# Cyber Stealth Attacks in Critical Information Infrastructures

Lorena Cazorla, *Member, IEEE,* Cristina Alcaraz, *Member, IEEE,*
and Javier Lopez, *Senior Member, IEEE*[*][†]

June 23, 2016

## Abstract

Current Critical Infrastructures (CIs) are complex interconnected industrial systems that, in recent years, have incorporated information and communications technologies such as connection to the Internet and commercial off-the-shelf components. This makes them easier to operate and maintain, but exposes them to the threats and attacks that inundate conventional networks and systems. This paper contains a comprehensive study on the main stealth attacks that threaten CIs, with a special focus on Critical Information Infrastructures (CIIs). This type of attack is characterized by an adversary who is able to finely tune his actions to avoid detection while pursuing his objectives. To provide a complete analysis of the scope and potential dangers of stealth attacks we determine and analyze their stages and range, and we design a taxonomy to illustrate the threats to CIs, offering an overview of the applicable countermeasures against these attacks. From our analysis we understand that these types of attacks, due to the interdependent nature of CIs, pose a grave danger to critical systems where the threats can easily cascade down to the interconnected systems.

**Keywords:** Critical Infrastructures, Control Systems, Countermeasures, Detection and Protection, Stealth Attacks.

## 1 Introduction

Information and Communication Technologies (ICTs) have now become essential elements in our society since they offer significant improvements in efficiency, cost reduction and enhancing quality of life. Mobile computing technologies, embedded systems, smart devices, wireless communication and the growth of the Internet are becoming the major driving forces. These enable management of information from anywhere, at any time and anyway, allowing an easier implementation and quicker operation of the great majority of today's competitors' infrastructures and their services [1]. In fact, most of these physical facilities are highly interconnected to other national (and international) systems through communication systems, and managed through software-based systems, where the atomic data are not only the integral elements of the infrastructure itself but are also needed between infrastructures in order for them to function properly [1].

[*] This work has been partially supported by the EU FP7 project FACIES (HOME/2011/CIPS/AG/4000002115), 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). Additionally, 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 (COFUND2013-40259).

[†] L. Cazorla, C. Alcaraz and J. Lopez are with the Department of Computer Science, University of Malaga, Campus de Teatinos s/n, 29071, Malaga, Spain, e-mails: {lorena,alcaraz,jlm}@lcc.uma.es.

Critical Infrastructures (CIs) are interconnections of a set of systems and assets, whether physical or virtual [1], which are integral to the social, political, and economic life of a nation and its citizens. Examples of these infrastructures can be water treatment systems, energy generation and distribution systems, finance, transportation, etc. In policy terms, the European Union (EU) considers a CI to be "*an asset, system or part thereof located in Member States which is essential for the maintenance of vital societal functions, health, safety, security, economic or social well-being of people, and the disruption or destruction of which would have a significant impact in a Member State as a result of the failure to maintain those functions*" [2]. Similarly, the United States (US) government considers critical infrastructures as those "*systems and assets, whether physical or virtual, so vital to the United States that the incapacity or destruction of such systems and assets would have a debilitating impact on security, national economic security, national public health or safety, or any combination of those matters*" - extract from Law 107-56, Section 1016, entitled critical infrastructure protection act of 2001 [3].

Any protection put into place to safeguard CIs should focus on preserving not only the physical elements of the infrastructure but also and most importantly its virtual (cyber) elements, as a disruption of these assets may trigger the same damage as the disruption of physical components, putting the security and safety of these interconnected systems at risk. In order to guarantee that CIs operate continuously, they are monitored by control systems to ensure the correct performance of processes and operations. In the industry, these systems are known as Supervisory Control and Data Acquisition (SCADA), and they belong to the category of Industrial Control Systems (ICSs). SCADA systems are composed of hybrid integral systems in which a set of control processes is widely distributed over large geographic locations, but any information has to be centralized at a single point, the SCADA center. To this end, remote substations comprise smart collectors (field devices) capable of interpreting ingoing/outgoing traffic, of sending information to the SCADA center or executing control actions in the field. These devices, widely known as PLCs (Programmable Logic Controllers) or RTUs (Remote Terminal Units), are connected to sensors in charge of perceiving measurement values (e.g., pressure) or actuators to carry out an action.

These operational features mean that ICSs are also CIs in themselves [1], and, together with the rest of the cyber elements of CIs, constitute what is called a Critical Information Infrastructure (CII) (given their critical nature, in the remainder of this paper we will refer to them in general as CIs). Any physical or virtual disruption related to communication or control may have devastating consequences for the continuity of services and business. Government and industry entities are already announcing the importance of addressing aspects of cyber-defense in their respective critical sectors, where CIIs are in the sights of potential attackers [4, 5, 6].

## 1.1 Identified Cyber-Attacks to CIs

One of the most dangerous threats that CIs face are cyber-attacks, where adversaries can remotely perform malicious acts that may have a disastrous impact on the infrastructures. This, together with an increasing number of threats, faults and errors registered, have alerted institutions worldwide. There are annual reports published by the different governments through specific organizations such as the European Union Network and Information Security Agency (ENISA) [7] and the Industrial Control System Cyber Emergency Response Team (ICS-CERT) [8, 9, 10], reflecting the current situation and the severity of potential threats. The number of specific incidents apparently continues to grow, requiring a major effort to establish security and protection measures immediately.

ENISA's work on managing incidents [7] in conjunction with the National Regulatory Authorities (NRAs) of the 28 EU member states was established in 2012 thanks to Article 13a of the framework Directive (2009/140/EC) [11]. According to the two latest reports, the number of incidents caused by natural disasters, human error, malicious actions, system faults and third party faults, and registered in the different sectors has already reached significant numbers. The majority of them targeted communication networks (51 in 2011 and 79 in 2012) based on fixed telephony (e.g., VoIP over DSL, cable, etc.), fixed Internet (e.g., dial up, DSL, cable, etc.), mobile telephony (e.g., UMTS, GSM), mobile Internet (e.g., UMTS, GSM). With very similar goals, ICS-CERT via the Critical Infrastructure Information Act (the CII Act) of 2002 manages incidents from owner organizations of CIs.

According to ICS-CERT, the number of incidents became more noticeable in 2010, the year in which information technologies started to be well-known, in which active remote accesses (e.g., Internet connections, connection to sub-networks, use of wireless technologies) also started to be exploited. The power grid industry is leader in the number of detected incidents (18 in total), followed by nuclear, chemical and water management, which received between 8 and 15% of the threats. The majority of the incidents reported were related to SSH (Secure Shell), brute-force attacks, scanning and spear-phishing (2 out of 3 attacks) in the power grid with the aim being to acquire credentials or personal information. As we can appreciate, one of the most dangerous threats that CIs face are cyber-attacks, where adversaries can remotely perform malicious acts that may have a disastrous impact on the infrastructures [12]. This is especially true when these cyber attacks target CIs and the adversaries' objective is to remain unnoticed while pursuing their goals, and so we face *stealth attacks*, a sophisticated and potentially very dangerous type of cyber attack. Usually these attacks are launched by powerful adversaries with the objective of extracting sensitive or reconnaissance information without being noticed, to sometimes, afterwards, use this information to launch malicious attacks to cause disruptions to CIs. Some examples of these attacks, perpetrated in 2010 are:

- CIKR Mariposa [13]. Mariposa was a botnet, performing operations of denial of service attacks, e-mailing spam, personal information theft, modifications in the web-browser's searches, and other similar cyber-attacks.

- Stuxnet worm [14]. The first malware code designed specifically for engineering controllers (i.e., PLCs/RTUs). The worm, with the ability to infect numerous network devices without leaving evidence of the attack, was primarily focused on reaching and manipulating critical sections of a particular PLC of Siemens. The origin of the infection was traced back to the unsuitable use of personal media devices (USB drivers).

In 2011, 197 reports of incidents were received; the water sector, topping the list with 81 incidents. Many of the reported incidents were related to spear-phishing for illicitly obtaining security credentials or unauthorized access to restricted systems, as well as other relevant attacks such as:

- Night Dragon attack [15]. Attack reported by McAfee, which was based on a combination of a set of potential threats (e.g., social engineering) and malware (e.g., Trojans) to breach the security of corporate networks in charge of managing control systems.

- Nitro Attacks [16]. Sophisticated attack that involved several companies in the chemical sector, primarily private companies involved in research, development, and manufacture of chemicals and advanced materials. The attack aimed to collect confidential data, and infected machines in the order of 27% in the USA, 20% in Bangladesh, 14% in United Kingdom, 6% in Argentina, 4% in Singapore, 4% in China, Taiwan, Germany and Czech Republic; 2% in Hong Kong, India, Netherlands and Finland; 1% in South Korea, France, Russia, Japan, Sweden, Norway, and Canada.

- Duqu [17]. Virus considered to be a mutation of Stuxnet but without the ability to self-replicate. Despite this feature, Duqu is able to reveal private information, configurations and accesses and has a similar behavior to Flame, described below.

The number of incidents remained equally high in 2012 with 198 registered [9]. 41% of the threats targeted the energy sector and its control systems, and the water sector witnessed the second highest number of incidents with 15% of the threats. The report of 2012 also noted two important aspects. On the one hand, systems connected to the Internet and protected through weak or default credentials were those that most received the most common attacks on the Internet; and on the other hand, more and more the water sector was becoming a specific target for attackers. This report presented some specific examples, such as the case of the water utility located in Springfield (Curran-Gardner public water district), which was attacked from an IP address located in Russia without leaving any evidence of this intrusion in the SCADA system. Another example of a cyber-attack is:

- Flame [18]. Worm originally designed to open back doors, infect and modify functions, in addition to stealing confidential data, destroying information or recording conversations.

In 2013, ICS-CERT received roughly 200 incidents [10]. The highest percentage of incidents was found to be in the energy sector (53%) followed by critical manufacturing (17%). The majority of these incidents were related to cyber-attacks such as watering hole attacks (with the intention of attacking those strategic points (e.g., servers, websites) that are frequently visited by targets), SQL (Structured Query Language) injection, and spear-phishing attacks. In the first quarter of 2014, the ICS-CERT reported attacks mainly on the energy and water sectors, followed by the transportation sector, where the main vulnerabilities targeted were weaknesses and flaws in the design of the systems [19].

Through this review of recent attacks, we can readily identify the real danger behind stealthy adversaries, and the need to understand them better in order to prevent attacks and counteract them, especially in critical contexts. The concept of stealth attacks was introduced for conventional networks by M. Jakobsson et al. in 2003 [20]. They were described in the literature as those attacks in which the cost and visibility of the attacker have to be minimized. Cyber stealth attacks "*allow a skilled but not very powerful attacker to target communication networks in a way that makes it unlikely that he gets traced and caught*" [20]. This type of adversary has proliferated in recent years targeting critical systems, since the first known high-scale stealthy attacks on CIs (Mariposa, Stuxnet).

These incidents showed the characteristics and sophisticated capabilities of these types of attacks, and proved that it is possible to adapt stealthy techniques used for conventional networks to threaten critical scenarios. However, besides these highly complex attacks, we understand that it is also possible to take this same knowledge on stealth attacks from general-purpose networks to implement stealthy cyber attacks on CIs in a less complex manner, but with potential, equally harmful results. CIs, especially ICSs, have, over the years, added ICTs to their infrastructures, but they have not incorporated sufficient security mechanisms to protect them [1], so they have inherited many threats and weaknesses from traditional networks. This lack of strong security mechanisms opens the door to multiple types of cyber attacks against CIs, one of the most powerful being stealthy attacks. Our work is, to the best of our knowledge, the first attempt to undertake the analysis of this kind of stealth attack in CIs.

The remainder of this paper studies all aspects of these attacks in relation to CIs. Section 2 presents the stages of a stealth attack. Section 3 describes the AICAn taxonomy. Section 4 provides a review and classification of the different types of cyber stealth attacks that can be launched against CIs. Section 5 reviews the countermeasures and prevention techniques available against stealth attacks. In Section 6 we discuss the effects of stealth attacks on the AICAn. Finally, in Section 7, conclusions and future work are outlined.

## 2 Stages of a Stealth Attack

Stealth attacks, as in any kind of (cyber) attack, are composed of three main stages or phases that have to be fulfilled so as to achieve the adversary's objectives, namely: (i) *stealthiness of the communication*, (ii) *stealthiness of the execution*, (iii) *stealthiness of the propagation*. Figure 1 illustrates these stages, where

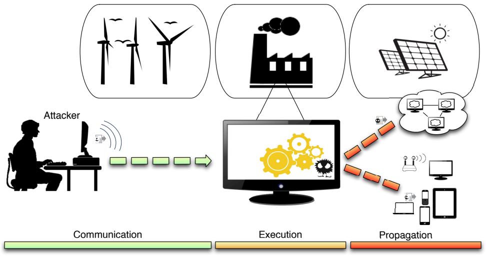

Figure 1: Stages of a stealth attack

each phase is based on the preceding one. Every single attack is different in nature, and can comprise one or more of the three stages mentioned, always following the established order: first the communication phase, then the execution of the attack and lastly its propagation.

In the specific case of stealthy attacks, they follow these three phases, but the adversary remains undetected while pursuing his objective. However, it is important to note that the success of a stealth attack depends on the intention of the adversary, since his objective might be to achieve only one or two of the stages; e.g., the attacker aims to scan the ports of a system unnoticed, to determine which ones are open, and he does not care about being detected afterwards. In this case, therefore, by succeeding in the first stage of development of the stealth attack, the adversary fulfills his tasks.

Figure 1 represents an external cyber attacker, that transmits the attack to the CI, mainly targeting the communication networks and the system's critical nodes. This first phase of the attack is the least intrusive stage of the attack, since sometimes the only aim of the adversary is to achieve this phase undetected. In a second step, the adversary achieves the execution of the attack within the CI itself, this execution could result in vast damage or compromised information, since the adversary remains unnoticed while extracting information or damaging the equipment. The last stage of the attack represented in the figure, is the propagation of the attack to other nodes or to other connected infrastructures. The successful achievement of this step reveals a highly sophisticated attack, launched by skilled adversaries, with good knowledge of the victim system.

However, the criticality of the attack depends on the intention of the adversary, i.e., it is not the same to subtract information as to cause irreparable damage to the CIs. Additionally, as we have mentioned, each attack achieves one or several of the aforementioned stages according to the objectives of the adversary, i.e., in the case of industrial spies, they may only want to extract information without being discovered, and without causing any harm to the CIs. In Section 4, we provide a review of the stealth attacks against CIs, indicating the scope of each attack and the intentions of the adversaries.

# 3 AICAn Taxonomy

In the current literature, there is a wide variety of attack taxonomies and studies on cyber-security for both conventional and critical systems [21, 22, 23, 24]. However, it is important to stress that the majority of these studies do not consider new ways to address recent security problems. For example, Lipson showed in [25] a chronological study of threats carried out since 1980, and most of these threats are still present in modern information systems. This means that the area of security remains open, where more attention needs to be paid by the scientific community, and more specifically, when ICTs are being adopted in critical contexts.

To complement these studies on stealth attacks in critical scenarios, we extend the taxonomy proposed in [21], based on the security properties *Availability* (A), *Integrity* (I) and *Confidentiality* (C), AIC. To this end, we consider the attack taxonomies given by the ENISA in [26], F. Skopik et al. in [27] and the security framework for ROLL (Routing Over Low Power and Lossy Networks) specified by IETF (Internet Engineering Task Force) in [28].

The motivation behind the extension of the taxonomy based on AIC is the fact that besides being attacked, there are multiple types of anomalies appearing all the time within a critical infrastructure, therefore it is necessary to include certain indicators of anomalies to study the effect they alone have, and when (stealth) attacks are present. In the critical infrastructures field, it is, for example, necessary to discern between infrastructural anomalies and control anomalies:

- *Infrastructural Anomalies* (InfAn), related to physical events (e.g., pressure, flow, radiation) relative to the critical infrastructure itself and its components.

- *Control Anomalies* (CAn), corresponding to any unexpected alteration in the control of critical systems caused by Hardware (HW) and Software (SW) faults, errors or intrusion.

- *Intrusion anomalies* (IntrAn), associated with those malicious actions within the physical infrastructure or its control systems that cause unforeseen incidents.

- Combinations of the above. For example, an IntrAn can trigger a CAn, or vice-versa; or an IntrAn can produce abnormal changes in the readings values causing an InfAn (e.g., a stealth attack).

Given the importance of taking into account the anomalies when detecting intrusion or security gaps, we therefore propose to include a new class within the taxonomy given in [21], denoted here as AICAn and depicted in Figure 2. This new taxonomy comprises the following threat classes:

Most of the stealthy attacks base their strategies on conventional threats against the availability (A), integrity (I) and confidentiality (C) of critical data, its hardware/software resources and user's information (credentials and roles) [21]. However, as mentioned above, adversaries can also take advantage of existing vulnerabilities or anomalies to attack the critical system's AIC. For this reason, we propose for this paper a new taxonomy based on AIC plus anomalies, denominated here as AICAn, where, for each category, we identify a subset of threats according to the their nature and type:

- Availability: these threats aim to reduce, as much as possible, the accessibility and disposition of resources and information of the system, infringing upon some of the aforementioned SCADA security requirements. These threats can be carried out through a set of actions related to denial of service/distributed denial of service (DoS/DDoS), or physical attacks. Depending on the intentions of the attacker (exhaustion of assets, operational disruption or reduction of functionalities), we identify two subcategories within the availability property: *Resource Availability* (RA) and *Information Availability* (IA).

- Integrity: correspond to those vulnerabilities exploited to distort critical sections of a node/object or its messages, such as an overflow or implementation attack. Availability attacks may also have a repercussion on the integrity of a node and its assets, thereby violating one of the essential security requirements of a SCADA system. We consider two sub-types of integrity threats: *Resource Integrity* (RI), and *Information Integrity* (II). Additionally, if an adversary is capable of manipulating security credentials and roles so as to impersonate the users or the administrator of the system identities, a threat to the *User Integrity* (UI) and *Host-User Integrity* (HUI) can arise.

- Confidentiality: concerns the adversary's ability to eavesdrop or deliberately expose sensitive information belonging to configurations or critical data, i.e., information on operational control (commands, alarms or measurements) or information associated with connectivity, routing tables, nodes location, existing vulnerabilities, etc. This allows the adversary to carry out subsequent attacks [29], and thus we have to differentiate between *Resource Confidentiality* (RC) and *Information Confidentiality* (IC) in our analysis.

- Anomalies: an anomaly is defined as something that deviates from the standard or common. If the system presents a specific set of rules/patterns of behavior, an anomaly would therefore be the introduction of new unknown patterns, or the breach of such rules/patterns. As we have stated, it iso possible to identify three anomaly categories: *Infrastructural*

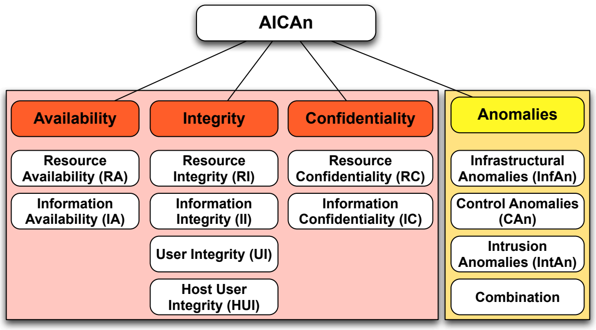
AICAn
Availability
Integrity
Confidentiality
Anomalies
Resource Availability (RA)
Information Availability (IA)
Resource Integrity (RI)
Information Integrity (II)
User Integrity (UI)
Host User Integrity (HUI)
Resource Confidentiality (RC)
Information Confidentiality (IC)
Infrastructural Anomalies (InfAn)
Control Anomalies (CAn)
Intrusion Anomalies (IntAn)
Combination

Figure 2: AICAn taxonomy

*Anomaly* (InfAn), *Control Anomaly* (CAn), *Intrusion Anomaly* (IntrAn), and any combination of them.

All of these threats, especially those related to availability, integrity, confidentiality and intrusion anomalies, can be the origin of the distortion or corruption of assets, destruction of assets, denial of service, information disclosure and eavesdropping [22]. To form the AICAn taxonomy, however, we have to consider the possibility that unforeseen events (anomalies) can also become potential threats, which may open up new security gaps that can be exploited through stealth attacks; or that these events may stem from these attacks as well.

Stealth attacks, as described above, happen in a scenario where the objective of the adversary is not only to successfully perform the attack, but also to do so with a minimal effort, and in a way that hides his existence and activities to the largest possible extent. It is therefore important to identify the methods or weapons employed by the adversaries, which are closely related to the AICAn taxonomy [20]. Firstly, *impersonation*, which attacks the integrity (I) of the system, and consists in introducing packets with stated originators different from the real originators, which can be performed by spoofing IP addresses or by using communication frequencies that have been assigned to others. This is always supposing that the originator of the impersonation is an honest party.

Secondly, the *lies* weapon threatens the integrity (I) of the system, where the attacker propagates incorrect information, such as incorrect routing tables. Lastly, *overloading*, which threatens the availability (A) of the system, is a technique that has been proposed as a possible technique to mount DoS attacks, where the attacker injects invalid messages (message with violated integrity, replayed message or junk message). Technically, overloading is difficult to implement as a stealth attack, nevertheless, it can be quite effective in controlling operations such as route discovery or routing table update.

## 4 Classification of Cyber Stealth Attacks

Stealth attacks can be categorized according to several parameters. In our review of the literature, we find there are five types of stealth attacks depending on the objective of the adversary: (i) *disconnection and goodput reduction* [20], (ii) *active eavesdropping* [20], (iii) *scanning and probing* [30], (iv) *covert and side channel exploitation* [31, 32] [33], and (v) *code injection* [34, 33].

### 4.1 Disconnection and Goodput Reduction

In this first type of attack, the adversary wishes to disconnect the network (a partition of the network or isolate particular nodes) or degrade its operation (its goodput). Here, the adversary does not need to control the nodes, but only needs to make them inadvertently get involved in the attack by tricking them into modifying their behavior (e.g., modifying their routing tables incorrectly) to cause disruption. This attack implies a threat to the availability and sometimes the integrity of the victim system, constituting a risk to the IA, RA and RI according to the AICAn taxonomy; also, these threats indicate possible anomalies in the infrastructure regarding confidentiality (CAn) and due to the intrusion itself (IntrAn).

An attacker may disconnect a victim in several ways, e.g, M. Jakobsson et al. [20] provide different variations of the disconnection attack in wireless mobile networks, where the power consumption of the devices is critical to their operation:

- *Disconnection due to the unreachability of the nodes*: the adversary disconnects the victim nodes making the other nodes believe they are unreachable (attack against the IA, RA). This attack has several variations, implementing different degrees of stealthiness by using these methods:
  - The adversary routes considerable amounts of traffic through the victim until it runs out of power. This attack is based on the cost that sending messages has in terms of the battery power consumed.
  - The adversary attacks all the known neighbors of the victim node making their batteries run out of energy. This causes disconnection as well, but it can be overcome by moving into another neighborhood.
  - The adversary routes traffic to the victim node and its neighbors, causing a portion of the messages to be dropped due to insufficient bandwidth. This version of the attack takes into account the response of a router trying to reach a node several times, and then concluding that the node is disconnected.
- *Removal of an entry in the routing table*: here, the adversary disconnects a node removing its entry in the routing tables of the network, making the victim node "disappear" (attack against the IA, RA, RI). It is also possible that the attacker forges the route discovery messages to convince the source node and other legitimate nodes that no route to the victim can be found.
- *Goodput reduction*: the disconnection of one or more nodes usually implies a reduction of the goodput of a network. The adversary can disconnect a large number of nodes, corrupt a large enough number of routing tables to increase the de facto traffic through each node, or degrade the power supplies of a large enough portion of the routers, virtually disabling them. This constitutes attacks against the IA, RA and RI of the AICAn taxonomy.

Stealthy implementation of these procedures allows a low exposure of the adversary during the attack. What we have previously discussed are stealth versions of the common DDoS attack [20]. Regarding stealth DoS, there are several ways of performing this type of attack, for example, M. Jakobsson et al. provided an overview on how it can be carried out against different types of wireless networks in [20, 35].

## 4.2 Active Eavesdropping

This second type of stealth attack comprises the modification of the routing information to hijack traffic from and to selected victim nodes [20]. Here the attacker can perform traffic analysis and selective filtering of packets without the knowledge of the victim, to actively eavesdrop on him and modify his behavior, e.g., making nodes of the network "disappear" and detouring the network traffic through compromised nodes. This attack usually threatens the confidentiality of the system (IC, RC), thus we usually see the activation of the indicator CAn in the presence of eavesdropping attacks. Sometimes it also introduces risks to the availability or integrity (IA, RI).

The simplest way to achieve this attack is to corrupt the routing tables of nodes on the path between a victim and the sender/receiver. The attacker can remove correct routing table entries and add incorrect ones in order to force rerouting [20]:

- For *incoming traffic*, i.e., packets going into the victim, the attacker forces all incoming traffic to be sent through a node he has previously corrupted. To receive traffic only from certain sources, the attacker can selectively tamper with the routing tables, allowing only those entries that are useful to the attacker to remain correct.
- For *outgoing traffic*, i.e., packets sent from the victim to another node in the network, the attacker modifies the routing tables of the victim and/or the routing tables of the nodes close to

the victim forcing traffic to be rerouted through a corrupted node.

To corrupt the routing tables of the network, the adversary can use the very tools of the routing protocols. The attacker can propagate routing tables where the entries are modified; another option is to make use of the route discovery process of the network to include new routes or report route error, in order to tamper with the routing tables.

### 4.3 Scanning and Probing

Scanning is a method for discovering exploitable communication channels. It implies a previous reconnaissance of the network or a particular host [30]. The objective of port scanning is to determine which ports of the system are open, and through them obtain valuable information; e.g., which services are running on the system that are available to the attacker, what services of the operating system are being used, parameters such as IP and MAC addresses, topological information, etc. The idea is to probe as many listeners as possible, and keep track of the ones that are receptive or useful to your particular need [36].

These types of attacks are the least dangerous in terms of threats to the AICAn of the system, therefore threatening the correct operation of the system, but they present a threat to the confidentiality of the resources (RC) and they can serve as a precursor to more powerful and disruptive attacks, thus they need to be always considered and monitored. C. Yin et al. [37] state that the port-scan is at the beginning of the process of intrusion, and there are varied techniques to scan the system, e.g., stealth scan, fragmentation scan, changes of scan order, slow scan, randomizing inter-probe timing, scan with forged address or distributed scan. G. Lyon states in [36] that several techniques have been developed over time for surveying the protocols and ports on which a target machine is listening.

During a normal TCP connection, the source initiates the connection by sending a SYN (synchronize) packet to a port on the destination system. If a service is listening on that port, the service responds with a SYN/ACK (synchronize/ acknowledgment) packet. The client initiating the connection then responds with an ACK packet, and the connection is established. If the destination host is not waiting for a connection on the specified port, it responds with an RST (reset) packet. Most system logs do not log completed connections until the final ACK packet is received from the source [38].

To scan the system, this standard behavior is modified in different ways. Here, we describe some of these variations, in order of degree of stealthiness:

- *TCP connect() scanning*: the most basic form of TCP scanning, where the connect() system call of the operating system is used to open a connection to every interesting port on a machine. If the port is listening, the connect() call will succeed; otherwise the port is unreachable. This technique is fast and does not need any super user permissions, however, it is easily detectable and filterable, since the target node will log the connection and error messages when the adversary initiates the connection to the port service and immediately shuts it down.

- *TCP SYN scanning*: sometimes referred to as half-open scanning, since the TCP connection is not fully opened. The attacker sends a SYN packet, as it would happen to open a real connection, and waits for a response. The response can be a SYN/ACK packet if the port is listening, or a RST packet if the port is not listening. When the adversary receives a SYN/ACK packet, he sends a RST packet to tear down the connection. This attack needs super user permissions to build the SYN packets. The advantage of this attack is that systems do not usually log these kinds of attempts at communication; however, it is easily detectable if the firewalls are configured to detect SYN packets targeting restricted ports.

- *TCP FIN scanning*: increasing the level of stealthiness, the FIN (finalize) scanning technique [39] is based on the idea that closed ports respond to FIN packets with RST packets, while open ports ignore them. The FIN scan's stealth packets are unusual because they are sent to a device without first going through the normal

TCP handshaking. Nevertheless, there are some systems that are not vulnerable to this type of scan, because they respond to a FIN packet with an RST packet regardless of the current state of the port.

- *Christmas scan*: this type of scanning technique sends a TCP packet to a remote device with the SYN, FIN, ACK flags set. This is colloquially called a Christmas tree scan because of the alternating bits turned on and off in the flags byte (00101001), like the lights of a Christmas tree. Similar to the FIN scan, a closed port responds to this packet with an RST packet, and an open port ignores it.

- *Null scan*: the adversary creates a TCP packet with all the TCP flags off. This is a type of packet that never occurs in the real world. As in the previous two situations, an open port receiving this kind of packet ignores it, and a closed port responds with an RST packet.

These last three attacks are denominated stealth scan attacks [38], because they do not usually generate a log entry on the scanned host, and they allow an attacker to determine which ports are open on a target node, without being detected by the host operating system. Many attacks in the literature use stealth scans and probes as a first stage in reconnaissance to gain insight into the characteristics of the system, to later trigger a more sophisticated and informed attack.

I. Dainotti et al. [40] provide a study on stealth scans carried out by botnets, in a coordinated and distributed infrastructure, targeting critical voice communications infrastructures. This scan attack is called sipscan and probes each target IP address with two packets: (1) an UDP packet sent to the port 5060 carrying a session initiation protocol (SIP) header, and (2) a TCP SYN packet that attempts to open a connection on port 80. This attack is usually the first step in a more sophisticated attack, where the attacker sends malware that infects the nodes of the network to make them act to profit the adversary.

## 4.4 Covert and Side Channel Exploitation

A side channel attack is very powerful in practice [41]. Here the adversary measures side channel information and is able to recover very sensitive information about the functional behavior of a system, without utilizing its dedicated interface [31]. Side channel attacks exploit the external manifestations of the system, like processing time, power consumption and electromagnetic emission to identify the internal computations [32]. This type of attack represents a threat to the confidentiality of the resource (RC) and in the particular case of side channel attacks that induce faults in the system, the anomalies indicators that are activated are InfAn, CAn and the IntrAn.

The aim of side channel attacks is usually to identify a “leakage” or source of secret data (side-channel analysis), where the attacker can use the results of this information to identify weaknesses in the system. The different types of side channel attacks are: timing attacks, power analysis attacks, electromagnetic analysis attacks, fault induction attacks, optical side channel attacks, and traffic analysis [31]:

- *Timing attack*: the adversary analyzes the running time of the system in order to extract knowledge about the type of computations and the parameters used. The main targets of timing attacks are cryptographic systems.

- *Power analysis attack*: here, the adversary measures the power consumption of the system to extract knowledge about it. There are several types of power analysis attacks, mainly targeting cryptosystems, which employ different methodologies and levels of sophistication to obtain the information; e.g., simple power analysis, differential power analysis or correlation power analysis.

- *Electromagnetic analysis attack*: this kind of attack implies the analysis of the electromagnetic variations of a system by the adversary. There are several types of electromagnetic attack which target very different kinds of systems; however, this kind of attack is most often designed for constrained cryptosystems.

- *Fault induction attacks*: the induction of faults in the system can result in erroneous operations that can shed some potentially valuable information about its operation.
- *Optical side channel attacks*: here the adversary is capable of retrieving information via the light emission from the monitors and LEDs (light-emitting diode) of a system. There are different kinds of displays and LEDs, and the information that can be extracted from them is varied.
- *Traffic analysis attacks*: this kind of attack provides the adversary with information about the topology of the network, through the analysis of the traffic flows.

A variation of a side channel attack is the use of *covert channels* [33], where there is a hidden connection between the transmitter and the receiver, thus there is a chance to extract or send valuable information through the channel without the system noticing. There are two types of covert channels: (i) communicating extra information to a host, and (ii) hiding the fact that the communication to a host exists [34]. Covert channels usually take advantage of places where random data is naturally transmitted, thus the encrypted information can be transmitted replacing this data. This technique is sometimes referred to as piggybacking [42], where the messages are hidden within the regular messages of the network. There are many varied ways of implementing covert channels, and the targets are multiple. However the commonality behind this type of attack is its dangerousness and its potential to induce multiple threats within the victim systems, targeting most AICAn variables. According to N. Tomar et al. in [33] the following vulnerabilities that can favor covert channels:

- *Virus and malware*: software such as viruses and Trojan horses can be introduced inside the victim's system, to perform activities such as capturing packets and injecting scripts into the victim's programs.
- *Important resources*: resources such as system files, disks, RAM, etc. are valuable to attackers, and vulnerable due to their criticality in the normal operation of the system.
- *Data sensitivity*: within the system coexist data with different degrees of sensitivity. The most sensitive data is the most interesting information to attackers, and thus the target of covert channel exploitation.
- *Vulnerable protocols*: several protocols implemented by CIs that are not properly secured, or they do not implement security mechanisms such as authentication (e.g., Modbus [43]). To protect the systems against covert channels attacks, it is important to strengthen their security.
- *Design robustness*: covert channels take advantage of principally two vulnerabilities of the system: design oversight, and weaknesses due to the system's design. Design oversight-derived vulnerabilities are unintentional and unforeseen, however weaknesses inherent in the system's characteristics are strong obstacles to the security of the system and provides a way of access for covert channels.
- *Packet headers*: as seen in [34], covert channels can be embedded in TCP and IP header fields, with very different objectives and functionalities.
- *Super user permissions*: an attacker can take advantage of an unintentional, careless or default assignment of super user permissions to processes, to create a covert channel.
- *Handshake trials*: communication protocols usually have handshake procedures to start the transmission of information. Some attackers use handshake trials to transfer information in an unnoticed way.
- *Public resources*: resources that are shared in the network, such as printers or hard drive disks, are vulnerable to attacks if they are not protected by security mechanisms.

- *Authentication*: as we have previously seen, some protocols and systems lack adequate authentication mechanisms, such as Modbus, DNP3 or ICCP [43]. This adds multiple vulnerabilities to the unprotected systems, among them, the use of covert channels by an attacker.

Most of these attacks introduce, as we have described, a wide range of AICAn threats, e.g., the attacks that exploit flaws or use malware are capable of threatening the availability (IA, RA) and the integrity of the system (II, RI), as well as compromising the integrity of the user and the host (UI, HUI); the confidentiality of the system can be also compromised (IC, RC), activating the CAn and sometimes the IntrAn indicators.

## 4.5 Code Injection

A *code injection-based attack* consists in introducing or “injecting” a tainted or illegitimate code within a computer program, in order to alter its outputs or change its course of execution [44], and cause different effects, e.g., compromise sensitive data, execute malware, etc. These attacks pose a threat to multiple variables of the AICAn taxonomy, allowing the adversary to interfere with the AIC of the system, and insert CAn and IntrAn anomalies.

Depending on the targeted system’s characteristics and the degree of stealthiness intended in the attack, it can be performed using two main channels: *system vulnerabilities*, and *malware infection* (i.e., infecting the system with malware, virus or Trojan horses). Injections exploiting design vulnerabilities appear when system designers and developers make incorrect assumptions about the use of the system’s services, e.g., (i) the input characters of a field will always be the regular and required ones (e.g., no colons, numbers or quotation marks are expected); (ii) the input of a field will never exceed a pre-determined size; (iii) the numeric values introduced as inputs in a system will always stay between the upper and lower bounds expected; (iv) the client supplied values cannot be modified by the adversary (e.g., cookies poisoning attack [45]); (v) it is safe to take pointers or array indexes from the requested input; (vi) the input will never provide false information or fake values (e.g., the size of a file); etc. [46].

On the other hand, malware can also pose successful and potentially harmful threats when implementing injection attacks (e.g., Stuxnet [14], Duqu [17], etc.). There are multiple types of code injections, and several ways of classifying them. We have decided to categorize them according to the target they are designed to inject, thus these attacks can be roughly summarized into the following four categories:

- *Database injection*: are the injections performed by the adversary to corrupt the databases of the system, or retrieve valuable information from it, without having the proper credentials to access the system. Database injections compromise the AIC of the system (IA, RA, II, RI, IC and RC) and activates the CAn and IntrAn indicators of anomalies. The most well-known attacks in this category are the SQL-injection attacks [35].

- *Command injection*: also known as shell injection attacks [47], can occur when the the system allows software to execute a command line. Therefore the attacker can make the system execute commands or functions to carry out unwanted tasks. This type of attack allows the attacker to threaten the AIC of the system (IA, RA, II, RI, IC and RC) and of the user (UI, HUI), in addition to introducing the CAn and IntrAn anomalies.

- *Website injection*: is the set of attacks that take advantage of flaws existing within websites, browsers or web applications that allow the adversary to introduce code and execute unwanted actions in an otherwise trusted environment (threatens AICAn like the previous attack). The most well-known attack within this category is *cross-site scripting* (XSS), which occurs when the adversary exploits a flaw detected on a web server to inject some code in the server, for his own use [48, 49]. Related attacks are the *Cross-Site Request Forgery* (CSRF) [50], where the adversary forces the victim to execute unwanted actions on a web application in which he is currently authenticated; or the *Server-Side*

*Includes* (SSI) Injection [51], where the attacker introduces scripts in HTML pages or executes arbitrary codes remotely.

- *OS injection*: comprise those attacks that target the stack, heap, pointers or internal variables determining the behavior of the system. Code injection at this level can make the operative system (OS) execute unwanted routines and procedures, inserted in the OS's running processes through the modification of the system variables to point to external code introduced by the attacker [52]. They threaten the AICAn as does the previous attack.

In a critical context, these attacks can target different parts of the infrastructure, namely the *corporate networks*, the *SCADA center* and the *remote substations*. The first are based on local area networks connected to the SCADA to gain access to critical data streams on SCADA servers, and are vulnerable to injections designed for conventional networks. The SCADA center is in charge of constantly monitoring the infrastructures through distributed substations. The remote substations are control subnetworks based on field devices (sensors, actuators) and communication interfaces (PLCs, gateways, etc.) in charge of sending sensorial measurements to the SCADA center. The SCADA center and the remote substations are vulnerable to injections specifically designed to target industrial devices and protocols.

Code injection attacks usually tend to implement some degree of stealthiness, since the adversary usually aims to retrieve valuable information from the system, or to force a desired (malicious) behavior without the end user being alerted. The actual level of stealthiness depends on the objective of the attacker, and also on the way the injection is tailored to the targeted system. According to Figure 1, it is possible to evaluate the degree of stealthiness of a given attack (in the communication, execution and transmission phases) and assess the potential threats and risks it poses.

## 4.6 Assessment of Stealthiness

We can differentiate two main kinds of behaviors in cyber stealth attacks: the *reconnaissance based attacks* and the *attacks with disruptive or tampering objectives*. These two main groups differ in the threats they pose to the correct operation of the CIs in terms of the AICAn taxonomy. Attacks with reconnaissance objectives, e.g., scanning and probing, or side channel attacks, are characterized by an adversary who tries to gather as much information as possible from the victim system, without being discovered in the communication phase (see Figure 1). In the case of this type of adversary behavior, the properties of the AICAn that are affected are usually related to the confidentiality, specifically the confidentiality of the resources (RC). In some of the cases, the attack is capable of retrieving certain information from the system, thus the IC property of the AICAn is compromised.

Some of the reconnaissance attacks might cause disruptions in the victim system, when the attacker intentionally induces faults to obtain information; in this case, the availability of the system can be affected, i.e., the IA and RA properties of the AICAn taxonomy; and the indicators of anomalies InfAn, CAn and IntrAn could be activated. Let us take a simple example, the *TCP connect() scanning* attack, where the attacker probes the ports of the system in search of useful open ports. This attack does not cause any disruption to the victim system, however the adversary is able to extract information about it, using just the communication phase of the attack to his own benefit. The information discovered in the reconnaissance attacks can be used by the adversary to launch more sophisticated attacks in a later step, using the knowledge acquired in the reconnaissance. The level of stealthiness achieved by this first group of attacks is determined by the stealthiness of its communication phase; i.e., whenever the adversary implements the attack in such a way that the victim system's warning mechanisms are not triggered by the reconnaissance actions, the attack can be categorized as stealthy.

Our second category of attacks, those with disruptive or tampering objectives, are characterized by an adversary who tries to achieve all the phases of the attack, i.e., communication, execution and sometimes propagation, stealthily. These attacks are much more complex, requiring highly skilled and informed attackers, capable of communicating with the system and executing the attack and if desired, propagating it to infect other components or target systems. Due to the possibilities they offer to the attacker, they are very dangerous to the victim system in terms of AICAn, because they can potentially disrupt all the AIC properties of the system and trigger all the different types of anomalies. The most representative attacks in this category are covert channel attacks and code injections.

To evaluate the level of stealthiness of a given attack it is necessary to evaluate each phase of the attack in order to determine if all of them are stealthy, and if the defensive mechanisms (e.g., Intrusion Detection System (IDS)) of the victim are not alerted by the attacker's actions. As an example, we consider a code injection attack where the adversary's objective is to stealthily achieve the three phases of the attack. Firstly, in the communication phase of the attack, the adversary can exploit vulnerabilities detected in the target system, or can make use of malware (virus, Trojan horses, etc.).

Both methods open the door to performing code injection stealthily if the attacker specifically designs the attack to avoid triggering the defense mechanisms of the victim system. Therefore the injection attack is considered stealthy at the communication stage if the vulnerability exploitation or the malware communication is stealthy. An example of this first phase is the exploitation of the industrial communication protocols used in the CIs, e.g., the Modbus TCP protocol, commonly used in SCADA and DCS (Distributed Control System) networks for process control, which do not provide authentication of the source of a request. This provides an adversary with a chance to attempt to gather information on the system being controlled and about the PLC [53].

In the second phase, the execution of the injected code (see Figure 1), the level of stealthiness achieved in this stage depends on the implementation of the attack and on the defense mechanisms available in the targeted system. If the attack is designed to perform its tasks in a way that avoids triggering any alarm, and the security mechanisms implemented are not finely tuned to detect this kind of attack, the injection can be considered stealthy in its execution stage. To illustrate this assessment in the context of critical infrastructure protection (CIP), we analyze the PLCs Modicon M340 from Schneider Electric, which has a disclosed vulnerability to CSRF attacks [54]. These devices incorporate a web server interface that processes requests from clients about the underlying infrastructure. However, the web server does not implement security mechanisms to verify their authenticity, thus an adversary could trick a client into sending an unintentional request to the web server, which would be considered authentic [55].

The injected commands could be sent to the PLC through a specially crafted HTTP request, for example, sending the victim a request embedded in an image `<img src="http://plc-web-server.com/?query_string"/>`, where the `query_string` would request the server to perform some malicious action that would be considered legitimate. The adversary could exploit this vulnerability to remotely reset or alter the PLC's configuration. Lastly, we can assess the stealthiness of the propagation stage of a code injection. Through the exploitation of vulnerabilities, the attack could in some cases be successfully disseminated. However, through the use of malware it is possible to stealthily communicate the injection attack to other victims, as we have seen in the Stuxnet worm [14], or its variation Duqu [17], that were specifically designed to attack a particular PLC manufactured by Siemens, and infect numerous network devices without leaving evidence of the attack.

Therefore, we conclude that cyber attacks with disruptive or tampering objectives can be stealthily carried out through the three phases illustrated in Figure 1. We also stress that these types of attacks should be classified as very dangerous to ICSs, since the adversary could launch a potentially harmful attack that executes malicious actions and propagates its effects without being noticed, threatening not only a CI, but spreading the threat to other dependent or interconnected targets.

Table 1: Cyber stealth attacks and their relation with AICAn

| Category | Stealth Attacks | Stealthiness | | | IA | RA | II | RI | UI | HUI | IC | RC | InfAn | CAn | IntrAn |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Disconnection and Goodput Reduction | Unreachability of the nodes | ○ | | | ✔ | ✔ | | | | | | | | ✔ | |
| | Removal of entries in routing tables | ○ | ❍ | | ✔ | ✔ | | ✔ | | | | | | | ✔ |
| | Goodput reduction | ○ | ❍ | | L | L | | ✔ | | | | | | ✔ | |
| Active Eavesdropping | Traffic hijacking | ○ | ❍ | | U | | | U | | | ✔ | U | | ✔ | |
| | Modification of the routing tables | ○ | ❍ | | U | | | U | | | ✔ | U | | ✔ | |
| Scanning and Probing | TCP connect() scanning | ○ | | | | | | | | | | ✔ | | | |
| | TCP SYN scanning | ○ | | | | | | | | | | ✔ | | | |
| | TCP FIN scanning | ○ | | | | | | | | | | ✔ | | | |
| | Christmas scan | ○ | | | | | | | | | | ✔ | | | |
| | Null scan | ○ | | | | | | | | | | ✔ | | | |
| Side-Channel Exploitation | Timing attack | ○ | | | | | | | | | | ✔ | | | |
| | Power analysis attack | ○ | | | | | | | | | | ✔ | | | |
| | Electromagnetic analysis attack | ○ | | | | | | | | | | ✔ | | | |
| | Fault induction attack | ○ | | | | | | | | | | ✔ | ✔ | L | ✔ |
| | Optical side channel attack | ○ | | | | | | | | | | ✔ | | | |
| | Traffic analysis attack | ○ | | | | | | | | | | ✔ | | | |
| Covert Channel Exploitation | Due to virus and malware | ○ | ❍ | ● | U | U | U | | L | L | ✔ | ✔ | | L | U |
| | Targeting important resources | ○ | ❍ | | L | U | L | U | | | ✔ | ✔ | | L | L |
| | Targeting sensitive data | ○ | ❍ | | L | | L | | | | ✔ | ✔ | | U | |
| | Using vulnerable protocols | ○ | ❍ | ● | | U | L | | ✔ | ✔ | ✔ | ✔ | | ✔ | |
| | Using design flaws | ○ | ❍ | ● | U | U | ✔ | ✔ | L | L | ✔ | ✔ | | ✔ | |
| | Using packet headers | ○ | ❍ | | | | | | | | ✔ | ✔ | | | |
| | Using super user permissions | ○ | ❍ | ● | L | U | U | ✔ | | | ✔ | ✔ | | ✔ | |
| | Using handshake trials | ○ | ❍ | | | | | | | | ✔ | ✔ | | | |
| | Using public resources | ○ | ❍ | | ✔ | ✔ | ✔ | L | | | ✔ | ✔ | | ✔ | |
| | Using lack of authentication | ○ | ❍ | | U | U | L | L | ✔ | ✔ | ✔ | ✔ | | ✔ | L |
| Code Injection | Database injection | ○ | ❍ | | ✔ | ✔ | ✔ | ✔ | | | ✔ | ✔ | | ✔ | ✔ |
| | Command injection | ○ | ❍ | ● | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | | ✔ | ✔ |
| | Website injection | ○ | ❍ | ● | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | | ✔ | ✔ |
| | OS injection | ○ | ❍ | ● | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | | ✔ | ✔ |

○: stealthy communication of the attack. ❍ : stealthy execution of the attack. ● : stealthy propagation of the attack.
✔: the threat violates a security property of AICAn. L: the threat is likely to break a security property of AICAn. U: the threat is unlikely to break a sec. property of AICAn.

Table 1 summarizes the contents that have been reviewed in this section, providing a tentative analysis of the threats that stealth attacks pose to CIs in relation to the AICAn taxonomy. In this table, divided into targeted areas and threat categories, it is possible to observe that attacks are closely related to one another, since attackers, irrespective of their modus operandi, generally base their goals on the execution of a set of combined threats to the AIC of the system, as discussed previously. The AICAn analysis is based on the discussion, by a group of experts, of the impact on AICAn by different implementations of each stealthy attack listed. It is important to note that the assignment of likelihood in this table is determined by the different implementations of each of the selected stealth attacks, and may vary if other examples are taken into account. However, we believe this study shows an interesting overview on the impact of stealth attacks on CIs from the point of view of AICAn. From Table 1 we can conclude that most of the stealth cyber attacks focus on altering the integrity of the information of the system, possibly inducing threats to the availability of resources and information, and consequently causing control anomalies.

Additionally, some of the more sophisticated attacks expand their scope to also exploit the system's vulnerabilities in order to alter the integrity and confidentiality of the resources and information, and introduce the possibility of impersonation (UI and HUI compromising), producing CAn and IntrAn anomalies. From this table, we conclude that most of these attacks focus on the exploitation of the vulnerabilities associated with control and also those vulnerabilities intentionally produced by intruders. We also note that threats classified as covert channel exploitation and code injection can become potentially harmful threats to CIs, since they can compromise or degrade a wider range of security properties necessary for the good operation of critical systems, endangering the availability, integrity and confidentiality of these systems.

# 5 Countermeasures and Prevention Mechanisms Against Stealth Attacks

Given the restrictive nature of stealth attacks where the adversary wants to carry out his actions unnoticed, they must be very precise and tailored to the target system. Therefore, the defense mechanisms and the countermeasures applied must always take into account the environment of the system that is being protected. In this section we discuss measures that counteract stealth attacks equivalent to those discussed in Section 4 in general-purpose networks, which are applicable to critical settings with the adequate adaptations to fit the constrained environment of CIs, e.g., protocol reinforcements, introduction of additional equipment within the network, physical measures, etc. An extensive review of the literature provides two main lines of action for the protection of CIs: *avoidance mechanisms* (passive protection) and *detection and recovery mechanisms* (active protection). We devote this section to providing some ideas about how to protect the systems or minimize the effects of these stealthy attacks.

Avoidance mechanisms are put into place to prevent threats and reduce risks, while detection and recovery provide early detection and warning against attacks, and help restore the system to its original working state, palliating the effect of anomalies or attacks. These protection mechanisms are applied to counteract the weapons used to perform the attacks. The most threatening of the weapons under consideration, i.e., the one with the least visibility and cost, is the use of impersonation. The use of lies is a weapon with an inferior degree of stealthiness than impersonation, however it is also threatening if the attacker uses it to propagate incorrect information to corrupt the targeted system. Overloading has the lowest degree of stealthiness, nevertheless a skilled adversary could make use of it to collapse a subsystem of a CI without drawing the attention of the system administrators.

In general terms, it is possible to employ different methods to counteract these weapons; the main avoidance mechanisms that can be used are: *cryptography*, *standardization* and *reputation mechanisms*. Apart from these, when addressing each different attack, it is possible to apply specific countermeasures, either active or passive protection. The use of *cryptographic authentication methods* improves resistance against stealth attacks, since cryptographic authentication is harder to forge than IP addresses, etc. It is also important to note that in the field of CIP, the most-used protocols (e.g., Modbus [53]) still lack authentication mechanisms, something that is advantageous to the attacker [35]. Additionally, the naturally scarce resources such as bandwidth, storage, computation capabilities or power, provide the adversaries with targets to easily bring down the operation of the network.

Nevertheless, the implementation of cryptography in constrained systems is challenging, thus it is necessary to consider the use of lightweight cryptographic primitives for authentication, e.g., symmetric cryptography or elliptic curve cryptography [41]. However, to only rely on authentication is insufficient to thwart stealth attacks, since the corruption of legitimate nodes' behaviors perverts the correct authentication processes [35]. Thus it is necessary to strengthen the authentication process by *applying recommended and standard procedures.* The IEC-62351-8 standard [56], focuses on the security of remote control substations, and underlines the need to implement access control mechanisms using the technique of *Role-Based Access Control* (RBAC) together with the restrictive principle of the minimum privilege. This principle states that the sole entities able to gain access to logical devices and modify their objects will be those (virtual and physical) entities with the suitable permissions to operate in the field.

To address this, authentication must be based on the assignation of subjects-to-roles and roles-to-rights, restricting the accesses to particular objects developed in substations (e.g., IEC-61850 objects). This difficulty is increased due to the knowledge uncertainty about the honesty of the different hosts. However, several of the aforementioned problems can be palliated (even solved) when deploying *reputation mechanisms* to protect the networks, so that even if the nodes are compromised by adversaries, the reliability of the system can still be assured. The use of

Table 2: Cyber stealth attacks summary table

| Category | Stealth Attack | Stealthiness | | | Weapons | Countermeasures |
|---|---|---|---|---|---|---|
| Disconnection and Goodput Reduction | Unreachability of the nodes | ◦ | | | Impersonation<br>Lies<br>Overloading | Cryptography<br>Reputation mechanisms |
| | Removal of entries in routing tables | ◦ | ❍ | | | |
| | Goodput reduction | ◦ | ❍ | | | |
| Active Eavesdropping | Traffic hijacking | ◦ | ❍ | | | Cryptography<br>Reputation mechanisms |
| | Modification of the routing tables | ◦ | ❍ | | | |
| Scanning and Probing | TCP connect() scanning | ◦ | | | | Stealth probes<br>Honeypots |
| | TCP SYN scanning | ◦ | | | | |
| | TCP FIN scanning | ◦ | | | | |
| | Christmas scan | ◦ | | | | |
| | Null scan | ◦ | | | | |
| Side Channel Exploitation | Timing attack | ◦ | | | | Hiding timing variations<br>Blinding techniques<br>Masking techniques<br>Protective casing<br>IDS and validation of computations<br>Disabling and masking of light signals<br>Encryption and masking of the channel |
| | Power analysis attack | ◦ | | | | |
| | Electromagnetic analysis attack | ◦ | | | | |
| | Fault induction attack | ◦ | | | | |
| | Optical side channel attack | ◦ | | | | |
| | Traffic analysis attack | ◦ | | | | |
| Covert Channel Exploitation | Due to virus and malware | ◦ | ❍ | ● | | Anti-malware<br>Resource monitoring<br>Special security mechanisms applied to sensitive data<br>Secure protocols<br>Design assessment and correction<br>Use of IDS<br>Proper policies to assign permissions<br>Handshake restrictions<br>Restricted access to public resources<br>Authentication mechanisms |
| | Targeting important resources | ◦ | ❍ | | | |
| | Targeting sensitive data | ◦ | ❍ | | | |
| | Using vulnerable protocols | ◦ | ❍ | ● | | |
| | Using design flaws | ◦ | ❍ | ● | | |
| | Using packet headers | ◦ | ❍ | | | |
| | Using super user permissions | ◦ | ❍ | ● | | |
| | Using handshake trials | ◦ | ❍ | | | |
| | Using public resources | ◦ | ❍ | | | |
| | Using lack of authentication | ◦ | ❍ | | | |
| Code Injection | Based on design flaws | ◦ | ❍ | ● | | Monitoring tools<br>Prevention and validation mechanisms |
| | Based on malware propagation | ◦ | ❍ | ● | | |

◦: stealthy communication of the attack. ❍ : stealthy execution of the attack. ● : stealthy propagation of the attack.

reputation has various advantages, such as the use of collaborative methods, which provides robustness to the design of the network and eliminates the connectivity dependencies between nodes [35].

Cryptography and reputation measures are especially beneficial for *goodput reduction attacks*. Although these two main countermeasures try to minimize and palliate all kinds of stealth attacks against the networks, they are particularly useful in the case of the disconnection attacks or the *active eavesdropping*, where once detected, the traffic going through the corrupted nodes can be averted or reduced [20].

*Scanning and probing attacks* are one of the most critical types of stealth attacks, since they open the door to other more sophisticated and more informed attacks. Some countermeasures against these attacks are provided by V. Marinova-Boncheva in the paper [30]. The author proposes the use of stealth probes to detect any attacker that prolongs his procedures for a long period of time, for example, checking for system vulnerabilities and open ports for a period of two months. To this end, the stealth probes collect information from the system, checking for methodical attacks that last an extended period of time, they sample a wide area and discover correlating attacks. Basically this technique implies the use of mixed signature-based and anomaly-based IDSs.

Another way to confront stealth scanning and probing is proposed by C. Yin et al. in [37], where they suggest the use of honeypots to detect the attacks and alert the system's administrators. A honeypot is "*an information system resource whose value lies in unauthorized or illicit use of that resource*"; it reacts like a normal machine, based on the type of operating system it simulates, while it is recording and transferring packets to scan detection mechanisms to learn the tactics and tools used by the attackers and alert the administrators of illegal accesses to the network it is protecting.

The countermeasures for *side channel attacks* are highly tailored to the type of exploitation and the actual implementation of the attack. G. Joy Persial et al. provide certain guidelines to counteract side channel attacks in their work in [31]:

- *Timing attacks*: this kind of attack can be prevented by hiding time variations or using blinding techniques [57]. A simple form of hiding variations is to make the computations in constant time. Another possibility is to always add certain computations to the execution of the algorithms to mask the timings. Other variations include hiding the internal state of the systems, so that the attacker is no longer able to simulate internal computations.

- *Power analysis attack*: the power consumption is reduced using masking and elimination techniques. Masking "*randomizes the signal values at the internal circuit nodes while still producing the correct cipher text*" [31]. It can be done at software level, adding random masks to data subsequently encrypted, or at hardware level where the system adds random mask bits to balance the degree of randomness of the resulting message.

- *Electromagnetic analysis attack*: this kind of attack can be prevented by covering the system with a protective casing that hides or attenuates the electromagnetic radiations. This case also prevents the attacker from accessing the individual physical components of the system.

- *Fault induction attack*: can be prevented by checking the computations [57] or verifying the signature of the sent messages to identify the failures. There are IDSs specifically designed to identify core failures and hijacks and the correct operation of the systems [58] [59].

- *Optical side channel attack*: to prevent the adversary from retrieving information from display monitors and leds, once the device is ready to be deployed. These lighting signals used for debugging should be disabled, or masked.

- *Traffic analysis attack*: counteracting this type of attack is very difficult [31], since it is necessary to encrypt the messages transmitted and mask the channel, to prevent the adversary from analyzing the traffic. In their work in [60], J. Deng et al. provide different countermeasures to prevent this attack, based on modifications of the routing schemes used by the nodes of the network.

Existing countermeasures for *covert channels* are varied, and comprise the use of commercial solutions such as antivirus and anti-malware SW, and restricting and strengthening the implementation of the network's protocols and policies. Examples are [33]:

- *Anti-malware*: as we have previously seen, software such as viruses, worms and Trojan horses can be introduced inside the victim's system, to capture packets and inject scripts into the victim's programs. Updated anti-virus and anti-malware SW can generally detect these behaviors.

- *Resource monitoring*: resources such as system files, disks, RAM, sockets, etc. are valuable to attackers, and thus adversaries frequently target them. Monitoring these resources with HIDS can provide insight into the system's status and help detect the presence of covert channels.

- *Data sensitivity*: information can be classified according to its level of sensitivity, thus special security mechanisms can be put into place to differing degrees to protect the data according to its sensitivity.

- *Secure protocols*: to protect the systems against covert channels attacks, it is important to strengthen the security of the network, thus implementing secure protocols, e.g., HTTPS instead of HTTP, helps prevent such attacks and protects the transmission of sensitive information.

- *Design robustness*: covert channels take advantage of design oversight vulnerabilities,and weaknesses due to the system's design. In the first case, these unintentional failures can be corrected once discovered, removing the covert channel. In the second case, they cannot be removed until the system is re-designed to eliminate the vulnerabilities. However, the use of good practices, such as secure programming or process desegmentation, can make the system more resilient against covert channels.

- *Network Intrusion Detection Systems (NIDS)*: such as Snort [61], monitor packet header fields such as ACK, SYN, to detect patterns that can indicate (unmask) the presence of covert channels.

- *Super user permissions*: super user permissions may be needed to execute software, but it is necessary to carefully evaluate the processes granted with these permissions, to avoid harmful routines that are able to damage the system.

- *Handshake restrictions*: handshake trials between systems can be a way used by a malicious actor to fool traffic monitoring systems, thus a limitation on these trials should be put into place.

- *Public resources*: the access to public resources such as printers or shared disks should be restricted and limited to the known users of the network, and reinforced with authentication methods for preventing covert channels. For example, the use of RBAC, Attribute-Based Access Control (ABAC), Kerberos or simple Public Key Infrastructure (PKI) could help.

- *Authentication*: methods like passwords, captchas [62] or biometric mechanisms can help protect the system against covert channels, as well as RBAC/ABAC, Kerberos or PKI. Additionally, the IEC/TS 62351-8 [56] standard for security in substations recommends the use of authentication mechanisms, and more particularly RBAC to reduce complexities in the entire SCADA network.

Prevention methods for covert channels are not restricted to just these points. Since the covert channels implemented for a system are highly tailored to its individual characteristics, each of the targeted environments will provide new challenges to the adversary. Thus, new behaviors will appear, and consequently, the targeted systems can be protected in different ways according to each specific situation.

Regarding the countermeasures that can be put into place to prevent and fight *code injection attacks*, in addition to the general measures that can be used (i.e., cryptography, standardization and reputation), it is possible to take two different approaches: *prevention and validation mechanisms* and *monitoring tools* (e.g., IDSs, antivirus, anti-malware SW).

To prevent code injection, it is important to secure the input and output handling, by introducing validation mechanisms, selective inclusion and exclusion procedures, standardized input and text formatting and encoding, parametric variables, dissociation and modularization of the procedures from the kernel of the system, good handling of super user credentials, isolation of some critical procedures, hash validation of executable images, and similar mechanisms [44, 63].

In order to detect the most sophisticated and stealthy injection attacks, it is important to deploy intelligent and finely tuned IDSs, capable of adapting to new dynamics and learning new attacks [64], beyond just relying on attack signatures and known events. These automatic and adaptive capabilities provide the detection systems with tools to detect and prevent highly targeted and complex stealth attacks [65, 66, 67].

Most of the countermeasures and preventive mechanisms discussed in this section can be categorized as avoidance mechanisms (passive protection), however, as cyber attacks against control systems are becoming increasingly aggressive and sophisticated, it is necessary to put into place active protection mechanisms, to address the continuous threats to the CIs [8, 68]. Thus, as discussed and as a complementary measure to avoidance mechanisms, detection and recovery mechanisms are the techniques put in place for early detection, prevention of and counteraction to risks in order to restore the system to its original working state, and palliate the effect of the attacks or anomalies happening within the system.

Given this definition, we classify the active protection mechanisms into two main categories: the methods that require the intervention of an operator, and the automatic methods. Within the first class, we find the early warning systems, the IDS, and all the situational awareness [21] mechanisms deployed to detect and alert the human operators of any attack or anomaly happening within the system under surveillance. To the contrary, the automatic methods are those tools deployed to provide an automatic response to the problems that arise, with little to no supervision from the human operators.

Currently there is little literature on the automatic or semi automatic response mechanisms, since their application to CIs is complex and potentially dangerous, due to the criticality of the environment. However, it is absolutely essential to start to deploy such techniques within CIs, since faster counteractions would help prevent the effect of attacks or anomalies from cascading to other interconnected and interdependent CIs [68]. Solutions that can provide these automatic functionalities are the *Intrusion Prevention Systems* (IPSs), SW that "*has all the capabilities of an intrusion detection system and can also attempt to stop possible incidents*" [69].

The IPS is often integrated as an extension of the IDS, but it usually receives less attention than IDS research due to the intrinsic complexity of developing the mechanisms that offer an automated and correct response against certain events. However, the increased complexity and speed of cyber-attacks in recent years shows the acute necessity for complex intelligent dynamic response mechanisms [64]. These systems can perform a wide variety of actions, from operations on files and re-routing, to automatic revocation of privileges for certain profiles of the infrastructure. Thus, using this module, it is not necessary to alert the system's human operator/administrator to launch countermeasure actions, the system itself could select and execute them in a semi-supervised or unsupervised way.

In Table 2, we summarize the analysis of the stealth attacks from the point of view of countermeasures and protection, also reviewing the level of stealthiness of the attacks corresponding to Figure 1. This evaluation takes into account their associated AICAn risks (see Table 1), always considering the worst scenario possible; i.e., the maximum level of stealthiness that an adversary can achieve using these techniques and approaches. Moreover, we provide an overview of the most suitable countermeasures applicable to prevent or react against the stealth attacks, outlined in the last column of this table. This set of tentative measures is a selection of procedures that come from the context of general-purpose networks (trying to palliate or avoid stealth attacks in these non-critical settings) and which can be applied to CIs with a few adaptations to fit the specific needs of critical environments (industrial protocols, additional equipment, etc.).

## 6 Discussion on Cyber Stealth Attacks

According to M. Jakobsson et al. [20], stealth attacks are better (i.e., more profitable) than regular attacks, which require a higher amount of energy and leave the attacker more exposed to detection. In the previous sections, we have identified five different types (main categories) of stealth attacks, namely: (i) disconnection and goodput reduction, (ii) active eavesdropping, (iii) scanning and probing, (iv) covert and side-channel exploitation, and (v) code injection attacks. We have described their objectives and scope and using the AICAn taxonomy, we have determined their potential threats to CIs.

This study therefore shows the danger inherent in attacks where the adversary tries to go unnoticed, since the system can be threatened for long periods of time without being protected, the actions against the infrastructure are varied and range from simple probing of the system to extraction of sensitive information, or disruption to the correct operation of the CIs affected. Additionally, the adversaries are able to propagate their threats to other nodes or interdependent CIs, thus creating cascading effects through the interconnected infrastructures.

Besides the vulnerabilities introduced in the scenario associated to the interest of the infrastructure to adversaries (sensitive data, potential of social disruption, etc.), the high complexity of the environment and their interconnected nature increase exposure to potential attackers and unintentional errors. According to NIST [70], a high number of interconnections present increased opportunities for DoS attacks, introduction of malicious code or compromised HW. Moreover, when dealing with a vast amount of nodes in the network, as happens in CIs, the number of entry points and paths exploitable by and adversary increases.

Nevertheless, there are several methods that help prevent and counteract the attacks studied. The main actions we find that currently are indicated to help in the case of stealth attacks are the preventive mechanisms, such as reputation or cryptography. We find therefore that it is essential to incorporate protection tools for control elements, governance, validation and testing of SW and HW components, to prevent any perturbation to the system's security properties. Moreover, protection of communication channels (using for example cryptography, virtual private networks, bump-in-the-wire, etc.) is also needed, since most of the cyber threats rely on attacks against the confidentiality (information or configurations of resources), in order to learn about the environment, conditions and elements of the victim system.

However, in the event of truly sophisticated stealth attacks, it is necessary to include a layer of protection that provides reactive recovery mechanisms capable of launching automatic reactions against an attack that is underway, to restore the normal operation of the system under attack, as soon as possible. Within this category we find the IDS and IPS modules, capable of advanced detection mechanisms, and in some cases, of launching some prevention actions and alerts to the security profiles responsible for the nodes under attack. Currently, there is little research on automatic and semi-automatic reaction systems, due to the inherent complexity of the modules, which is vastly increased in the case of CIs, where any disturbance of their operation is of critical relevance.

## 7 Conclusions

In this paper we have provided an overview of the different types of stealth attacks that can potentially target the CIs. We have discussed these attacks in their different stages through the AICAn taxonomy, and evaluated the potential risks these attacks can pose to the critical environments in terms of availability, integrity, confidentiality and the anomalies that can occur in the infrastructure. We conclude that stealth attacks are potentially very dangerous to CIs, and it is extremely difficult to fully secure networks against them, nevertheless we have reviewed several methods that help to prevent and to counteract some of these attacks, focusing on the conjunction of active and preventive security mechanisms. The establishment of the AICAn taxonomy and the study of criticality at each stage of the stealth attacks presented in this paper summarize the main risks deriving from stealthy attacks that can target the CIs in the world today. An extended analysis of this work could help determine and boost the capabilities of the security measures currently in place to detect stealth attacks, and it could help ascertain and identify the best countermeasures to prevent the damages derived from these attacks. The use of simulations would be very valuable to assess the risks and consequences of stealthy attacks in highly complex interdependent scenarios, thus we intend to develop a prototype of such a system, providing an AICAn-based model of the infrastructure where different kinds of stealth attacks can be launched in different areas of the system. Simulations in this area would help us understand the cascading effects across CIs and integrate machine learning algorithms to help predict the complex dynamics found in these types of scenarios.

# References

[1] C. Alcaraz, G. Fernandez, and F. Carvajal, “Security Aspects of SCADA and DCS Environments,” *Critical Infrastructure Protection*, vol. 7130, pp. 120–149, 2012.

[2] C. Directive, “114/EC of 08 December 2008 on the Identification and Designation of European Critical Infrastructures and the Assessment of the Need to Improve their Protection,” *Official Journal of the European Union*, vol. 345, 2008.

[3] Congress of the United States of America, “Public Law 107 - 56 - Uniting and Strengthening America by Providing Appropriate Tools Required to Intercept and Obstruct Terrorism (USA Patriot Act) Act of 2001,” USA PATRIOT ACT, October 2001, washington D.C. [Online]. Available: http://www.gpo.gov/fdsys/pkg/PLAW-107publ56/content-detail.html

[4] Homeland Security News Wire, “Black Hat Event Highlights Vulnerability of U.S. Critical Infrastructure,” Online News, July 2013, last Accessed May 2014.

[5] Computer News, “Chinese Hacking Team Caught Taking Over Decoy Water Plant,” Online News, August 2013, last Accessed May 2014. [Online]. Available: http://www.technologyreview.com/news/517786/chinese-hacking-team-caught-taking-over-decoy-water-plant/

[6] J. Lopez, R. Setola, and S. Wolthusen, *Critical Infrastructure Protection: Advances in Critical Infrastructure Protection: Information Infrastructure Models, Analysis, and Defenses.* Springer, 2012, vol. 7130.

[7] ENISA, “Analysis of Annual Incident Reports 2012,” *Annual Incident Reports*, vol. 13, pp. 1–30, 2012.

[8] US DHS ICS-CERT, “Incident Response Summary Report,” September 2011, last access July 2013. [Online]. Available: http://www.uscert.gov

[9] ——, “ICS-Monitor Malware Infections in the Control Environment,” US CERT, December 2012, last accessed, April 2014. [Online]. Available: http://www.uscert.gov

[10] ——, “ICS-Monitor Brute Force Attacks on Internet-Facing Control Systems,” June 2013, last accessed, April 2014. [Online]. Available: http://www.uscert.gov

[11] European Comission, “Directive 2009/140/EC of the European Parliament and of the Council,” L337/37, November 2009, last Accessed May 2014. [Online]. Available: https://resilience.enisa.europa.eu/article-13

[12] B. Genge, I. Kiss, and P. Haller, “A System Dynamics Approach for Assessing the Impact of Cyber Attacks on Critical Infrastructures,” *International Journal of Critical Infrastructure Protection*, 2015.

[13] M. Thompson, "Mariposa Botnet Analysis," Technical report, Defence Intelligence, Tech. Rep., 2009.

[14] A. Matrosov, E. Rodionov, D. Harley, and J. Malcho, "Stuxnet Under the Microscope," *ESET LLC (September 2010)*, 2010.

[15] McAfee, "Global Energy Cyberattacks: Night Dragon," Version 1.4, McAfee Foundstone Professional Services and McAfee Labs, Tech. Rep., February 2011.

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

[17] Kaspersky Lab Expert, "Duqu: Steal Everything," 2011, last accessed, April 2014. [Online]. Available: http://www.kaspersky.com/about/press/major_malware_outbreaks/duqu

[18] K. Munro, "Deconstructing Flame: the Limitations of Traditional Defences," *Computer Fraud & Security*, vol. 2012, no. 10, pp. 8–11, 2012.

[19] US DHS ICS-CERT, "ICS-Monitor Incident Response Activity," National Cybersecurity and Communications Integration Center, April 2014, last accessed, May 2014. [Online]. Available: https://ics-cert.us-cert.gov

[20] M. Jakobsson, S. Wetzel, and B. Yener, "Stealth Attacks on Ad-hoc Wireless Networks," in *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, vol. 3. IEEE, 2003, pp. 2103–2111.

[21] C. Alcaraz and J. Lopez, "Wide-Area Situational Awareness for Critical Infrastructure Protection," *IEEE Computer*, vol. 46, no. 4, pp. 30–37, 2013.

[22] B. Miller and D. Rowe, "A Survey SCADA of and Critical Infrastructure Incidents," in *Proceedings of the 1st Annual conference on Research in information technology.* ACM, 2012, pp. 51–56.

[23] B. Zhu, A. Joseph, and S. Sastry, "A Taxonomy of Cyber Attacks on SCADA Systems," in *Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing.* IEEE, 2011, pp. 380–388.

[24] C. Myers, S. Powers, and D. Faissol, "Taxonomies of Cyber Adversaries and Attacks: a Survey of Incidents and Approaches," *Lawrence Livermore National Laboratory (April 2009)*, vol. 7, pp. 1–22, 2009.

[25] H. F. Lipson, "Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues," DTIC Document, Tech. Rep., 2002.

[26] ENISA, "Existing Taxonomies," 2005-2013, last Access on August 2013. [Online]. Available: http://www.enisa.europa.eu/activities/cert/support/incident-management/browsable/incident-handling-process/incident-taxonomy/existing-taxonomies

[27] F. Skopik and Z. Ma, "Attack Vectors to Metering Data in Smart Grids under Security Constraints," in *Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual.* IEEE, 2012, pp. 134–139.

[28] T. Tsao, R. Alexander, M. Dohler, V. Daza, and A. Lozano, "Routing Over Low Power and Lossy Networks," pp. 1–50, January 2012, last Access on August 2013. [Online]. Available: https://datatracker.ietf.org/doc/charter-ietf-roll/

[29] E. Rescorla and B. Korver, "Guidelines for Writing RFC Text on Security Considerations," *IETF, RFC-3552*, vol. 1, pp. 1–44, 2003. [Online]. Available: https://tools.ietf.org/html/rfc3552

[30] V. Marinova-Boncheva, "A Short Survey of Intrusion Detection Systems," *Problems of Engineering Cybernetics and Robotics*, vol. 58, pp. 23–30, 2007.

[31] G. Joy Persial, M. Prabhu, and R. Shanmugalakshmi, "Side channel Attack-Survey," *Int J Adva Sci Res Rev*, vol. 1, no. 4, pp. 54–57, 2011.

[32] J. Kong, O. Aciiçmez, J.-P. Seifert, and H. Zhou, "Hardware-software Integrated Approaches to Defend Against Software Cache-based Side Channel Attacks," in *15th International Symposium on High Performance Computer Architecture.* IEEE, 2009, pp. 393–404.

[33] N. Tomar and M. S. Gaur, "Information Theft Through Covert Channel by Exploiting HTTP Post Method," in *Tenth International Conference on Wireless and Optical Communications Networks (WOCN).* IEEE, 2013, pp. 1–5.

[34] A. Hintz, "Covert channels in TCP and IP Headers," Presentation at DEF CON Security Conference, Las Vegas, NV, USA, 2002.

[35] M. Jakobsson, X. Wang, and S. Wetzel, "Stealth Attacks in Vehicular Technologies," in *IEEE 60th Vehicular Technology Conference*, vol. 2. IEEE, 2004, pp. 1218–1222.

[36] G. F. Lyon, *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning.* Insecure, 2009.

[37] C. Yin, M. Li, J. Ma, and J. Sun, "Honeypot and Scan Detection in Intrusion Detection System," in *Canadian Conference on Electrical and Computer Engineering*, vol. 2. IEEE, 2004, pp. 1107–1110.

[38] IBM. (2013, November) IBM X-Force Trend and Risk Report. IBM. [Online]. Available: http://xforce.iss.net/xforce/xfdb/405

[39] U. Maimon, A. Kantor, and O. Dov, "Scan Detection," Jan. 3 2005, uS Patent App. 11/025,983.

[40] A. Dainotti, A. King, K. Claffy, O. Papale, and A. Pescapé, "Analysis of a/0 Stealth Scan from a Botnet," in *Proceedings of the 2012 ACM Conference on Internet Measurement.* ACM, 2012, pp. 1–14.

[41] J. Fan, X. Guo, E. DeMulder, P. Schaumont, B. Preneel, and I. Verbauwhede, "State-of-the-Art of Secure ECC Implementations: A Survey on Known Side-Channel Attacks and Countermeasures," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2010, pp. 76–87.

[42] M. M. Islam, R. Pose, and C. Kopp, "Suburban Ad-hoc Networks in Information Warfare," in *Proc. 6th Australian InfoWar Conference, Geelong, Australia*, 2005.

[43] Modbus-IDA, "Modbus Application Protocol Specification," 2006. [Online]. Available: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf

[44] J. A. Ambrose, R. G. Ragel, and S. Parameswaran, "RIJID: Random Code Injection to Mask Power Analysis Based Side Channel Attacks," in *Proceedings of the 44th annual Design Automation Conference.* ACM, 2007, pp. 489–492.

[45] D. Gollmann, "Securing Web Applications," *Information Security Technical Report*, vol. 13, no. 1, pp. 1–9, 2008.

[46] A. Grasso and P. H. Cole, "Definition of Terms Used by the Auto-ID Labs in the Anti-Counterfeiting White Paper Series," *Auto-ID Labs University of Adelaide, White Paper*, 2006.

[47] Z. Su and G. Wassermann, "The Essence of Command Injection Attacks in Web Applications," in *ACM SIGPLAN Notices*, vol. 41, no. 1. ACM, 2006, pp. 372–382.

[48] L. K. Shar and H. K. Tan, "Defending Against Cross-Site Scripting Attacks," *Computer*, vol. 45, no. 3, pp. 55–62, 2012.

[49] M. Van Gundy and H. Chen, "Noncespaces: Using Randomization to Enforce Information Flow Tracking and Thwart Cross-Site Scripting Attacks," in *NDSS*, 2009.

[50] A. Barth, C. Jackson, and J. Mitchell, "Robust Defenses for Cross-Site Request Forgery," in *Proceedings of the 15th ACM conference on Computer and communications security*. ACM, 2008, pp. 75–88.

[51] T. Jim, N. Swamy, and M. Hicks, "Defeating Script Injection Attacks with Browser-enforced Embedded Policies," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 601–610.

[52] OWASP, "The Ten Most Critical Web Application Security Risks," October 2010.

[53] Symantec, "TCP MODBUS - Unauthorized Read Request," last accessed, April 2014. [Online]. Available: http://www.symantec.com/security_response/attacksignatures/detail.jsp?asid=20674

[54] National Vulnerability Database, "Vulnerability Summary for CVE-2013-0663," NIST, April 2013. [Online]. Available: http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-0663

[55] US DHS ICS-CERT, "ICSA-13-077-01A Schneider Electric PLCs Vulnerabilities," June 2013, last accessed, April 2014. [Online]. Available: http://ics-cert.us-cert.gov/node/642

[56] *IEC/TS 62351-8 Power Systems Management and Associated Information Exchange - Data and Communications Security - Part 8: Role-Based Access Control*, IEC/TS Std., September 2011.

[57] J.-J. Quisquater and F. Koene, "Side Channel Attacks: State of the Art," *project CRYPTREC*, 2002.

[58] J. Reeves, A. Ramaswamy, M. Locasto, S. Bratus, and S. Smith, "Intrusion Detection for Resource-constrained Embedded Control Systems in the Power Grid," *International Journal of Critical Infrastructure Protection*, vol. 5, no. 2, pp. 74–83, 2012.

[59] R. Berthier and W. H. Sanders, "Specification-based Intrusion Detection for Advanced Metering Infrastructures," in *2011 IEEE 17th Pacific Rim International Symposium on Dependable Computing (PRDC)*. IEEE, 2011, pp. 184–193.

[60] J. Deng, R. Han, and S. Mishra, "Countermeasures Against Traffic Analysis Attacks in Wireless Sensor Networks," in *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*. IEEE, 2005, pp. 113–126.

[61] M. Roesch, "Snort-lightweight Intrusion Detection for Networks," in *Proceedings of the 13th USENIX conference on System administration*. Seattle, Washington, 1999, pp. 229–238.

[62] M. Blum, L. Von Ahn, J. Langford, and N. Hopper, "The CAPTCHA Project (Completely Automatic Public Turing Test to tell Computers and Humans Apart)," *School of Computer Science, Carnegie-Mellon University*, 2000. [Online]. Available: http://www.captcha.net

[63] P. Ratanaworabhan, V. Livshits, and B. Zorn, "NOZZLE: A Defense Against Heap-spraying Code Injection Attacks," in *USENIX Security Symposium*, 2009, pp. 169–186.

[64] S. Bologna and R. Setola, "The Need to Improve Local Self-Awareness in CIP/CIIP," in *First IEEE International Workshop on Critical Infrastructure Protection*. IEEE Computer Society, 2005, pp. 84–89.

[65] S. Avallone, C. Mazzariello, F. Oliviero, and S. P. Romano, "Protecting Critical Infrastructures from Stealth Attacks: A Closed-Loop Approach Involving Detection and Remediation," in *Critical Information Infrastructure Security*. Springer, 2013, pp. 209–212.

[66] S. D'Antonio, F. Oliviero, and R. Setola, "High-Speed Intrusion Detection in Support of Critical Infrastructure Protection," *Critical Information Infrastructures Security*, pp. 222–234, 2006.

[67] F. Pasqualetti, F. Dorfler, and F. Bullo, “Attack Detection and Identification in Cyber-Physical Systems,” *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[68] European Commission, *COM(2011) 163 - Achievements and Next Steps: Towards Global Cyber-Security*, ser. COM(2011) 163. Publications Office, 3 2011.

[69] K. Scarfone and P. Mell, “Guide to Intrusion Detection and Prevention Systems (IDPS),” *NIST special publication*, vol. 800, p. 94, 2007.

[70] Smart Grid Interoperability Panel Cyber Security Working Group and others, “NISTIR 7628-Guidelines for Smart Grid Cyber Security vol. 1-3,” 2010.