *correctness*. These metrics are especially important for CCSs, since they help measure how good the IDS' detection process is into detecting the threatening events, and not confusing them with harmless system dynamics.

- *Accuracy*: is "*the degree of closeness of measurements of a quantity to that quantity's actual (true) value*" [13]. In IDSs, accuracy has to do with the rate of False Positives (FP) and False Negatives (FN) of the system, i.e., the IDS can only be considered accurate when its rate of FP and FN is minimum or negligible.

- *Precision*: is a characteristic inherent to the measurement system, statistically it is defined as the dispersion of quantitative data, regardless of its accuracy [26]. It is, therefore, the degree to which repeated measurements, taken in unchanged conditions, show the same results.

- *Currency*: in terms of correctness, currency is defined as the degree to which data remain current, i.e., not obsolete. It is also known as the *freshness of the data*[26].

In the same context of correctness, there are two specific metrics that are truly interesting in the case of an IDS: *sensitivity* and *specificity*. They are statistical measures related to both the accuracy and precision of a classification system, and are defined as follows [12]:

- *Sensitivity*: also "*true positive rate*", measures the proportion of actual positives correctly identified as such. Sensitivity shows how good a test actually is by calculating how often the test will correctly identify a positive.

- *Specificity*: measures the proportion of negatives that are correctly identified as such. This measurement shows how accurate the test is with false positives, and can be considered as the percentage of times a test will correctly identify a negative result.

5.4 Maintainability Metrics

This section presents those metrics related to *maintainability*. As we have discussed, those metrics have a strong relationship with the set of metrics in charge of assessing the availability of the system. In this set of metrics we find very specific metrics that are commonly used in ICT systems, and which can be applied directly to CCS or CIs.

- *Planned Maintenance*: refers to the preventive maintenance events that are programmed to verify the correct operation of a system. These planned maintenance times either for the CI or the IDS should be keep to a minimum and always ensure the continuous operation of the CCS.

- *Repair Time*: also "*Mean Time To Repair*" (MTTR_1) is the average time required to repair a system that has failed. MTTR_1 has to be kept to a minimum in critical systems, which implement redundancy and replication mechanisms to compensate the failure of a single component.

- *HW Costs*: refers to the average cost of replacing a determined component or system. HW costs for complex CCS are usually elevated, in the case of deploying an IDS with its own dedicated HW, it must provide high compatibility with the subjacent system, in order to provide a sustainable service for years.

- *SW Costs*: refers to the average cost that the replacement of given SW incurs. In the case of an IDS SW component, it is important to verify its compatibility and capability of providing a survivable service over time.

- *Restoration Time*: is the estimated time required for a system to be restored to its original operation, in the case of failure. This metric has an important impact on availability, thus CIs should have support mechanisms put in place to reduce its time and impact as much as possible.

- *Failures Over Time*: or failure rate is the frequency with which a system fails. When deploying a new component (e.g., an IDS), it is important to ensure that its failure rate is within an acceptable threshold for the CCS, and that its faults will not have an impact on the subjacent CI.

- *Maintenance Man-Hours*: refers to the manpower needed to maintain the system. In the case of IDS components, the evaluation of this metric depends on their deployment (dedicated HW or integrated SW), since the computation of time will be dependent (or not) on the infrastructure.

- *Upgrade Events*: refers to the estimated frequency of upgrade events needed in a system. The IDS component should not need frequent upgrades, in order to avoid generating too much traffic or demanding too many computation resources, in a critically constrained environment.

- *Recovery Time*: also "*Mean Time To Recovery*" (MTTR_2), refers to the average time a given system will take to recover from a failure. Similar to MTTR_1, it indicates the time lapsed before the system returns to its normal operation. It has to be kept under a given threshold, in order to avoid failures cascading to other dependent systems.

- *Mean Time Between Failures* (MTBF): is the predicted lapse of time that occurs between inherent failures of a system during operation. This metric gives an estimation on when the failures will occur.

- *Mean Downtime*: is the average time the system is not operative, due to maintenance or a failure. Any down time suffered by a CI is critical for interdependent systems, and for society. In the case of the IDSs deployed within CCS, it is vital that any faults occurring in the IDS do not cascade to the subjacent CS and the CI.

5.5 Dependability and Safety Metrics

Dependability metrics and *safety* metrics are really representative of CCSs. In this section we present an example of some of these dependability and safety metrics that can be useful to evaluate in CCSs. In the field of safety, we have found metrics that are mostly based on statistics that provide insight into the probable occurrence of certain events over time. Dependability metrics have a strong link to those of survivability, mostly related to availability, maintainability, etc. For a more comprehensive list of this type of metric, we refer the interested reader to the IEC 61508 Standard [1].

- *Mean Time To Unsafe Failure* (MTTUF): represents the average time that a system will operate safely before the occurrence of a failure that produces an unsafe system state [23]. MTTUF should be as high as possible, indicating that there are few probable unsafe failures for the whole system, and thus a robust dependable system.

- *Reliability*: as a function of time, or survivor function $R(t)$, is the probability of a system which does not fail in a determined time interval [23]. From $R(t)$ it is possible to compute the reliability of the whole system.

- *Stability*: metrics, such as *Mean Square Stability* (MSS) [30], refer to the equilibrium and stability properties of a system. Systems with high stability will suffer less faults and provide a better service.

- *Safety specific metrics*: compliance with metrics such as the safety design stability metric, or the safety requirements traceability metric [35] can ensure that the IDS component deployed within a CCS is a safe system.

- *Safety Integrity Level* (SIL): metrics that are available at IEC 61508 [1] allow developing systems observing the level of safety of the system. IDS solutions complying with the specifications of this standard will ensure that the SIL of the subsystem is adequate for the CCS.

- *Percent System Safety Hazards*: measures the safety hazards of the system against the system's hazards [35]. The lower this percentage is, the safer and more robust the CI is; and thus, its components (e.g., an IDS).

5.6 Security Metrics

The last set of metrics we have identified are those metrics related to *security*. These metrics are usually strongly linked to the specific context where the security is to be implemented. In our study, they are distributed among the other sets of metrics. According to the NISTIR 7564 [31], the security metrics are based on two aspects: *correctness* and *effectiveness*.

- *Correctness*: usually related to the process of the development of the system, and to the compliance of its operation with the expected behavior. Standardization, quality assurance or development guided by good practices are targeted to improve the correctness of the system, and therefore to improve their security.

- *Effectiveness*: measuring effectiveness is usually based on the security-enforcing mechanisms, determining how well they function and if the system shows signs of vulnerabilities. The aforementioned metrics can help evaluate the effectiveness of the system, and thus help gain insight into the security of the system.

5.7 Discussion

The main objective of this study is to provide a structured analysis which illustrates the steps to follow to

determine if an IDS solution is suitable for deployment within CCSs. Furthermore, with little modification, it is possible to extend this work in order to determine whether or not any given component is suitable for deployment within a CI. To this end, we have structured our analysis based on the guidelines of two main methodologies, the NFR Framework [19] and the GQM approach [54]. This study has been divided into three stages: *the analysis of requirements*, *the identification of satisficing techniques* and *the identification of representative metrics*.

The first stage has been developed in Section 2, where the CCS requirements and special constraints are identified. In Section 3 this analysis has been extended in order to discern the requirements imposed on an IDS solution we wish to deploy within CCS, always taking into account the previously analyzed CCS requirements. The requirements have been analyzed following the NFR Framework guidelines [19]. Here we found that the IDS is required to comply with numerous softgoals which constrain the features of the current IDS. The compliance with the NFRs provides, indicates an IDS that is suitable for deployment in critical environments such as CCSs.

These NFRs, in contrast to functional requirements, represent high-level characteristics and information about the system under study. In SW engineering, the capture and definition of requirements is an iterative process that is refined in each cycle. In our work, we have performed several iterations to determine the NFRs that are most suitable for a generic CI (see Figure 2), stopping our analysis at this point. If this process is continued, given a concrete CI (e.g., Smart Grid) and an instance of the IDS tool to evaluate, a further cycle of refinement focused on this concrete scenario would provide the final NFRs that define the needs of this particular infrastructure and IDS.

Our analysis, however, stops before this last cycle of specialization, remaining purely theoretical, since we would like for our study to remain as open and versatile as possible to be applied to the different types of CIs in existence. Once the NFRs have been refined according to a given concrete scenario, a validation process should be done to determine the completeness and validity of the NFRs selected. However, as previously stated, due to their abstract nature, NFRs are difficult to define and test, therefore, they are usually evaluated subjectively [19].

Validation of some NFRs can be done through theoretical approaches, e.g., network security requirements can be evaluated from a game-theory point of view, in the form of a game between attacker and defenders using the Nash Equilibria [44]. Or through an expert group interview process (Delphi method [37], checklists [46]) to determine the adequacy of the selected NFRs for the problem at hand, and to have access to feedback about the completeness of the requirement set.

Since we consider that the final refinement cycle of NFRs should be done taking into account the real characteristics of a CI, we provide our study as a guideline for future reference when studying and analyzing the requirements and constraints of concrete critical scenarios. Thus the final refinement cycle of the NFRs and, in consequence, the validation process of the set of requirements selected remain as future work.

The second stage of our study provides insights into the way to comply with the identified NFRs (see Section 4). We have outlined the *satisficing techniques* proposed to achieve and satisfice the softgoals identified in the first stage of the study. We have therefore described sets of techniques that help achieve certain goals for the IDS, e.g., standardization, testing and development guided by good practices are techniques that help the IDS be more robust and sustainable, hence helping the system achieve sustainability.

In an effort to concretize the results of a study based on abstract characteristics (NFRs), we use the GQM approach [54] to identify sets of metrics that could help analyze from a quantitative point of view the selected set of requirements. The metrics we have identified for our study are merely examples of important quantitative information that can be used to evaluate a critical system, and any component that we want to add to this system, especially to ensure that the newly added sub-system will not introduce new risks and threats into the CI. This part of our analysis is developed in Section 5.

We have separated our metrics into five large sets of metrics for the sake of clarity, however, as we have

shown, they can overlap. On the other hand, these five large sets have been characterized from a high abstraction level due to the scope of this research, which ends in a refinement cycle before the concretization of the scenario. In practice, it is necessary to formalize the exercise by considering, for example, the templates offered by ISO 27004 [3] or NIST SP 800-55 [2]. Both standards provide sufficient guidance to compute, through more tangible calculation, the quantitative level of a study resulting in concrete sets of well defined metrics. In this way, the refinement of the metrics offers the means to develop and use assessment measures to determine the effectiveness of an information system and its controls.

It is important to note, that any metric we want to apply to a given system (e.g., CIs, the cloud, the IoT) has to be validated in order to learn whether these metrics are suitable for this system's evaluation, i.e., if it makes sense to use these measurements in this scenario, and whether or not they are representative for what we want to validate. However, this part of the study needs the refined final set of NFRs that takes into account the actual CI and the concrete IDS solution, which compose the final scenario of application. In our study, we have not included the validation stage of our proposed metrics. For each of them, we have discussed and exemplified the usefulness of the measurements for our specific scenario, the IDS for CCS.

It is possible to tackle the validation of the metrics using two methods: through a formal process of validation, or through an expert group interview process, such as the Delphi method [37]. In such a complex scenario, where numerous actors and interdependencies are in place, we consider the most viable method of validation for these metrics is through the expert group interview process, a task that additionally can result in very valuable feedback for the world of academia and also for the CI's management. A supplementary study which could provide additional information in the validation process consists in testing the selected metrics within other non-critical environments.

The results of this experimentation would provide additional insight to the experts when reviewing the metrics for the given IDS within a CI. This information would provide the experts an overview of, for example, the performance of the IDS in general networks, indicating whether the IDS solution is a priori too costly for a critical setting or not. However, these experiments are difficult to set, and of course, removing the IDS from a critical scenario would only be relatively useful in terms of evaluating the metrics. This is because the requirements of the CI could constrain the application of the IDS too much, which would perform perfectly in a general environment. Thus, due to the need to refine and validate NFRs according to a determined scenario, and the need of concretize the metrics as previous steps, this last stage of the study also remains as future work.

6 Conclusions

In our work, we have performed an analysis of the requirements and constraints that CIs force on any security component that we want to include to the CCS of the infrastructure. We have particularly focused on the inclusion of IDS solutions, and the characteristics such a tool should have in terms of NFRs in order to be added to a critical scenario. To this end we have followed a three-stage analysis where we have defined the NFRs that characterize the IDS for CCSs, a set of satisficing techniques that helps tailor an IDS solution to the critical setting, and lastly several sets of metrics that quantitatively assess the suitability of the component for critical scenarios.

To the best of our knowledge, this is the first attempt to formalize from an SW engineering point of view the requirements (NFRs) that have to be considered when implementing IDS solutions for critical settings, while attempting to provide quantifiable insight about the adequacy of the solutions. We believe that this is a very useful approach to this problem, in an environment where legacy systems are continuously patched to improve their functionalities, usually without taking into account sound SW engineering design principles.

Our analysis stops at a theoretical point, where the actual scenario (concrete CI and IDS solution) have not yet been defined. This implies that our study is applicable to different settings, but also that fur-

ther refinement of the NFRs and metrics is needed to take into account the particularities of each scenario. Moreover, since the refinement cycles are not finished at this stage, the necessary validation procedures of NFRs and metrics have not been performed in our study, they remain as future work. Validation of NFRs and metrics are difficult, we believe a valid strategy would be the expert group interview processes, such as the Delphi method, providing experts with the sufficient insight into the concrete scenario through simplified simulations. From our point of view, our three-stage analysis and the outcome of this study as a model would help sustain the improvement of security within CIs by ensuring the introduction of efficient and compatible components within the CCSs.

Acknowledgment

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through the project PERSIST (TIN2013-41739-R), and by the Andalusian government through the project PISCIS (P10-TIC-06334). The first author has been funded by a FPI fellowship from the Junta de Andalucía through the project FISICCO (P11-TIC-07223). The second author has received funding from the Marie Curie COFUND programme "U-Mobility" co-financed by Universidad de Malaga, the EC FP7 under GA No. 246550 and the Spanish Ministry of Economy and Competitiveness (COFUND 2013-40259).

References

[1] IEC 61508 -3 - Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems, 1998.

[2] NIST SP 800-55 - Performance Measurement Guide for Information Security, 2008.

[3] ISO 27004 - Information Technology Security Techniques Information Security Management Measurement, 2009.

[4] Guide to Industrial Control Systems (ICS) Security, 2011.

[5] NERC CIP Compliance, 2011.

[6] Collins Concise English Dictionary. HarperCollins Publishers, 2013.

[7] Guidelines for Smart Grid Cybersecurity: Vol. 1, Smart Grid Cybersecurity Strategy, Architecture, and High-Level Requirements, 2013.

[8] C. Alcaraz, G. Fernandez, and F. Carvajal. Security Aspects of SCADA and DCS Environments. *Critical Infrastructure Protection*, pages 120–149, 2012.

[9] C. Alcaraz and J. Lopez. Wide-Area Situational Awareness for Critical Infrastructure Protection. *IEEE Computer*, 46(4):30–37, 2013.

[10] C. Alcaraz and S. Zeadally. Critical Control System Protection in the 21st Century. *Computer*, 46(10):74–83, 2013.

[11] Cristina Alcaraz and Javier Lopez. Analysis of Requirements for Critical Control Systems. *International Journal of Critical Infrastructure Protection*, 2012.

[12] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus AF Andersen, and Henrik Nielsen. Assessing the Accuracy of Prediction Algorithms for Classification: An Overview. *Bioinformatics*, 16(5):412–424, 2000.

[13] IEC BIPM, ILAC IFCC, IUPAP IUPAC, and OIML ISO. *International Vocabulary of Metrology, Basic and General Concepts and Associated Terms*. JCGM, 2008.

[14] SIG Bluetooth. Bluetooth Specification, 2007.

[15] André B Bondi. Characteristics of Scalability and their Impact on Performance. In *Proceedings of the 2nd international workshop on Software and performance*, pages 195–203. ACM, 2000.

[16] Joel Brenner. ISO 27001: Risk Management and Compliance. *RISK MANAGEMENT-NEW YORK-*, 54(1):24, 2007.

[17] Daniele Catteddu, Massimo Felici, Giles Hogben, Amy Holcroft, Eleni Kosta, Ronald Leenes, Christopher Millard, Maartje Niezen, David Nuñez, Nick Papanikolaou, et al. Towards a Model of Accountability for Cloud Computing Services. In *Intr. Workshop on Trustworthiness, Accountability and Forensics in the Cloud*, 2013.

[18] E. Chien and G. OGorman. The Nitro Attacks, Stealing Secrets from the Chemical Industry. *Symantec Security Response*, 2011.

[19] Lawrence Chung, BA Nixon, E Yu, and J Mylopoulos. Non-Functional Requirements in Software Engineering. 2000.

[20] European Commission. *COM(2009) 149 - Protecting Europe from Large Scale Cyber-Attacks and Disruptions: Enhancing Preparedness, Security and Resilience*. Publications Office, 2009.

[21] European Commission. *COM(2011) 163 - Achievements and Next Steps: Towards Global Cyber-Security*. Publications Office, 2011.

[22] Computer News. Chinese Hacking Team Caught Taking Over Decoy Water Plant. Online News, August 2013. Last Accessed May 2014.

[23] Todd A DeLong, D Todd Smith, and Barry W Johnson. Dependability Metrics to Assess Safety-Critical Systems. *IEEE Transanctions on Reliability*, 54(3):498–505, 2005.

[24] M. Dunn Cavelty and M. Suter. The Art of CIIP Strategy: Tacking Stock of Content and Processes. *Critical Infrastructure Protection*, 7130:15–38, 2012.

[25] ENISA. Analysis of Annual Incident Reports 2012. *Annual Incident Reports*, 13:1–30, 2012.

[26] Donald G Firesmith. Common Concepts Underlying Safety Security and Survivability Engineering. Technical report, DTIC Document, 2003.

[27] Terry Fleury, Himanshu Khurana, and Von Welch. Towards A Taxonomy Of Attacks Against Energy Control Systems. *Critical Infrastructure Protection II*, 290:71–85, 2009.

[28] Anne Geraci, Freny Katki, Louise McMonegal, Bennett Meyer, John Lane, Paul Wilson, Jane Radatz, Mary Yee, Hugh Porteous, and Fredrick Springsteel. IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries. 1991.

[29] Alan Gillies. *Software Quality: Theory and Management*. Lulu. com, 2011.

[30] Oscar R González, Jorge R Chávez-Fuentes, and W Steven Gray. Towards a Metric for the Assessment of Safety Critical Control Systems. In *Proc. 2008 AIAA Guidance, Navigation and Control Conference*, 2008.

[31] Wayne Jansen. NISTIR 7564. Directions in Security Metrics Research. *NIST. gov-Computer Security Division-Computer Security Resource Center*, April 2009.

[32] K. McLaughlin. Intrusion Detection for SCADA Systems - the Smart Grid Protection Against Cyber Attacks. US CERT, 2015.

[33] Balajee Kannan and Lynne E Parker. Fault-Tolerance Based Metrics for Evaluating System Performance in Multi-Robot Teams. In *Performance Metrics for Intelligent Systems Workshop*, 2006.

[34] Kaspersky Lab Expert. Duqu: Steal Everything, 2011. Last accessed, April 2014.

[35] S Phani Kumar, P Seetha Ramaiah, and V Khanaa. Identification of Software Safety Metrics for Building Safer Software based Critical Computing Systems. *International Journal of Computer Science and Application*, 2010.

[36] Hui Lin, Adam Slagell, Catello Di Martino, Zbigniew Kalbarczyk, and Ravishankar K Iyer. Adapting Bro into SCADA: Building a Specification-based Intrusion Detection System for the DNP3 Protocol. In *8th Cyber Security and Information Intelligence Research Workshop*, 2013.

[37] Harold A Linstone, Murray Turoff, et al. *The Delphi Method: Techniques and Applications*, volume 29. Addison-Wesley Reading, MA, 1975.

[38] Aleksandr Matrosov, Eugene Rodionov, David Harley, and Juraj Malcho. Stuxnet Under the Microscope. *ESET LLC*, 2010.

[39] John F Meyer and William H Sanders. Specification and Construction of Performability Models. In *2nd Intr. Workshop on Performability Modeling of Computer and Communication Systems*, pages 28–30, 1993.

[40] Modbus-IDA. Modbus Application Protocol Specification, 2006.

[41] National Institute of Standards and Technology (NIST). Framework for Improving Critical Infrastructure Cybersecurity, February 2014.

[42] Andrea Nicholson, S Webber, S Dyer, T Patel, and Helge Janicke. SCADA Security in the Light of Cyber-Warfare. *Computers & Security*, 31(4):418–436, 2012.

[43] David Nuñez, Carmen Fernandez-Gago, Siani Pearson, and Massimo Felici. A Metamodel for Measuring Accountability Attributes in the Cloud. In *IEEE 5th International Conference on Cloud Computing Technology and Science*, 2013.

[44] Vicky Papadopoulou and Andreas Gregoriades. Nonfunctional Requirements Validation Using Nash Equilibria. *SCIYO. COM*, page 41, 2010.

[45] Stig Petersen and Simon Carlsen. WirelessHART versus ISA100. 11a: The Format War Hits the Factory Floor. *IEEE Industrial Electronics Magazine*, 5:23–34, 2011.

[46] A Ananda Rao and Merugu Gopichand. Four Layered Approach to Non-Functional Requirements Analysis. *arXiv preprint arXiv:1201.6141*, 2012.

[47] J. Reeves, A. Ramaswamy, M. Locasto, S. Bratus, and S. Smith. Intrusion Detection for Resource-Constrained Embedded Control Systems in the Power Grid. *Intr. Journal of Critical Infrastructure Protection*, 2012.

[48] S.M. Rinaldi. Modeling and Simulating Critical Infrastructures and their Interdependencies. In *System sciences, 2004. Proceedings of the 37th annual Hawaii international conference on*, pages 8–pp. IEEE, 2004.

[49] Daniel P Siewiorek and Robert S Swarz. *Reliable Computer Systems: Design and Evaluation*, volume 3. AK Peters Massachusetts, 1998.

[50] Federal Standard. 1037C, Telecommunications: Glossary of Telecommunication Terms. *Institute for Telecommunications Sciences*, 7, 1996.

[51] M. Thompson. Mariposa Botnet Analysis. Technical report, Technical report, Defence Intelligence, 2009.

[52] US DHS ICS-CERT. ICS-Monitor Malware Infections in the Control Environment. US CERT, December 2012. Last accessed, April 2014.

[53] US DHS ICS-CERT. ICS-Monitor Incident Response Activity. National Cybersecurity and Communications Integration Center, April 2014. Last accessed, May 2014.

[54] Rini Van Solingen, Vic Basili, Gianluigi Caldiera, and H Dieter Rombach. Goal Question Metric (GQM) Approach. *Encyclopedia of Software Engineering*, 2002.

[55] Kenneth R Van Wyk. *Secure Coding: Principles and Practices*. " O'Reilly Media, Inc.", 2003.

[56] Martin Weik. *Computer Science and Communications Dictionary*. Kluwer Academic Pub, 2000.

[57] Z. Yu, J.J.P. Tsai, and T. Weigert. An Adaptive Automatically Tuning Intrusion Detection System. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 3(3):10, 2008.

[58] B. Zhu and S. Sastry. SCADA-Specific Intrusion Detection/Prevention Systems: A Survey and Taxonomy. In *Proceedings of the 1st Workshop on Secure Control Systems (SCS)*, 2010.