Computer Security Aspects in Industrial Instrumentation and Measurements

M. Lazzaroni and V. Piuri
Dipartimento di Tecnologie dell'Informazione
Università degli Studi di Milano
Crema (CR), Italy

C. Maziero
Graduate Program in Computer Science
Pontifical Catholic University of Paraná
Curitiba (PR), Brazil

Abstract—**Industrial Control Systems (ICS), formerly isolated proprietary systems, are giving place to highly-connected systems, implemented using widespread operating systems and network protocols on public networks. Such standardization trend opens the door to security threats previously restricted to the corporate and personal computing areas. This paper presents how the main concepts of security in conventional computing systems can be exploited in dependability aspects of ICS, and presents some considerations on how security aspects of a standard-based could be applied to a typical control board used in industrial applications involving measurement.**

Keywords - security, safety, availability, dependability, industrial measurement.

I. INTRODUCTION

As computer systems become more pervasive and connected, their security-related aspects should receive increasing attention. Industrial Control Systems (ICS), formerly isolated (or barely connected) proprietary systems, are giving place to highly-connected systems, implemented using widespread operating systems and network protocols in public networks. Such standardization trend is imposed by interoperability and cost reduction needs. However, it also opens the door to security threats previously restricted to the corporate and personal computing areas. This paper presents the main concepts of security in conventional computing systems, discusses their mapping on dependability aspects of ICS, and exposes some considerations about security aspects of a standard-based control board typically used in industrial applications.

This paper is structured as follows: section II presents basic concepts from the classic computer security area, like the fundamental security properties, the notions of threat, vulnerability, and attack, and the main parts of a typical security infrastructure; section III discusses security-related aspects currently found in industrial control systems, presenting some possible types of incidents and frequent approaches to deal with security in such environments; section IV present some considerations on how such concepts could be applied to a simple control board frequently used in typical industrial applications; section V discusses how the computer security concepts traditionally used in the personal/corporate computing area can be mapped in the dependability concepts usually found in industrial environments; finally, the conclusions are given in Section VI.

II. BASIC CONCEPTS

This section presents some concepts important in the computer security field. For a computer system to be considered secure, it should ensure a set of fundamental *security properties*, which may be challenged by *security threats*. The system design should therefore adhere to some *security principles* that dictate how the system should be organized and built, in order to prevent such threats. Moreover, systems frequently present *security vulnerabilities* due to poor design or implementation, allowing *attacks* to be performed against its security properties. This section discusses such concepts in detail and presents the main elements that constitute the *security infrastructure* of a computing system.

A. Security properties

The security of a computer system may be expressed through some fundamental properties [1], [2]:

- *Confidentiality*: the system resources should only be read by users authorized to do so;
- *Integrity*: the system resources should only be modified or even destroyed by users authorized to do such operations;
- *Availability*: the system resources should be available to the allowed users whenever they request to use them.

By system resource, we understand all facilities provided by the computing system, like data, processing power, storage area, local or network services, *etc*. Additionally, two further properties are often related to computer security and may be important in many scenarios:

- *Authenticity*: all system entities are authentic; in other words, data associated to such entities, or provided by them, are true and reflects the real world information that they represent, like user identity, data coming from a sensor, *etc*.;

978-1-4244-2833-5/10/$25.00 ©2010 IEEE

- *Non repudiation*: all actions relevant to security made in the system are known and cannot be denied or hidden by their respective authors.

It is a responsibility of the operating system to ensure such properties for all resources under its responsibility. Such properties may be threatened by human or software errors, or by intentional activities performed by actors inside or outside the system.

B. Security Threats

A security threat is any action that may incur in the violation of one or more security properties in the system. Some examples of threats to the fundamental security properties are:

- *Threat to confidentiality:* a process to scan the memory areas of other processes, or files from other users, looking for sensitive data, like credit card number patterns, passwords, and so on;
- *Threat to integrity:* a process to install a malicious driver or kernel module, allowing its user to control the system;
- *Threat to availability:* a user to request to herself all the system's disk space, preventing the other users to save their files, or to create new files.

Of course, for each possible threat, there should be a mechanism in the system to prevent it, like access control mechanisms, disk/memory quotas, processing priorities, software authenticity verification. Even with such controls, threats may become real security incidents, if there are vulnerabilities that allow them to occur.

C. Security principles

Beyond the software engineering techniques normally used for building correct systems, building a secure system requires respect to other principles, related both to the system building and to the behavior of users and attackers. Some of the most relevant security principles are:

- *Minimum privilege*: all users and programs should operate using the minimum possible set of privileges or access permissions. This way, damage caused by errors or malicious actions are minimized.
- *Complete mediation*: all accesses to resources, both direct and indirect, should be verified by the security mechanism; bypassing such mechanism should not be possible;
- *Secure default*: the security mechanism should clearly identify the allowed accesses; if an access is not explicitly allowed, it should be denied. This principle avoids accesses not foreseen in the system project being accidentally authorized.
- *Mechanism economy*: the project of a protection system should be small and simple, allowing it to be fully analyzed, tested, and validated.
- *Minimum sharing*: mechanisms shared among users are potential source of security problems, due to the possibility of unforeseen information flows. Thus, shared mechanisms should be minimized, for instance, if a given operating system feature can be implemented both as a system call and as a library call, this last form should be preferred, since it involves less sharing.
- *Open project*: the robustness of a protection mechanism should not be based on ignorance or secrecy; instead, the project should be open, its strength depending on only a few items, like passwords and keys. An open project also allows its evaluation by third parties, providing additional assurance of its security.
- *Adequate protection*: each computing resource should have a protection level consistent with its value. For instance, the protection level required by a bank file server its distinct from that required by a small business server.
- *Ease of use*: the security mechanisms usage should be simple and straightforward; otherwise, they will be avoided or bypassed by the users.
- *Efficiency*: the security mechanisms should use computing resources, like processing time and memory space, in an efficient manner, in order to not impact the performance of the system.
- *Weakest link*: the security of the whole system is bounded to the security of its weakest component, which may be the operating system, the applications, the network connection, or even the users.

These principles should serve as guidelines for building, configuring, and running a computing system with security requirements. Most security problems occurring in actual systems come from not observing such principles.

D. Security Vulnerabilities

Vulnerability is a problem present in the specification, implementation, configuration, or operation of a software or system, which may be used for violating a security property in it. Some examples of vulnerabilities can be shown:

- an error in the implementation of a file sharing service that allows unauthorized users to access files remotely from a computer;
- a user account without password, or with a factory default password, allowing unauthorized users to access the system;
- empty disk quotas, allowing a user to fill the entire disk and thus to prevent other users to use the system.

E. Attacks

An attack is the act of exploring a vulnerability to violate a security property in the system. According to [3], [4] there are basically four kinds of attacks to a computing system, as reported in the following:

- *Interruption:* consists in preventing the normal infor-

mation flow among the system users or components; this is an attack against availability;

- *Interception:* consists in having unauthorized access to information; this is an attack against confidentiality;
- *Modification*: consists in modifying the information, or part of the system, violating its integrity;
- *Fabrication:* consists in producing false information or installing malicious components in the system; this is an attack to authenticity.

Furthermore, there are passive attacks, which aim to capture confidential data, and active attacks, aiming to introduce modifications in the system to benefit the attacker or to prevent its use by the legitimate users. Also, attacks can be local, when performed by valid users of the system, or remote, when performed through the network, without using a local account. A program specially built to explore a given vulnerability is called an *exploit*.

Most attacks to computing systems target at increase the power of the attacker inside the system, thus they are usually called privilege escalation attacks. In general, such attacks explore vulnerabilities in system programs that execute with administrative privileges, or in the operating system kernel itself, through system calls.

On the other hand, the denial of service attacks (DoS) aim at lowering the system availability, preventing valid users to access it. Such kind of attack is very frequent in networked environments, to prevent access to web and e-mail servers. In an operating system, local DoS attacks can be easily built by requesting all the local resources, like memory, file descriptors, or disk space, and thus preventing the other users to use them. The system should define limits for the resource allocation, to prevent greedy processes or users to monopolize the resources.

Recently, attacks to confidentiality gained attention. Such attacks are addressed to stealing sensitive user information, like passwords and banking information, with minimal or no interference in the system activities (to avoid being detected). Programs built for that purpose are normally called *spyware*.

There is a clear distinction between an *attack* and a *security incident*. A security incident is any event, intentional or accidental, that compromises one or more security properties. An attack is always intentional. A system intrusion and the accidental leaking of confidential data are both security incidents, but the last one is not an attack.

F. Security infrastructure

Generically, the whole set of software and hardware components critical to the security of a computing system are called its *Trusted Computing Base* (TCB) or *Security Kernel*. All components whose failure may put the system security at risk are considered to be in its TCB. Typical TCB elements include the hardware protection mechanisms (memory segment and page tables and processor operation modes) and the several operating system subsystems that ensure security properties, like file access control and resource quota controls.

Several techniques are used to ensure the security of a computing system. Such techniques are roughly classified in three areas:

- *Authentication*: techniques used to identify users and resources in a system. They can use from simple login/password pairs to digital certificates or sophisticated biometric approaches. In the basic authentication procedure, a user identifies herself to the system; in case of success, a user session is open, for which some system entities are created, like processes or threads, to represent such user inside the system.
- *Access control*: techniques used for defining which actions are to be allowed or denied in the system. Generally the access control rules are defined through an access control policy, which applies to all users and resources in the system.
- *Auditing*: techniques used to register and analyze the activities performed in the system, to for accounting purposes, to detect suspect user's behavior, or to analyze security incidents.

Under a broader view, the trusted computing base of a computer system includes several other items far beyond the operating system itself. The maintenance of the security properties depends on the correct behavior of most of the system components, from hardware to users.

The hardware provides several features essential to system protection: virtual memory mechanisms allow to isolate the kernel and the processes from each other; the software interrupt mechanism allows provides a controlled interface to access kernel services; processor execution modes allow to restrict the instructions and port accessible to the distinct types of code that constitute the system; also, more recent hardware provide mechanisms to prevent executing code outside of certain memory areas, and offer virtualization capabilities to improve isolation among subsystems.

Programming languages also play a role in computer security, since most security vulnerabilities come from programming errors. Strict control of vector allocation and indexing, restricting the "wild" use of memory pointers and restrictive name scoping are examples of features important for secure programming. Finally, applications also have their quota of responsibility in security, by having correct implementations and fully validating all input data. This is particularly important in multi-user applications, like corporate systems and web-based applications, and privileged processes that receive requests from users or the network, like web servers.

III. Security Aspects in Industrial Control Systems

An overview on *Industrial Control Systems* (ICS) and other similar systems when security aspects shall be considered is reported in this Section. In industrial processes, many kinds of ICS are used as, for example, *Supervisory Control and Data Acquisition* (SCADA) systems, *Distributed Control Systems* (DCS), and other further and smaller control system configurations such as *Programmable Logic Controllers* (PLC) [5] or custom-designed control boards. It should be considered that all the aforementioned control systems are often highly interconnected and mutually dependent. This Section provides a brief overview on typical threats and vulnerabilities affecting these systems.

Initially, ICSs were isolated systems typically using proprietary hardware and software (*proprietary solutions*). Nowadays, standard-based systems are replacing proprietary

solutions, which increase the possibility of cyber security vulnerabilities and incidents when these systems are connected in a public network. If the typical features of the IT solutions are now available in ICS, *e.g.* remote access and management capabilities, on line re-configurability, on-line programming, and so on, then typical vulnerabilities found in IT solutions are thus possible and would be considered. This smaller isolation of the ICS from the outside world (*i.e.* the public network) imposes ICS to be more secure. It would be noted that ICSs have characteristics that differ from traditional information processing systems. In fact, operations have effects on the physical world and, in particular, imply significant risk to the health and safety of human lives (not bounded to the operators) and serious damage to the environment, production losses, and compromise of information. Moreover, safety and efficiency often conflict with security in the design and operation of control systems.

The following incidents on ICS are possible, for example:

- Corrupted information may be sent to the system operators, possibly leading to unauthorized changes or to inappropriate and unsafe actions; this is a typical scenario of a data modification or fabrication attack, as discussed in section II.*e*; solutions to this issue involve data integrity and authentication techniques, to ensure that the measurement data is coming from the good sources and were not modified;
- Delay of information flow through the ICS network; this availability problem may be due to incidental failures or intentionally; this last case may also be viewed as a modification attack, because in real-time applications, timeliness is as important as the data contents.
- Instruction, command, and code corruption, or similar integrity problems (as seen in Section II.*a*) that may lead to damage, environmental impacts, and life risk;
- Unwanted and not managed change in ICS software configuration, settings, and operational parameters;
- ICS software infected with malware, opening doors for several kinds of active or passive security problems, as discussed in Section II.*e*.
- Malicious actions may have an impact on the system output. An attack consumes system resources and may alter the behavior of the measurement/control application, leading to unexpected results, like non-compliant items or low quality production.
- Attacks may even have dramatic consequences on the system safety. In particular, protection mechanisms may be prevented from running by an attacker, intentionally or due to an attack side-effect.

Many approaches are conceivable to overcome the aforementioned incidents. In particular, the following aspects can be taken into account:

- A network architecture which includes firewalls, to prevent network traffic from passing directly between the corporate and ICS networks.
- Separate authentication mechanisms and credentials for users in the corporate and ICS networks, as dictated by the *minimum privilege* and the *adequate protection* principles.
- Using a network topology with multiple layers, in which the most critical communication occurs in the most secure and reliable layer, also according to the *minimum privilege* principle, as discussed in previous Section II.*c*.
- Restricting physical access to the ICS network and related devices or improve the security level using a combination of physical access controls with locks, card readers, and/or guards, and so on for example. This means complying with the *complete mediation* principle even for physical accesses.
- Protecting individual ICS components from exploitation. This includes quickly deploying security patches, after testing them under field conditions; disabling all unused ports and services (this approach is also used in good practices electronics design with different motivation); restricting ICS user privileges to only those required for each user's role; tracking and monitoring audit trails and using security controls like antivirus software and file integrity checking software where technically feasible, to prevent, deter, detect and finally mitigate malware. It would be noticed that such high level solutions are often not possible in proprietary, custom designed control boards, which are very popular in low-cost solutions. However, there are many situations in which low-cost boards are used to control large and complex systems. Security threats to such proprietary control systems may also have dramatic impacts on the system's safety requirements.
- It would be noticed that in case of adverse security conditions the system should be able to maintain the correct functionality or to drive itself back to a secure (and obviously safe) condition.
- Redundancy techniques such as the *k* out of *n* approach, already applied to improve the availability and reliability of a system, can also be useful to improve its security. Using such approaches, security problems in a given component can be easily isolated or just ignored. Furthermore, redundancy is a key approach for system dependability, as discussed in section V.
- A consideration is here mandatory: incidents are often inevitable. Thus, an incident response plan should be clearly defined in advance. In case of a security incident, the corresponding response plan should be operated as soon as possible to bring the system back to its normal operation.

Finally, wireless networks are becoming a reality in distributed measurement and control systems. Security aspects in wireless networks are roughly similar to those of wired ones, except that physical access to a wireless network is much easier. Due to this, wireless networks are more vulnerable to attacks such as traffic monitoring (confidentiality risks), packet injection (inte-

grity risks) and traffic flooding (availability risks). Strong authentication and traffic encryption help to preserve the measurement/control system security, but they may impose an excessive processing cost for small sensors and actuators.

IV. Security Aspects on a Control Board

Now, the authors are interested in studying the aforementioned security aspects when a simple custom designed control board, used for industrial control purpose, is connected to a network. In particular, this Section is devoted to discuss how the security aspects may affect or impact the operation of a control and data acquisition board typically used in instrumentation and control systems, as depicted in Figure 1. Measurement aspects are very important due to the fact that measures are used in decision processes, so corrupted or missing data may lead to incorrect decisions.

Such application scenarios frequently use simple control units based on low-cost microcontrollers. These device are very popular in industrial control. In fact, microprocessor requires external support chips but microcontrollers are a single chip solution. Microcontrollers have on-chip non-volatile memory for program storage, interface functions on-chip (*e.g.* serial interfaces, Analog-to-Digital conversion capability, timers) but also lower performance than microprocessor. For instance, a typical control board is equipped with an 8 or 16-bit microcontroller (but 32-bit microcontroller are also used), 16K to 64 KB of internal memory, some digital and analog input/output lines, relays, interfaces for common communication buses (CAN, I^2C, *etc*.), and other I/O channels (USB, Ethernet, serial, RS-232, RS-422/485, specific sensors, *etc*). Some boards are also equipped with a basic user interface (LCD displays, push buttons, LEDs). In very simple boards, jumpers and dip-switch are often used in order to set function and to configure the boards.

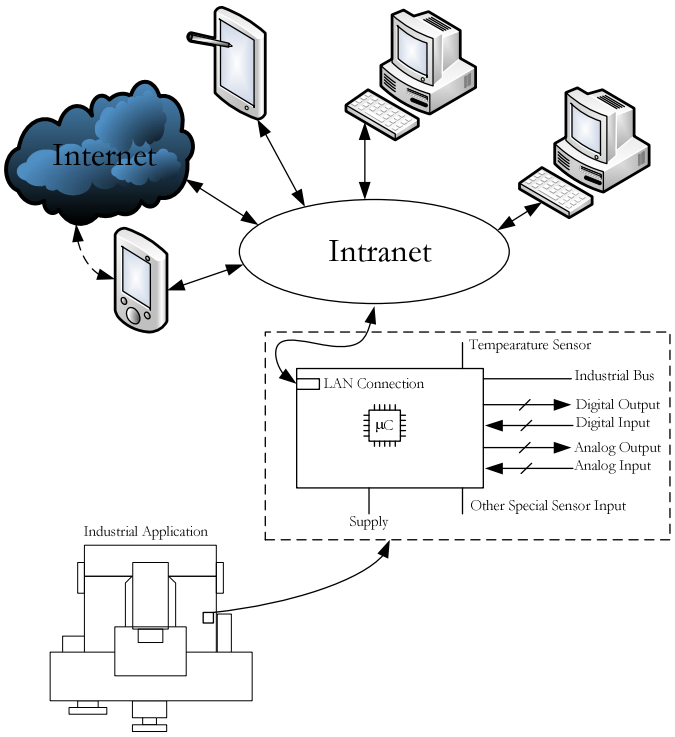

Figure 1. A typical instrumentation scenario.

Control boards, also the smaller ones, can be frequently connected to an Ethernet-based network. This possibility allows integrating the low-level control aspects to higher-level information systems using standard networking protocols and services, like TCP/IP, HTTP, and the Internet. However, some precautions should be taken before deploying such a system.

The first step is to test the system in a controlled environment, to minimize concurrent network activity that could interfere with the board communication. It should be noticed that it is not infrequent that the insertion of a single device/board compromises the integrity or privacy of a network and its sensitive information, thus it is mandatory to perform extensive testing with new equipment/board before adding it to a secure network. Furthermore, it would be considered that even simple microcontroller-based boards are able to generate a high volume of network traffic.

Such a simple control board could be used in order to analyze the possibility of security incident situations and scenarios like those discussed in the previous sections.

V. Discussion

This Section brings a discussion concerning how the aforementioned aspects can be related to the system dependability. Dependability is the ability to deliver service that can justifiably be trusted. This particular definition of dependability stresses the need for justification of trust. A further definition of dependability can be here given: dependability of a system is the ability to avoid service failures that are more frequent and more severe than is acceptable to the user. In dependability, the following attributes are present: *i*) availability: readiness for correct service; *ii*) *reliability*: continuity of correct service; *iii*) *safety*: absence of catastrophic consequences on the user and the environment; *iv*) *confidentiality*: absence of unauthorized disclosure of information; *v*) *integrity*: absence of improper system alterations; *vi*) *maintainability*: ability to undergo modifications and repairs. For the considered application, *security* is the concurrent existence of availability, confidentiality, and integrity. Thus, it is easy to see that threats to security, as discussed in Section II, are also threats to dependability, as they affect dependability attributes. Figure 2 shows the complete taxonomy of dependable computing [6].

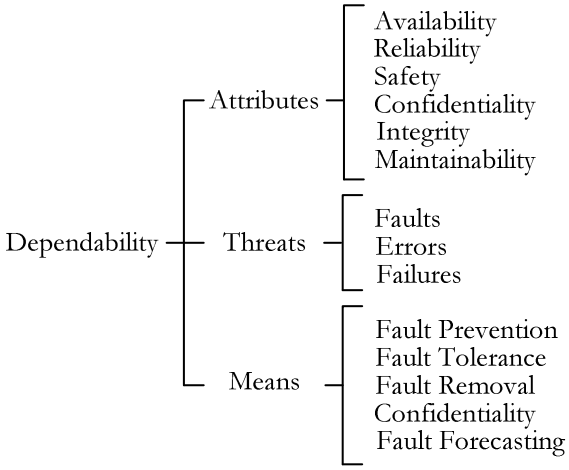

Figure 2. The dependability tree tipically used in IT application.

All aspects investigated in this paper can be seen as dependability problems, in particular for the availability point of view.

VI. CONCLUSIONS

As industrial instrumentation and measurement systems evolve, they aggregate more sophisticate software, both applications and operating systems. Thus, security threats traditionally present only in corporate/personal computing now represent a real menace for such systems. In this paper a parallel between conventional computer security concepts and industrial systems' dependability have been presented.

REFERENCES

[1] R. Sandhu, and P. Samarati, "Access Control: Principles and Practice," in IEEE Communications, September, 1994..

[2] Lichtenstein, S., A review of information security principles. Computer Audit Update, 1997(12): 9–24.

[3] Pfleeger, C. and Pfleeger, S. L., Security in Computing, 4th Edition. Prentice Hall PTR, 2006.

[4] Saltzer, J. and Schroeder, M., The protection of information in computer systems. Proceedings of the IEEE, 63(9):1278 – 1308, 1975.

[5] Guide to Industrial Control Systems (ICS) Security, Special Publication 800-82, National Institute of Standards and Technology Special Publication 800-82 (final public draft) Natl. Inst. Stand. Technol. Spec. Publ. 800-82, (September 2008).

[6] A. Avizienis, JC. Laprie, B. Randell, and C. Landwehr, IEEE Transactions on Dependable and Secure Computing, vol. 1, n. 1, January-March 2004, pp. 11-33.