Towards Intrusion-Tolerant Process Control Software

Hugo Ortiz Paulo Sousa Paulo Veríssimo
LaSIGE, University of Lisbon, Portugal
ortiz@lasige.di.fc.ul.pt, {pjsousa,pjv}@di.fc.ul.pt

Abstract

The security of critical infrastructures like water, gas or power grid control systems has been discussed more thoroughly in recent years due to recent events that have questioned their security. Terrorist groups are betting on cyber attack methods due to obvious advantages: it is cheaper than traditional methods, it is very difficult to be tracked, terrorists can hide their personalities and location, do the attack remotely from anywhere in the world, affect a large number of people, and finally, there are no physical barriers or checkpoints to cross. One has to understand that, despite some systems being considered secure, attackers will continue to discover new vulnerabilities, to try new attacks and some of those attempts will succeed. One approach to address this problem that is gaining momentum recently is intrusion tolerance. Based on this paradigm, there already are intrusion-tolerant network architectures that enhance the protection of critical infrastructures. However, even using such enhanced protection mechanisms, control systems remain with a certain level of vulnerability, which can be decreased if the process control software (PCS) itself is prepared to tolerate intrusions. This paper justifies the importance of developing intrusion-tolerant process control software and presents some insights on how to do it.

1 Introduction

Industrial control systems (ICS) are becoming one of the most relevant areas in the research of embedded-control applications. In the beginning, these systems were isolated systems running proprietary control protocols using specialized hardware and software. However, ICS are now starting to be similar to IT systems. Widely available, low cost Internet Protocol (IP) devices are now replacing proprietary solutions, which increases the possibility of cyber security vulnerabilities and incidents [26]. ICS are being designed and implemented using industry standard computers, operating systems (OS) and network protocols. Although this is essential to promote corporate connectivity and remote access capabilities, it provides notably less isolation for ICS from the outside world than predecessor systems, creating a greater need to secure these systems [14, 8, 12, 4]. It is a matter of time until hackers understand how to attack control systems underlying critical infrastructures.

It is extremely important to understand the impact differences of compromised systems. If an attacker compromises a home banking service, the system can be quickly recovered, for example, through backup databases. However, if an ICS is compromised, attackers get access to crucial resources, which may be part of critical infrastructures like water, gas or power grid control systems, and their actions will certainly have severe consequences on the

equipment being (mis-) controlled, on the services provided and on the services' clients.

Intrusion tolerance is a new approach to address accidental and malicious faults, such as attacks and intrusions, in complex and distributed systems [30]. The idea is to assume that: systems remain to a certain extent vulnerable; attacks on components or sub-systems can happen and some will be successful; and one has to ensure that the overall system remains correct and operational. Some works have taken this approach when designing group communication (e.g., [25, 6]) or protocols and services for replicated systems (e.g., quorum systems [17, 32], state machine replication [5, 2]], or a hybrid of quorums and state machine replication [1, 7]).

This leads to the idea of handling - reacting, counteracting, recovering, masking - a wide set of intentional and malicious faults (we may collectively call them intrusions), which may lead to the failure of the system security properties if nothing is done to counter their effect on the system state [28]. In short, instead of trying to prevent every single intrusion, these are allowed, but tolerated: the system has the means to trigger mechanisms that prevent the intrusion from generating a system failure.

Based on this approach, the goal of our work is to investigate ways to develop intrusion tolerant software for process control. In this way, a higher system security level can be obtained, since the system will perform its operation even in the presence of attacks and intrusions. The remainder of the paper is organized as follows: Section 2 mainly points out the relevance and the need of ICS security; Section 3 summarizes previous and ongoing work on this subject; Section 4 describes what is missing in order to better protect critical infrastructures; and finally, Section 5 presents some conclusions and directions for future work.

2 A Real Threat

Terrorism is changing. Nowadays, terrorist groups are betting on cyber attack methods due to the set of advantages it offers. First of all, cyber attacks are cheaper than traditional methods. Secondly, since an attack can be done remotely from anywhere in the world, terrorists hide their personalities and location, becoming hard to be tracked. Finally, without any physical barrier or checkpoint to cross, they can attack several targets affecting a large number of people.

Nowadays, a cyber attack can affect a whole country if some critical infrastructure (e.g., power grid) is networked through computers, which is common in most developed countries. In other words, the more technologically developed a country is, the more vulnerable it becomes to cyberattacks against its critical infrastructures.

2.1 ICS Threats and Vulnerabilities

Threats to control systems can come from numerous sources, including adversarial sources such as hostile governments, terrorist groups, industrial spies, disgruntled employees, malicious intruders, and natural sources such as from system complexity, human errors and accidents, equipment failures and natural disasters. The following is a list of possible threats to ICS [21]:

- **Attackers** - they usually break into networks for the thrill of the challenge or for bragging rights in the attacker community.

- **Bot-network operators** - they take over multiple systems to coordinate attacks and to distribute phishing schemes, spam, and malware attacks.
- **Criminal groups** - they seek to attack systems for monetary gain.
- **Insiders** - the disgruntled insider is a principal source of computer crime. Insiders may not need a great deal of knowledge about computer intrusions because their knowledge of a target system often allows them to gain unrestricted access to cause damage to the system or to steal system data.
- **Phishers** - individuals or small groups that execute phishing schemes in an attempt to steal identities or information for monetary gain.
- **Spammers** - individuals or organizations that distribute unsolicited e-mail with hidden or false information to sell products, conduct phishing schemes, distribute spyware/malware, or attack organizations (*e.g.,* DoS).
- **Spyware/malware authors** - Individuals or organizations with malicious intent carry out attacks against users by producing and distributing spyware and malware. Several destructive computer viruses and worms have harmed files and hard drives, including the Melissa Macro Virus, the Explore.Zip worm, the CIH (Chernobyl) Virus, Nimda, Code Red, Slammer, and Blaster [15].
- **Terrorists** - they seek to destroy, incapacitate, or exploit critical infrastructures to threaten national security, cause mass casualties, weaken the country's economy, and damage public morale and confidence.
- **Industrial Spies** - industrial espionage seeks to acquire intellectual property and know-how by clandestine methods.

The following lists vulnerabilities that may be found in typical ICS. Any given ICS will usually exhibit a subset of these vulnerabilities, but may also contain additional vulnerabilities unique to the particular ICS implementation that do not appear in this listing.

- **Policy and Procedure:** this kind of vulnerabilities are often introduced into ICS because of incomplete, inappropriate, or nonexistent security documentation, including policy and implementation guides (procedures). Security documentation, along with management support, is the cornerstone of any security program.
  - Inadequate security policy for the ICS;
  - No formal security training and awareness program;
  - Inadequate security architecture and design;
  - No specific or documented security procedures were developed from the security policy for the ICS;
  - Absent or deficient ICS equipment implementation guidelines;
  - Lack of administrative mechanisms for security enforcement;
  - Few or no security audits on the ICS;
  - No ICS specific continuity of operations or disaster recovery plan (DRP);
  - Lack of ICS specific configuration change management.
- **Platform Configuration:** this kind of vulnerabilities can occur due to flaws, misconfigurations, or poor maintenance of their platforms, including hardware, operating systems, and ICS applications.
  - Platform configuration:
    - OS and vendor software patches may not be developed until significantly after security vulnerabilities are found;
    - OS and application security patches are not maintained;

    - OS and application security patches are implemented without exhaustive testing;
    - Default configurations are used;
    - Critical configurations are not stored or backed up;
    - Data unprotected on portable device;
    - Lack of adequate password policy;
    - No password used;
    - Password disclosure;
    - Password guessing;
    - Inadequate access controls applied.
  - Platform hardware:
    - Inadequate testing of security changes;
    - Inadequate physical protection for critical systems;
    - Unauthorized personnel have physical access to equipment;
    - Insecure remote access on ICS components;
    - Machines with dual network interface cards (NIC) connected to different networks;
    - Undocumented assets;
    - Vulnerable to radio frequency and electro-magnetic pulse (EMP);
    - Lack of backup power to critical assets;
    - Loss of environmental control;
    - Lack of redundancy for critical components.
  - Platform software:
    - Vulnerable to buffer overflows;
    - Installed security capabilities not enabled by default;
    - Vulnerable to DoS attacks;
    - Vulnerable to packets that are malformed or contain illegal or otherwise unexpected field values;
    - Use of insecure industry-wide ICS protocols;
    - Use of clear text;
    - Unneeded services running;
    - Inadequate authentication and access control for configuration and programming software;
    - Intrusion detection/prevention software not installed;
    - Logs not maintained.
  - Platform Malware Protection:
    - Malware protection software not installed;
    - Malware protection software or definitions not current;
    - Malware protection software implemented without exhaustive testing.
- **Network:** these kind of vulnerabilities may occur from flaws, misconfigurations, or poor administration of ICS networks and their connections with other networks [23].
  - Network configuration:
    - Weak network security architecture;
    - Data flow controls (such as access control lists) not employed;
    - Poorly configured security equipment;
    - Network device configurations not stored or backed up;

    - Passwords are not encrypted in transit;
    - Passwords are not changed regularly;
    - Inadequate access controls applied.
- Network hardware:
    - Inadequate physical protection of network equipment;
    - Unsecured physical ports;
    - Loss of environmental control;
    - Non-critical personnel have access to equipment and network connections;
    - Lack of redundancy for critical networks.
- Network Perimeter:
    - No security perimeter defined;
    - Firewalls nonexistent or improperly configured;
    - Control networks used for non-control traffic;
    - Control network services not within the control network.
- Network Monitoring and Logging:
    - Inadequate firewall and router logs;
    - No security monitoring on the ICS network.
- Communication:
    - Critical monitoring and control paths are not identified;
    - Standard, well-documented communication protocols are used in plain text;
    - Authentication of users, data or devices is substandard or nonexistent;
    - Lack of integrity checking for communications.
- Wireless Connection:
    - Inadequate authentication and data protection between clients and access points.

2.2 Common Attacks

Understanding attack vectors is essential to build effective security mitigation strategies. The level of knowledge in the control system community regarding these vectors should increase in order to mitigate these risks. Effective security depends on how well the community of control system operators and vendors understand the ways that architectures can be compromised. The following is a discussion of some attacks that are usually used against ICS [18, 22, 19].

2.2.1 Backdoor Attacks via Network Perimeter

Industrial control system networks as common networking environments possess innumerable vulnerabilities and holes that can provide an attacker a 'backdoor' to gain unauthorized access. A backdoor is an undocumented way to gain access to a program, online service or an entire computer system. They are often simple shortcomings in the architecture perimeter, or embedded capabilities that are forgotten, unnoticed, or simply disregarded. Process control systems, often have inherent capabilities that are deployed without sufficient security analysis and so can provide access to attackers once they are discovered. Usually, backdoors in the network perimeter are the greatest concern (firewalls, public-facing services, and wireless access).

2.2.2 Attack into Control Systems via Field Devices

This kind of system architectures usually support the capability to remotely access terminal end points and telemetry devices through telephonic and dedicated means. Some of these devices are equipped with embedded file servers and web servers to facilitate robust communication of operational and maintenance data. However, since these devices are part of an internal and trusted domain, an attacker will try to compromise them to obtain an unauthorized vector into the control system architecture. Thus, field devices such as remote terminal units (RTUs) are viable targets to be investigated by attackers, during their reconnaissance and scanning phase of the attack. Since the connections between the devices and the control system are not monitored for malicious or suspect traffic, attackers can use the communication protocols to scan back into the internal control network. They can also alter the data that is sent to the control master or change the behavior of the device itself.

2.2.3 Database and SQL Data Injection Attacks

Database applications have become core application components of control systems and traditional security models attempt to secure systems by isolating them and concentrating security efforts against threats specific to those computers or software components. These networks usually comprise independent systems that rely on one another for proper functionality, creating an expanded threat surface. As field devices, the compromise of database applications creates additional resources an attacker can use for both reconnaissance and code execution since they usually interact with other core components of control systems. The information contained in databases makes them high-value targets for any attacker, and the cascading effect of corrupted database content can impact other core components of the system, such as data acquisition servers, historians and even the operator HMI (Human-Machine Interface)console.

2.2.4 Man-in-the-Middle Attacks

Control system environments have traditionally been (or been intended to be) protected from non-authorized persons by air gapping. In these networks, data that flows between servers, resources, and devices is often less secured. Three of the key security issues that arise from assumed trust are the ability of an attacker to (1) re-route data that is in transit on a network, (2) capture and analyze open critical traffic that is in plaintext format, and (3) reverse engineer any unique protocols to gain command over control communications. By combining all of these, an attacker can assume exceptionally high control over the data flowing in a network, and ultimately direct both real and 'spoofed' traffic [1] to network resources in support of the desired outcome. To do this, a 'man-in-the-middle', or MITM, attack is executed. Because the attack is on the control domain, this plaintext traffic can be harvested (sniffed) and taken offline for analysis and review. This allows the attacker to review and re-engineer packet and payload content, modify the instruction set to accommodate the goal of the attack, and reinject the new (and perhaps malicious) packet into the network.

[1] A spoofing attack is a situation in which one person or program successfully masquerades as another by falsifying data and thereby gaining an illegitimate advantage.

2.3 Reported Incidents

Monitoring and controlling this kind of systems is an enormous undertaking, requiring constant supervision. Any single point of failure can disrupt the entire process flow and can potentially cause a domino effect that shuts down the entire system. Control systems that are not properly monitored can greatly affect the economy. Regarding power grid control systems, the following is a recap of some blackouts, which have thus far been attributed to equipment failure and lack of operator training, causing production loss, physical loss, physical destruction, and being responsible for several human fatalities.

On August 25, 2003, more than 100 electric plants were shut down, including 22 nuclear power plants, affecting 50 million people in the U.S. and Canada. This was the biggest blackout in North American history, forcing the closure of 10 major airports, causing the cancellation of 700 flights, and leaving 350,000 people stranded on the New York City subway. A broken alarm at First Energy, a northern Ohio utility company, may have allowed too much to go wrong before technicians noticed the problem. The reversal of power happened so fast that operators did not have time to react, and within about 10 seconds, vast sections of the grid were overwhelmed. The failed lines in Ohio started a cascade that crashed several systems, despite a structure built for this type of defense.[9]

On August 29, 2003, a failure of England's National Electric Grid caused a blackout in Central and Southern London affecting more than 250,000 people, 270 sets of traffic lights, and 1,800 trains. According to the latest findings, there was a fault in the volt system that apparently had not been properly maintained. [20]

On September 28, 2003, a power failure left most of Italy without power for several hours interrupting rail and air traffic and jamming emergency phone lines. Thousands were forced to take refuge in Rome's subways. As investigations revealed more information, it was found that the Italian response was either lacking, or too slow, and that Italian operators had made a wrong decision when coping with the interruption from Switzerland and France. Consequently, a cascade of power line outages resulted within Italy, and along its border. [24]

Although some concerns exist for possible sabotage, the breakdowns are reported not to be the result of terrorist or sabotage attacks. In any case, they demonstrate how much damage can be caused if every system is not properly safeguarded, monitored, and maintained. Those with the appropriate skills, knowledge, and access could generate major catastrophes and greatly hurt the country's economic stance. It is imperative that the critical infrastructure be secured in order to protect the resources of the nation and sustain its economic health.

3 State of the Art

In order to protect these systems, one has to ensure that they operate correctly despite the occurrence of accidental and malicious faults (including security attacks and intrusions). However, it is not just threats from the outside that are posing problems, but those from the inside (e.g., careless or disgruntled employees) as well.

Multiple private and public sector entities, university researchers and other professionals are working to help secure control systems. Their efforts include developing standards, providing guidance to members, reducing systems vulnerabilities, improving service responsiveness, among others.

The I3P [11] (Institute for Information Infrastructure Protection) is a consortium made up of 27 entities managed by Dartmouth College, including academic research centers, government laboratories and non-profit organizations, that was established to address security issues facing the U.S. information infrastructure. In 2005, the institute launched the Process Control Systems Security Research Project (funded by the Department of Homeland Security and the National Institute of Standards and Technology) which focuses on cyber security research on improving the robustness of the information infrastructure in the oil and gas sector. Initiatives completed include a source code checking tool, an intrusion detection and event correlation tool for process control systems, and a tool for building a business case for investing in security.

The LOGIIC [16] (Linking the Oil and Gas Industry to Improve Cyber Security) consortium brought together 14 organizations to identify ways to reduce cyber vulnerabilities in process control and SCADA (Supervisory Control And Data Acquisition) systems. The project goals were to identify new types of security sensors for process control networks, to develop better ways to protect the critical infrastructures and finally to transfer that technology and know-how to actual field operations. The result was a monitoring system based on the very latest commercial enterprise detection and correlation technologies adapted to monitor control networks, which was tested in five vulnerability scenarios based on cyber compromises commonly used in the hacker community.

IRRIIS [13] (Integrated Risk Reduction of Information-based Infrastructure Systems) is an ongoing project that aims at increasing dependability, survivability [2] and resilience of large complex critical infrastructures. The project goal is the development of a novel simulation environment called SimCIP for modeling, simulation and analysis of this kind of infrastructures. Moreover, the development of a Middleware Improved Technology (MIT) to facilitate information exchanges between different critical infrastructures is also on the scope of the project, which can help to prevent or mitigate cascading failures.

MAFTIA was the world's first project to investigate a comprehensive approach for tolerating both accidental faults and malicious attacks in large-scale distributed systems, thereby enabling them to remain operational during attack, without requiring time-consuming and potentially error-prone human intervention. This consortium brought together significant expertise from the fault tolerance, distributed computing, cryptography, formal verification, computer security and intrusion detection communities. Bringing together research groups from different disciplines resulted in novel work that bridged the gaps between those fields in many ways, including the integration of intrusion detection and fault tolerance concepts in the conceptual model, the recursive use of fault prevention and fault tolerance techniques to create trustworthy components, the use of distributed cryptography techniques for secure replication, group communication, authorization and secure trusted services, techniques for building intrusion-tolerant, intrusion detection systems, and techniques for combining cryptographic and formal methods approaches to analyze security protocols [29].

CRUTIAL (CRitical UTility InfrastructurAL Resilience) is a European project within the research area of critical information infrastructure protection, with a specific focus on the infrastructures operated by power utilities, widely recognized as fundamental to national and international economy, security and quality of life. CRUTIAL's innovative approach resides in modeling interdependent infrastructures taking into account the multiple dimensions of interdependencies, and attempting at casting them into new architectural

[2] Survivability is the ability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents.

patterns, resilient to both accidental faults and malicious attacks. The project's results will help in designing and assessing new electric power systems and information infrastructures. Thus, they will enable to reduce the current (unfortunately repetitive) blackouts, in terms of frequency, duration and extent, and provide insights to electric power companies and standardization bodies for exploiting resilience in critical utilities infrastructures. One of the concrete results of CRUTIAL is an intrusion-tolerant distributed firewall, which was developed by some of the authors [31, 3, 27]. This firewall is capable of maintaining correct operation even in the presence of accidents, attacks and intrusions.

International conferences and workshops in this area, like CRITIS (International Workshop on Critical Information Infrastructures Security) or ITCIP (Conference on Information Technology for Critical Infrastructure Protection) are recently beginning to emerge. Such conferences are extremely important, since they bring together researchers and professionals from universities and private companies and public administrations interested or involved in all security-related heterogeneous aspects of Critical Information Infrastructures.

4 What is Missing?

Even if one uses all project results described in Section 3 and all security guidelines presented in Section 2, the maximum we can get is an industrial control system where is hard to penetrate and compromise the controllers. However, an attacker who can get access to a controller (*i.e.*, to the process control software), has the power to destabilize the physical controlled process (e.g., the production of electrical energy, the supply of water or gas). Therefore, our goal is to build intrusion-tolerant PCS (process control software).

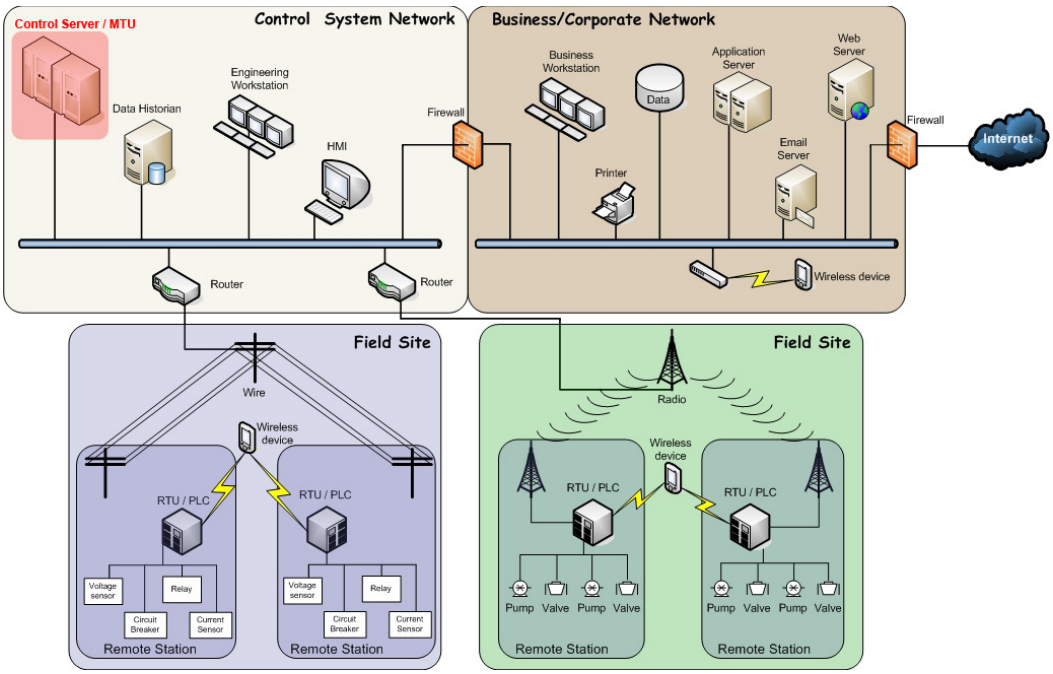

Figure 1: Typical Industrial Control System

Figure 1 illustrates a common implementation of an industrial control system. As we

can see, the control system network houses a control server (also known as MTU - Master Terminal Unit), the communication routers, the HMI (Human-Machine Interface), engineering workstations, and the data historian, which are all connected by a local area network (LAN). This part of the system is known as control center and is responsible for collecting and logging information gathered by the field sites, displaying information to the HMI, as well as centralized alarming, trend analysing and reporting. Furthermore, it may generate actions based upon detected events. The field site performs local control of actuators and monitor sensors. As mentioned in Section 2.2.2, field sites are often equipped with a remote access capability to allow field operators to perform remote diagnostics and repairs, usually over a separate dial up or WAN connection.

What was been presented in the state of the art (Section 3) mostly tries to improve the network security of the system (e.g., intrusion detection, event correlation tools, monitoring systems and advanced firewalls). As depicted in Figure 1, our goal is to make the control server (marked in red) intrusion-tolerant, that is, the PCS running in the control server should tolerate Byzantine faults, namely intrusions. In this way, even if an attacker penetrates the control system network, the control server is able to resist the attack campaign that the attacker will certainly deploy, and continue to perform its operation, albeit perhaps in a degraded mode.

Currently, there are two major techniques for building real time software with fault tolerance capabilities: *recovery blocks* and *N-version programming*. Both are based on traditional hardware fault tolerance and mainly use redundancy and diversity. The basic *recovery blocks* (RB) scheme consists of an executive, an acceptance test (AT), and primary and alternate try blocks (variants). Many implementations of RB, especially for real-time applications, include a watchdog timer (WDT) [10]. The executive orchestrates the operation of the RB technique, that is, it first attempts to ensure the acceptance test by using the primary alternate (or try block). If the primary algorithm's result does not pass the acceptance test, then n alternates will be attempted until an alternate's result passes the AT. If no alternates are successful, an error occurs. *N-version programming* technique (NVP) consists of an executive, n variants (versions), and a decision mechanism (DM). The executive orchestrates the NVP technique operation where n versions execute concurrently. The results of these executions are provided to the decision mechanism, which operates upon them to determine if a correct result can be adjudicated. If one can, then it is returned, otherwise, an error occurs.

However, these techniques cannot be directly applied in the development of intrusion-tolerant PCS because they do not address intrusions. Even if one develops different versions of a certain PCS and builds a system that runs these versions in parallel, an attacker may intrude not only the different versions/variants, but also the RB executive or the NVP decision mechanism. An attacker is an intelligent adversary that will not restrict his actions to the PCS itself, affecting also the behavior of any other components in order to fulfill his goals.

5 Conclusions and Future Work

This paper described the current status of industrial control systems (ICS) security and has shown why is it important to develop intrusion-tolerant process control software (PCS). PCS is the brain of any ICS, and therefore its correctness is vital, namely if it is being used in the context of a critical infrastructure. Current research on ICS security focus on developing security guidelines and/or protection mechanisms that decrease the probability of PCS

being affected by an attacker. However, if an attacker is able to circumvent these protection mechanisms, he will have a clear path to PCS and to the physical process controlled by it.

We have described two classic techniques that can be used to build real time software capable of tolerating (accidental) software design faults. However, these techniques do not allow to tolerate (intentional and malicious) intrusions. We are currently working on how to extend these techniques such that intrusions can be tolerated.

References

[1] M. Abd-El-Malek, G. Ganger, G. Goodson, M. Reiter, and J. Wylie, *Fault-scalable Byzantine fault-tolerant services*, Proc. of the 20th ACM Symposium on Operating Systems Principles, 2005, pp. 59–74.

[2] Y. Amir, C. Danilov, D. Dolev, J. Kirsch, J. Lane, C. Nita-Rotaru, J. Olsen, and D. Zage, *Scaling Byzantine fault-tolerant replication to wide area networks*, in Proc. Int. Conf. on Dependable Systems and Networks, 2006, pp. 105–114.

[3] A. Bessani, P. Sousa, M. Correia, N. F. Neves, and P. Veríssimo, *Intrusion-tolerant protection for critical infrastructures*, Technical Report DI/FCUL TR-07-8, Department of Computer Science, University of Lisboa (2007).

[4] E. Byres, D. Hoffman, , and N. Kube, *The special needs of SCADA/PCN firewalls: Architectures and test results*, In Proc. of the 10th IEEE Int. Conf. on Emerging Technologies and Factory Automation (2005).

[5] M. Castro and B. Liskov, *Practical Byzantine fault-tolerance and proactive recovery*, ACM TOCS **20** (2002), no. 4, 398–461.

[6] M. Correia, N. F. Neves, L. C. Lung, and P. Verissimo, *Worm-IT – a wormhole-based intrusion-tolerant group communication system*, Journal of Systems and Software **80** (2007), no. 2, 178–197.

[7] J. Cowling, D. Myers, B. Liskov, R. Rodrigues, and L. Shrira, *HQ-Replication: A hybrid quorum protocol for Byzantine fault tolerance*, Proc. of 7th Symposium on Operating Systems Design and Implementations, 2006, pp. 177–190.

[8] J.D. Fernandez and A.E. Fernandez, *Scada systems: Vulnerabilities and remediation*, Journal of Computing Sciences in Colleges **20** (2005), no. 4, 160–168.

[9] N. Gibbs, *Lights out*, Time Magazine (2003), 30–39.

[10] H. Hecht, *Fault tolerant software for real-time applications*, ACM Computing Surveys **8** (1976), no. 4, 391–407.

[11] I3P, *Institute for information infrastructure protection*, http://www.thei3p.org.

[12] V.M. Igure, S.A. Laughter, and R.D. Williams, *Security issues in SCADA networks*, Computers & Security **25** (2006), no. 7, 1–9.

[13] IRRIIS, *Integrated risk reduction of information-based infrastructure systems*, http://www.irriis.org.

[14] T. Kropp, *System threats and vulnerabilities [power system protection]*, Power and Energy Magazine **4** (2006), no. 2, 46–50.

[15] J. Leyden, *Why power plants need anti-virus*, The Register (2005).

[16] LOGIIC, *Linking the oil and gas industry to improve cyber security*, http://www.cyber.st.dhs.gov/logiic.html.

[17] D. Malkhi and M. Reiter, *Byzantine quorum systems*, Distributed Computing **11** (1998), no. 4, 203–213.

[18] T. Nash, *Backdoors and holes in network perimeters*, http://www.us-cert.gov/control_systems/pdf/backdoor0503.pdf (2005).

[19] T. Nelson, *Control systems security center common control system vulnerability*, http://www.us-cert.gov/control_systems/pdf/csvul1105.pdf (2005).

[20] BBC News, *Power cut causes chaos*, http://news.bbc.co.uk/1/hi/england/london/3189755.stm (visited January 2008).

[21] NIST, *Guide to industrial control systems (ICS) security*, Second Public Draft (2007).

[22] U.S. Department of Homeland Security National Cyber Security Division, *Control systems cyber security: Defense in depth strategies*, Control Systems Security Program (2006).

[23] McAfee White Paper, *Mitigating the top 10 network security risks in scada and process control systems*, Mcafee (2007).

[24] E. Povoledo, *Most of italy is blacked out for several hours*, New York Times (2003), Section A6.

[25] M. K. Reiter, *The Rampart toolkit for building high-integrity services*, Theory and Practice in Distributed Systems **938** (1995), 99–110.

[26] C. Smith, *Connection to public communications increases danger of cyber-attacks*, Pipeline and Gas Journal **230** (2003), no. 2, 20–24.

[27] P. Sousa, A. Bessani, M. Correia, N. Neves, and P. Veríssimo, *Resilient intrusion tolerance through proactive and reactive recovery*, Proc. of the 13th IEEE Pacific Rim Dependable Computing conference, 2007.

[28] P. Veríssimo, *Intrusion tolerance: Concepts and design principles. a tutorial*, Technical Report 02-06 **4** (2002), no. 2, 46–50.

[29] P. Veríssimo, N. Neves, C. Cachin, J. Poritz, D. Powell, Y. Deswarte, R. Stroud, and I. Welch, *Intrusion-tolerant middleware: The road to automatic security*, IEEE Security & Privacy **4** (2006), no. 4, 54–62.

[30] P. Veríssimo, N. Neves, and M. Correia, *Intrusion tolerant architectures: Concepts and design*, Architecting Dependable Systems **2677** (2003), 3–36.

[31] P. Veríssimo, N. Ferreira Neves, and M. Correia, *CRUTIAL: The blueprint of a reference critical information infrastructure architecture*, Proc. of the 1st International Workshop on Critical Information Infrastructures, 2006.

[32] L. Zhou, F. Schneider, and R. Van Rennesse, *COCA: A secure distributed online certification authority*, ACM TOCS **20** (2002), no. 4, 329–368.