Network resilience

Luigi Coppolino, Salvatore D’Antonio & Luigi Romano
University of Naples Parthenope, Naples, Italy

Abstract

Generally speaking, resilience is the ability of a system to provide an acceptable level of service even in the face of faults and challenges to normal operation. In this chapter, focus is on two specific categories of challenges, namely, (i) cyber attacks and (ii) technology changes. We present a number of resilience-enhancing techniques, which are suitable for a wide variety of current and future Critical Infrastructures (CIs). We do so in a practice-oriented view and with respect to some of the most widely used and/or emerging network technologies for building CIs, specifically satellite networks, IP-based systems, and Wireless Sensor Networks. The techniques have been implemented in a distributed architecture, which integrates four main functions: monitoring, detection, diagnosis, and remediation. The ideas and results presented in this chapter are part of the lessons we have learned in INTERSECTION and INSPIRE, two FP7 projects we have been involved in. Additional information is available on the websites of the projects.

Keywords: Critical Infrastructure Protection, SCADA, Monitoring, Detection, Diagnosis, Remediation

1 Introduction

Resilience is an (relatively recently) abused word, in a number of fields. A sensible definition of resilience in computer networking is the one given by the ResiliNets research initiative [1]: “Resilience is the ability to provide and maintain an acceptable level of service in the face of faults and challenges to normal operation.” In this chapter, we will refer to the above definition of resilience, in the context of Critical Infrastructures (CIs) and with emphasis on two specific categories of challenges, namely, (i) cyber attacks and (ii) technology changes.

The reason why we decided to focus on cyber attacks is that we are witnessing a dramatic increase of external-borne security incidents, while internal incidents are basically stable and accidental incidents have increased only slightly (most probably, such a slight increase is mainly due to the increased complexity of the equipment, which results in more operator mistakes and interactions.
faults in general) [2]. In particular, the shared communication infrastructure has become an obvious target for disrupting a Supervisory Control and Data Acquisition (SCADA) network. For example, an attacker may exploit a vulnerability of the wireless trunk of a SCADA communication infrastructure to prevent real-time delivery of SCADA messages, which would result in the loss of monitoring information or even of the ability to control entire portions of the SCADA system. Evidence is showing that current CIs are exposed to major security risks. As an example, in [3] it is reported that Cyberspies have penetrated the U.S. electrical grid and installed malicious software programs that could be used to disrupt the system. The recent Internet worm Stuxnet specifically targeting SIEMENS WinCC SCADA systems [4] is another remarkable example of the vulnerability of current CIs. The worm exploits systems running WinCC SCADA systems configured with the default hardcoded password. Changing the password is not a viable option, since it could interrupt communications between the WinCC software and the database and interfere with the operations. Thus, SIEMENS recommended not to change the password for guaranteeing business continuity. Even worse, when SIEMENS released a new tool for finding and removing the malicious software, along with a full-fledged security update for its SCADA management products, it was found that removing the worm might harm industrial systems. More precisely the company warned its users to contact the customer center with the following note: “As each plant is individually configured, we cannot rule out the possibility that removing the virus may affect your plant in some way” [5]. Finally, if one gets an opportunity to talk privately to the personnel in charge of information technology (IT) security at electric utility companies or at the Department of Homeland Security, they say that they are extremely worried about security exposure of their SCADA systems [6].

The reason why we decided to focus on technology changes is that new technologies are being increasingly used for building CIs, with major impacts on the security level, and it will be even more so in the future. Traditional CIs were intrinsically secure systems, due to a combination of factors, some of which are briefly described in the following:

- They consisted (almost exclusively) of special purpose devices, which were based on proprietary technologies.
- Individual sub-systems operated almost in isolation; that is, they did not interact with the external world, with the exception of the system being controlled.
- They were largely based on dedicated (as opposed to shared) communication links.
- They massively relied on proprietary (as opposed to open) communication protocols.

These trends have been largely subverted, and it will be even more so in the future. First, to achieve interoperability, open communication protocols are being increasingly used, thus exposing SCADA systems to the same vulnerabilities that threaten general purpose IT systems.

www.witpress.com, ISSN 1755-8336 (on-line)
Networks (WSNs) have become an integral part of a wide variety of CIs, for a number of reasons [7], both technical (WSN technology has the potential of significantly improving the sensing capabilities of SCADA sub-systems [8,9] as well as the resilience of the overall SCADA architecture [10,11]) and political (governments around the world have recognized the importance of WSNs as a key technology for the protection of CIs and have issued formal directives – as well as funded specific programs – for favoring the development of WSN technology in the context of CI protection [12,13]). Third, Commercial Off-the-Shelf (COTS) components are being massively used for implementing SCADA systems [14]. Fourth, sub-systems are being connected using the infrastructure of the corporate Local Area Network (LAN), or even Wide Area Network (WAN) links, possibly including the public Internet as well as wireless/satellite trunks.

The typical architecture of current CIs has a hierarchical structure, which integrates heterogeneous devices and network trunks, also via shared network connections.

As it is apparent from the figure, there are multiple paths for accessing a CI. The US-CERT review [6] identifies several attack patterns to break into a CI, in particular (i) evading the firewall protection typically in place to isolate the business LAN from the Internet and the control LAN from the business LAN;

![Diagram of alternative communication paths into and out of a typical architecture of a next generation CI.](image.png)

Figure 1: Alternative communication paths into and out of a typical architecture of a next generation CI.
(ii) exploiting the dial-up access typically available for Remote Terminal Unit (RTU) control and management, which typically requires no authentication or a password for authentication; (iii) exploiting the connection reserved to vendor support; (iv) exploiting database links typically used for logging actions of the production control system; (v) gaining tight control over neighboring utilities by a peer-to-peer connection to the control system; (vi) flooding the control system LAN; (vii) sending commands directly to the sensors/actuators, which generally accept any properly formatted command; (viii) exporting the human–machine interface (HMI) screen, as an example with a reverse virtual network computing (VNC) connection, hence obtaining the complete control guaranteed by the operator control panel; (ix) manipulating the data in the database; and (x) conducting a man-in-the-middle attack, which allows the attacker to manipulate both the stream of commands toward the sensors/actuators and the data gathered from the periphery.

In the INTERSECTION [15] and INSPIRE [16] projects, we have investigated the vulnerabilities stemming from the aforementioned attack patterns and developed effective techniques for detecting, diagnosing, and countering the effects of attacks to a CI.

In the remainder of this chapter, we illustrate a variety of resilience-enhancing techniques that are suitable for current and future CIs. We do so in a practice-oriented view and with respect to specific network technologies. More precisely, the rest of the chapter is organized as follows. In Section 2 we present a distributed system comprising a number of components, whose operation is orchestrated in order to actually improve the resilience of the communication network supporting a typical CI. Section 3 illustrates how intrusion detection techniques can be used to protect satellite networks against cyber attacks that exploit vulnerabilities affecting satellite networks. Section 4 illustrates how intrusion detection and remediation techniques can be used to protect an IP-based network segment against Distributed Denial of Service (DDoS) attacks. Section 5 illustrates how tight integration of detection, diagnosis, and reconfiguration techniques can be used to effectively protect a CI consisting of a WSN zone and an IP-based network segment. Finally Section 6 closes the chapter with our final remarks.

We emphasize that most of the ideas and results presented in this chapter are part of the lessons we have learned from two FP7 projects, namely, INTERSECTION and INSPIRE. Additional information is available on the websites of the projects.

2 A component-based framework for improving network resilience in CIs

In this section we present the conceptual architecture of a distributed system comprising a number of components, whose operation is orchestrated in order to actually improve the resilience of the communication network supporting a
typical CI. The functional modules of the resilience-enhancing framework are depicted in Figure 2.

The choice of a distributed system was motivated by the possibility of separating concerns among a well-defined set of entities, each conceived to deal with a particular aspect of the problem. This has two important advantages: First, it simplifies the task of each involved entity; second, it allows a deeper specialization of each module (which can thus be modified without necessarily affecting the performance of the overall system).

The framework encompasses four main functions: monitoring, detection, diagnosis, and remediation. Monitoring aims at gathering and aggregating status data from diverse parts of a communication: both wired and wireless sensors, RTUs, historians, and supervisory stations. In order to cope with the heterogeneity of the formats of such data, grammar-based adaptable parsers [17] are employed to translate raw events to an intermediate format, so that they can be merged in a single data stream for further analysis and processing. Therefore, data provided by monitoring probes disseminated throughout the network are parsed and then used to feed the detection components, each adopting a different approach for detecting intrusions/faults. The output of each detection module is processed by a diagnoser, which is in charge of (i) clearly identifying the causes of the attacks/faults, (ii) accurately estimating their consequences on individual system components, and (iii) selecting the most suitable technique to treat the detected intrusion/fault among the implemented remediation mechanisms. The basic idea is to use an evidence-accruing fault/intrusion tolerance system to choose and carry out one out of multiple recovery/reconfiguration strategies, depending upon the perceived severity and/or on the adjudged nature of the fault/attack.
The framework [18] naturally lends itself to a context-aware deployment of the available resources, thus allowing a more effective placement of the different system modules. Dynamic deployment of the distributed system components is actually needed in order to ensure both flexibility of the architecture and resilience in the face of changes in network technologies, application services, and traffic conditions.

In the following sections, we show how the proposed framework can be used to detect, diagnose, and remediate cyber attacks against a CI that uses diverse network infrastructure technologies.

3 Intrusion detection and reaction in satellite networks

In this section we illustrate how intrusion detection techniques can be used to protect satellite networks against cyber attacks that exploit vulnerabilities affecting the so-called Performance-Enhancing Proxies (PEPs) [19]. PEPs are key elements of the transmission control protocol (TCP) accelerators, which are located at the edges a typical geostationary satellite link [20], for improving the performance of the TCP connections.

The major drawback when running PEP is the violation of the TCP end-to-end paradigm. In fact, PEP receives TCP segments, forwards them to the TCP receiver, and sends local acknowledgments (ACKs) to the TCP sender. The rationale of the use of PEP is to make the growth of the TCP congestion window faster and then to increase the actual transmission rate, which is affected by the ACK reception timing.

In the target scenario, the use of PEP allows to achieve a relevant improvement of the network performance [21]: TCP sender experiences a Round-Trip Time (RTT) lower than 1 ms or at most few ms instead of more than 500ms experienced in case of an end-to-end TCP connection.

In this context, a PEP intentional or unintentional failure occurring after having sent an ACK and before checking the proper delivery of the corresponding packet to the actual receiver leads to an irreversible violation of the end-to-end reliability (Figure 3).

As an example of attack exploiting PEP vulnerabilities, let us assume that an end-system uses a satellite link to download a file from an FTP (file transfer protocol) server. In order to perform the file transfer, the user terminal creates a TCP connection with the FTP server in order to set up an end-to-end reliable byte stream. However, PEP agents installed at the edges of the satellite link split the end-to-end TCP connection into three “sub-connections” in order to improve TCP performance:

1. A sub-connection from the FTP client to the satellite terminal running standard TCP,
2. A sub-connection from the satellite terminal to the satellite gateway running an optimized TCP version, and
3. A sub-connection from the satellite gateway to the FTP server running standard TCP.
TCP splitting is transparent to the TCP end-points: The TCP server believes that received ACKs come directly from the TCP client, which is receiving packets successfully. In the following, we describe what actually happens if PEPs are used while setting up a TCP connection:

- An end-to-end TCP three-way handshake is performed in order to set up a connection between the client and the server; the exchanged packets have the TCP SYN flag on.
- Three different sub-connections are created and managed by the PEP agents.
- Each PEP uses a local cache to store the sent packets, which have not been acknowledged yet; the motivation is to allow possible retransmissions in case some packets get lost.

In this network scenario, a malicious user accesses PEP running in proximity of the satellite gateway and installs a malware application, which drops all the TCP packets from the PEP to the client network. Although there is nothing wrong experienced by the FTP server, that is, the TCP sender, the TCP connection will be kept open in order to allow the reception of a packet notifying the successful execution of the file transfer. That packet, instead, will not be generated and, thus, the TCP sender will become aware of the failure of the file transfer after a certain time period whose value depends on both TCP and application timers. The effect of the described attack is that the FTP client is not allowed to download the file.

In this scenario, attack detection relies on the use of two detection tools disseminated throughout the network to be protected. The first tool, called SYN detector, runs on satellite gateway and keeps track of the active TCP connections. The second tool, referred to as TCP traffic analyzer, runs on the access routers of the client and the server network, and is in charge of monitoring all the TCP connections having nodes of the satellite network as source or destination. The TCP traffic analyzer measures to total amount of transferred bytes for each active TCP connection. Both SYN detector and TCP traffic analyzer

Figure 3: TCP spoofing application over satellite.
collect TCP/IP statistics and forward them to a detection module that implements the attack detection logic. Outputs of the detection module are forwarded to the diagnostic module, which evaluates whether the detected anomaly can be considered as an attack or not. If an attack is detected, an alarm is generated.

The remediation module sends a control message to the PEP of the satellite terminal under attack, which causes its shutdown. Therefore, the PEP-related attack is counter-measured by removing the PEP from the network scenario, which implies a degradation of the network performance and the perceived quality of service. An attack report is produced and sent to the satellite network administrator. The report describes the pattern of the attack and is aimed to support network administrator in order to properly configure the PEP.

4 Detection and remediation of a distributed attack over an IP-based network

In this section we illustrate how intrusion detection and remediation techniques can be used to protect an IP-based network segment against DDoS attacks.

Usually, a DDoS attack abuses network protocols in order to saturate the resources of a network server, thus preventing legitimate users from using the provided service. Attackers usually attempt to consume the limited resources of the victim without directly violating it.

A common way to perform a DDoS attack is to take control of other hosts that are then used by the attacker to perpetrate the distributed attack.

A typical DDoS attack scenario involves several components widely distributed throughout the network: a “master,” who initiates and orchestrates the distributed attack; several “agents,” or “zombies,” who receive commands from the master and launch the real attack; and a target node, which represents the victim of the attack.

An attacker, who intends to perpetrate a DDoS attack against a web-based component of a CI, first needs to “recruit” the necessary computing power. Indeed, the greater the available computing power, the greater the effects of the attack, since the capability of the attacker to saturate the victim’s resources increases. For this reason, the attacker needs to opportunely violate several hosts, in order to recruit them as “accomplices.” There exist several ways of compromising agent nodes: A trojan horse could be exploited, allowing the attacker to download an agent on the violated machine. Weaknesses in the software code that accepts remote connections can also be exploited for the recruitment process.

Once the victim has been identified, the attacker launches the attack by running the attack code on the master host. The master communicates with agents in order to instruct them about the attack to perform against the target by means of common packet flooding procedures. The distributed flooding is properly orchestrated by the master, in order to amplify the effects of the DDoS attack. Let us assume that each agent perpetrates a SYN FLOODING attack on the victim host. SYN FLOODING is a well-known type of attack, wherein the agent
sends a succession of TCP SYN requests to the victim without completing the
three-way handshake process with the expected ACK message. Since the victim
reserves memory for handling the connection associated with every TCP syn-
chronization request, such attack can rapidly saturate the victim host's resources
if the SYN request-sending rate exceeds the threshold defined by the time-out
mechanism for the synchronization process on the server site.

In the INTERSECTION project, we have developed techniques that can
effectively detect a DDoS attack to a CI and mitigate its effects. Such techniques
have been implemented in a prototype system, which has been validated using
the TFN2K tool for simulating a DDoS attack. TFN2K consists of two mod-
ules: a command-driven client to be installed on the master host and a daemon
program for the agent host. TFN2K exploits four different flooding mechanisms
for attacking the target: SYN FLOODING, UDP flooding, PING flooding, and
SMURF attack. The target just consists of a host running a web server applica-
tion, while the master and several agents running TFN2K modules represent the
distributed attack system. To make the deployment scenario more realistic, the
agents were placed in distinct network sites within the testbed infrastructure.
Target host and master will also be placed in different networks.

The detection process is performed by monitoring the system at two dif-
ferent levels: A detection module analyzes traffic metrics and compares their
values with specific patterns of activities and alerts the diagnostic module if
anomalies are detected; the host-level probe monitors the target machine by con-
trolling operating system parameters and identifies anomalous values of such
parameters, which are symptoms of the ongoing attack. The diagnostic module
correlates these two classes of symptoms and spots a SYN FLOODING attack.
Then, it sends an alarm to the remediation module. The level of confidence of
the decision-making process depends on the magnitude of the symptoms and on
the presence of both classes of symptoms.

Network monitoring probes are disseminated throughout the network in
order to effectively observe the evidences of the distributed attack.

The remediation process is initiated with the reception of an alert event from
the diagnoser. According to the event type, network policy, and available reme-
diation strategies, the remediation module selects the remediation strategy to be
implemented. In the case of a DDoS attack, typical policy will combine general
amelioration in the form of rate-limiting traffic filtering with a more focused
remedy based on creating a white list to clients that return a correctly formed
SYN-ACK. After pushing out the remedy, the remediation points are continu-
ally monitored for the amount of traffic they discard. If it falls below a policy-
determined threshold, it will withdraw the remedy.

5 Diagnosis-driven reconfiguration of WSNs

In this section, we illustrate how tight integration of detection, diagnosis, and
reconfiguration techniques can be used to effectively protect a CI consisting of a
WSN zone and an IP-based network segment. By diagnosis, we mean the capability of (i) clearly identifying the causes of the attacks and (ii) accurately estimating their consequences on individual system components.

The WSNs are often used in a physically semi-protected but nevertheless hostile environment, meaning one where people interested in altering the normal operation of the network can have physical access to the sensors, compromise them, or place equipment, such as intruder nodes, within the sensor network.

A well-known attack that can be launched by an intruder node against the WSN is commonly referred to as Sleep Deprivation Attack. This attack can be launched in a variety of ways depending on the particular routing protocol and its specific implementation. As an example, it is possible to send many broadcast routing packets. In this case, the attack is amplified if the fake routing packets force some nodes to change their parents, as in this case each fooled node will notify the change to all its neighbors, generating more traffic. Another way to conduct the attack is by sending unnecessary routing requests (RREQ) or by sending forged routing reply (RREP) packets that force the creation of loops in the WSN. In this case, due to the loops, packets are forwarded and stay alive for a longer time, hence resulting in unnecessary retransmissions and additional route messages. In the deployed scenario, we used a WSN composed of several CrossBow [22] IRIS motes and an MIB520 USB programming board. Sensors were equipped with TinyOS 2.x [23] and use Collection Tree Protocol (CTP) [24]. The attacker was exploiting a vulnerability of the CTP protocol [25] to conduct a Sleep Deprivation Attack by storing and forwarding multiple times a legitimate packet.

The attack has two negative effects on the WSN: (i) the discharge of batteries of all the nodes along the route (the path identified by triple arrows in Figure 4) from the malicious node to the base station and (ii) a denial of service for those nodes whose path toward the base station (identified with the symbol “X” in Figure 4) crosses the attacked overloaded path to the base station.

In a realistic scenario, all the packets reaching the base station are forwarded to a proxy, which in turn forwards them, on a TCP/IP channel, to the application server for the data delivery to the actual consumer (Figure 5). Every data packet generated by a node and reaching the base station is encapsulated into the payload of a TCP packet and sent to the application server via a virtual private network (VPN). In case of an attack, all the packets sent by the malicious node will reach the application server, which will recognize them as valid packets, as they are duplicates of valid packets, thus resulting in a manipulated view of the field. A smarter attack can use this behavior to attack the CI without generating extremely high duplicate packet rates, hence evading the intrusion detection system (IDS) for the WSN, but still compromising the application connected to the sensor network.

In order to detect this and other WSN-targeted attacks, it is necessary to correlate alerts generated by a WSN-aware IDS and probes deployed in the traditional IP network.

In [22], we presented a Hybrid IDS for WSN. The proposed solution relied on cooperation between IDS Local Agents and an IDS Central Agent. Each IDS
Local Agents run in sensor nodes and operate with an anomaly-based approach. The IDS Central Agent is deployed in the PC connected to the base station and correlates, with a signature-based approach, alerts thrown by the IDS Local Agents to other control information collected from the network.

It is possible to correlate (Figure 5) alarms given by the WSN-specific IDS and results given by an application-level probe for detecting the described attack. The WSN-specific IDS generates alarms based on the analysis of the network and periodically (e.g., every minute) calculates the packet generation rate at every node. Such data are sent in Internet Protocol Flow Information Export (IPFIX) [26] format to the correlation module, which analyzes them together with alarms triggered by the application-level probe for packet arrival.
rates at the application server higher than a threshold value defined in the training stage.

By correlating the data gathered by the two probes, the diagnostic module is able to detect the Sleep Deprivation Attack and even to identify the ID of the malicious node. The identity of the ID of the malicious node can be exploited while applying remediation and recovery actions. The reaction module is in charge of triggering reaction strategies. Strategies are selected from a list of possible options, reordered based on their effectiveness with respect to the diagnostic results (e.g., attacker ID, level of damage, and class of attack), and finally sequentially applied until the attack is stopped. Examples of possible reaction strategies for the described attack are (sorted by effectiveness) as follows:

- Over the air reprogramming of the malicious node
- Switching off (i.e., setting to sleep mode) the malicious node
- Isolation of the malicious node and if necessary manual shut down and reprogramming of the node

Based on the results of the applied strategies, the diagnostic report is iteratively updated.

6 Conclusions

Generally speaking, resilience is the ability of a system to provide an acceptable level of service even in the face of faults and challenges to normal operation. In this chapter, focus was on two specific categories of challenges, namely, (i) cyber attacks and (ii) technology changes.

With respect to the aforementioned challenges, the lesson we learned is that achieving network resilience in a sophisticated CI is only possible if four key functions – namely, monitoring, detection, diagnosis, and remediation – are effectively integrated in a high-performance distributed architecture. We presented a number of resilience-enhancing techniques, which are suitable for a wide variety of current and future CIs. We did so in a practice-oriented view and with respect to specific network technologies, namely, satellite networks, IP-based systems, and WSNs. We first presented the conceptual architecture of the resilience-enhancing distributed system. We then illustrated how intrusion detection techniques can be used to protect satellite networks against cyber attacks that exploit vulnerabilities affecting satellite networks. We also discussed how intrusion detection and remediation techniques can be used to protect an IP-based network segment against DDoS attacks. Finally, we illustrated how tight integration of detection, diagnosis, and reconfiguration techniques can be used to effectively protect a CI consisting of a WSN zone and an IP-based network segment.

The ideas and results presented in this chapter are part of the lessons we have learned in INTERSECTION and INSPIRE, two FP7 projects we have been involved in. Additional information is available on the websites of the projects.
Acknowledgments

The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under Grant Agreement No. 225553 (INSPIRE project) and Grant Agreement No. 216585 (INTERSECTION project).

References