BEST PRACTICES IN THE DESIGN OF A SECURE CONTROL SYSTEM

S. Poulsen

CERN, Geneva, Switzerland

ABSTRACT

Process controls systems play an important role in the operation of physics laboratories. They are also ubiquitous in society at large, where complex processes, equipment and machinery needs to be operated, controlled and monitored. Control systems are essential for tasks as diverse as electricity distribution and ventilating your office.

Long ago, these systems were simple electrical-mechanical or analogue electronic devices built around or rather deeply integrated with a process and its equipment. Now these systems have evolved into full-blown information management systems, heavily integrated with organization-wide networks and business IT systems. This has resulted in a new set of IT security risks that must be dealt with in the specific context of process control. This paper will deal with some of the relevant issues that have emerged from practical experiences at CERN. Since process control systems cover many different processes and technologies, the paper attempts to be rather general, and highlight the main domains where an effort is required. These domains are part of more general IT security management frameworks, in particular the ISO/IEC 17799 standard. The paper illustrates how this standard applies to issues of a modern process control environment, just as it provides guidance in general purpose IT management. As process control "goes IT", so should the management processes involved.

INTRODUCTION

Originally, controls systems were implemented as stand-alone and isolated systems. They were supplied by vertically integrated companies selling solutions based on proprietary equipment and hardware. When software was part of the supply, it was the vendor's software platform. Access to these systems was limited to physical access and the systems were protected against unauthorized access via simple perimeter controls such as key-based systems. There was little scope for logical access. Signal transmission, via hardwired contacts or even field-buses, had physical limits in the order of hundreds of meters. The main risk was someone cutting the wires.

New design trends and implementations have become prevalent, just as they have in the field of general computing. This is a combination of what is technically possible, what is economically feasible and what is required by different stakeholders. Now, the architecture is modular with simple, powerful, standardized interfaces. Some modules have emerged as *dominant designs* that offer seducing functionality, are easy to use and cheap to deploy. We now implement control systems using "office" IT technologies such PC operating systems, IP/Ethernet networking and commodity processors. These are all becoming effectively *de-facto standards* in modern process control.

Modern systems also offer many expansions: They come with open interfaces to third-party systems (e.g. via OPC interfaces). Suddenly, other parts of the organization integrate with the control system, which now becomes part of other business management systems. Think of control systems that link to the maintenance management system, and automatically trigger preventive maintenance after a defined time of operation. In a commercial company, it could also be a matter of creating links between the logistics of the production and the orders generated by e-business customers.

The control system is slowly becoming more of an *information management* system and starts to resemble any complex IT *infrastructure*. Its most important modules go beyond what is strictly needed to control the process. It covers networking, databases and applications; it includes Intranet and Internet applications for information dissemination. Process devices like Programmable Logic Controllers (PLCs) become information servers with implementations of standard IP-based protocols. Thus, control systems also have users that are not linked directly to the process; where the users were before limited to trained operators, working in the vicinity of the equipment, any information "consumer" may now have access to the process independently of his location.

IT SECURITY IN PROCESS CONTROL

As process control systems converge into information management systems, and as they use common-off-the-shelf (COTS) IT infrastructure components, IT security becomes relevant. There is more room for errors, intentional or un-intentional damage or even information theft. For now, there is limited understanding of these risks. Typically, an organization's IT security is well managed by experts, but they do not always fully appreciate the issues specific to control systems. For this, the process specialists have the necessary knowledge. However, these are most often not IT specialists. There is potentially a lack of pre-emptive or preventive treatment on behalf of the people ultimately responsible for deploying and operating these systems.

Improving this situation is in fact feasible. Management tools are already available that can serve as *frameworks* or *standards* for the treatment of information risks. One example is ISO-17799 (BS-7799). These standards are specifically developed for traditional information management systems, and do not treat specific issues of control systems. However, they can effectively assist when treating the intrinsic security risks of control systems. There are also well established methods to deal with classic safety and process security issues in designing critical control systems, such as Failure Mode Effect Analysis (FMEA). These methods should still be deployed, but this paper suggests they be *complemented* by IT management tools. In parallel, formal approaches are emerging to *label* commercial control systems with a security label, just as it is done for IT equipment [2].

OBJECTIVES

Typically, the overall security objectives of generic IT systems are specified as the *triade* of confidentiality, integrity and availability (CIA) as defined in ISO-17799. Incidentally, these objectives can also describe the objectives of a process control system, even a classic system without IT technologies. Traditionally, the main objectives are the availability and integrity of the system. Confidentiality was less of an issue in the past, since there was not much information to protect, and only the immediate operators could access it. Now, as process control systems are also information management systems, and potentially make this information available to people *off* – as well as *on* - the organization's premises, confidentiality can become relevant. Even a physics laboratory may want to limit what is published.

We have a *qualitative* definition of objectives in terms of the *triade*. An important addition, would be to define these objectives in *quantitative* terms. The expected availability of a control system should be described, since it can actually be measured. Since it can be measured, it can also be managed. Typically, a process control environment must operate around the clock; availability can be expressed as a number of "9"s (e.g. 99.9%, 99.99%, etc), or simply as the number of service interruptions. Actual performance can be compared to baseline performance, and actions can be taken to improve a system that falls short of the predefined requirements.

It is also possible to make simple measurements of the level of integrity and confidentiality for a given system. One could imagine counting the number of integrity and confidentiality *breaches* over a given period. This seems obvious, but many organizations do not include process control in IT incident statistics, and the performance of the process control systems may not be followed. This tendency is due to change. When control systems become directly linked to the "business" objectives, an agreed level of for instance availability must be supplied to a client, and this agreement will eventually be formalized in a *service level agreement*, even within organizations.

RISKS

In process control, as in IT at large, one should deal with IT security as part of a broader risk management process. This process starts by identifying the risks; creating and maintaining a catalogue of events that might cause harm to the process in the largest sense. It continues by deciding how to *mitigate* the risks, implementing these measures and evaluating their effectiveness. It is essential to establish a clear link between a risk and – when judged necessary – the *security control* reducing the risk. This should be a *cost-benefit analysis*; there is no reason to purchase and install anti-virus software on a SCADA[1] PC of a control system, if it is impossible that the system be attacked by viruses. If this risk exists, the cost of the anti-virus software must be balanced against the cost of an

[1] Supervisory Control And Data Acquisition; commonly deployed in small and large process control systems

infection, both in terms of direct damage to the computer, but also in terms of indirect damage to a process managed by this computer. There are methods to deal with this quantitatively. That is, assign average yearly monetary values to the risk with and without implementing the security control. I.e., "*on average virus-infections will cost X without anti-virus and Y with anti-virus*". This allows balance the benefit with the cost of installing the anti-virus package. At least rough estimated values (or scales) should be derived; neglecting this, we risk misallocating funds within limited budgets, even in not-for-profit organizations, such as physics laboratories.

Old and new risks

There are many classic risks in control systems. They cover failure in the process (e.g. the supply of raw materials in a production environment, the equipment (e.g. a motor) or the control equipment itself. They also include manual operation errors, and physical hazards like fires and floods. A modern control system – built from IT components - with the many features and functionalities that it inevitable offers, introduces new risks. These risks arise from a combination of *threat agents* and *vulnerabilities* in the systems being deployed. When identifying risks, it is important to consider both aspects. The threat agents are people in- and outside the plant that knowingly or unknowingly expose the systems to information theft, hacking attacks, and virus and worm infections. The vulnerabilities present in the newly deployed control systems allow these attacks to take place. They may be absence of anti-virus software on PCs, un-patched software (operating systems and applications), weak network passwords or incorrectly configured network services (e.g. file sharing).

These new risks have both a *direct* and an *indirect* component. There is a direct risk of compromise to the control systems. An inventory must be made of these risks; some are general and well-known to PC based systems. Information is constantly inventoried in the databases such as the NIST NVD[2]. Others are specific to networked process control equipment. They are less published, but it is essential to stay informed about these risks and, evaluate their impact on the organization's process control systems. One of the leading specialised vulnerability databases is the BCIT ISIK[3] (subscription based).

It is also essential to appreciate the indirect risks; they may be the more important. The IT security experts can only comprehend these with input from the process specialists. Indirect risks could be production delays during downtime after a system compromise, and the image loss and customer liability that could result. On a broader horizon, in a process environment, indirect risks could include legal liabilities, if neglecting potential process control failures that would result in pollution.

SOLUTIONS AND BEST PRACTICES

A number of *best practices* are proposed in dealing with and mitigating new information risks related to emerging threats and vulnerabilities in process control systems. This is not intended to be a complete inventory. Rather, it is some items that can be considered the most important and essential. However, maybe the most important practice is constantly to review this, and adapt to a rapidly evolving security landscape. Such reviews may form parts of formal and regular *audits*, just as it is done in general business IT environments.

Strategies

Standardizing: In this context, it means standardizing on the modules of the process control system, such as PLCs, OSs or application programs (e.g. SCADA software). To manage the inherent risks involved with using these modules, it is necessary to perform a number of tasks for every type and product deployed. This is to identify potential vulnerabilities, optimise configuration parameters, etc. Ideally, it is a formal security assessment and certification of the module and its baseline configuration within the organization. There is also developer and operator training, as well as maintenance contracts to set up, to assure due delivery of any security patches and other updates. As this is done for every new product, there are economies of scale to exploit, by leveraging the already approved and known products. The optimal level of standardization may not be complete standardization, that is, *one* product or *one* supplier. Even for security purposes, it may be advantageous to keep options and some kind of competition between suppliers, to apply pressure to solve problems and respond to the customers' security concerns. Spread your bets.

---

[2] National Vulnerability Database, National Institute of Standards and Technology, http://nvd.nist.gov

[3] British Columbia Institute of Technology, Industrial Security Incident Knowledgebase, http://www.bcit.ca

Consolidating: This covers different aspects, such as centralization of disparate information systems on a reduced set of servers and OS platforms. Since this means a reduction in the amount of hardware and operating systems to manage, it often also brings some level of standardisation. It offers advantages, such as managing less individual machines, meaning fewer machines to patch and less hardware to protect from hazards and humans. By using virtualization technologies, and managing legacy hardware and software platforms as *virtual machines,* problem diagnostics or even re-deployment can be managed remotely and lead to shorter down time, which is critical in around-the-clock process control. Importantly, it allows for optimized usage of under-utilized hardware platforms, and purchasing less hardware is in alignment with most organizations' strategy of reducing cost.

Minimizing: The need to limit a process control system to its essential functions is often neglected. *Function creep* during development, deployment and the complete lifecycle means increasing complexity, which again means less security. A process control systems may start out being simple, but then gets integrated with other information management systems. This creates new vulnerabilities that are not identified before they become security incidents. Priorities sometimes change, so that the process control becomes a side issue and the general information management aspects dominate. An example would be once stand-alone PLC that gets equipped with an Ethernet interface, connected to the corporate network and serves Web clients to interact with its I/O boards. This risks opening up the PLC to unauthorized accesses, and compromises its stability if it suffers from network vulnerabilities.

Maintaining: In a process environment, it is generally accepted that a maintenance program be implemented for hardware that requires certain preventive and corrective actions as recommended by the supplier. New technologies involving software are now integrated, but the maintenance aspect is often overlooked. Software is not subject to tear and wear, but still needs regular maintenance due to discovery of unknown flaws. In contrast to hardware, which is sold with a guarantee, software is normally sold with a disclaimer discharging the manufacturer from any responsibilities (even if you subscribe to software updates). Even if software does initially work, it becomes obsolete. This is part of the manufacturers' strategy, to ensure constant revenues by introducing new versions with new features, and declining to mend the old ones, thus forcing the client into new purchases; software is not a one-off purchase, but rather a recurrent yearly expense. As a consequence, the cost of integrating new technologies in process control is higher than one might expect, and this should be evaluated when designing the system and evaluating it for economic feasibility. The problem with updates, when you buy them, is *how* and *when* to patch the process control system. In a traditional IT environment one generally wants to stay completely up-to-date with the latest patches. In a process control environment, this may not be desirable, since patching inherently means unavailability, and a risk of subsequent problems. Thus, the system architecture must allow running the process control system in a safe manner, *without* being up-to-date. Workstations that are included in centralized patch distribution schemes must not be part of the core functions of the process control, if downtime is not acceptable.

System Architecture

The system architecture is initially decided upon during the design process, and is paramount to the security of the system. If problems are discovered during operation, they may be difficult to solve if the solution jeopardizes the initial architecture. The system architecture – as well as all other design steps - should take into account the global objectives of the *triade*. The starting point is a modular or rather *layered* design, and for this we can borrow from classic security architectures, physical or logical, that organizes the system into layers, as shown in Figure 1. The *core* layer of the system is the basic process functionality. It must be as simple as possible, to let us verify it – let alone understand it. It must also be as protected as possible, since it means most to availability and integrity. This layer should be reduced to real-time field-buses, I/O functions, process regulation, safety functions and local human interfaces for the basic operation. The most essential process and user requirements must be implemented by the core. Anything that could potentially compromise core functions must be excluded; if a network service on a SCADA PC is not required, it should be excluded.

Outside the core, one or more layers of extra functions could be placed; everything *not* fit for the core should be located here. Remote SCADA systems, data logging and control room panels could be delegated to the *options* layer. When there are interactions between core functions and peripheral functions, it must be via a module that supervises and authorizes this. In a networked infrastructure, where the different layers integrate via IP networks, such a control mechanism could be implemented

via *firewalls* or *proxy* services. Also, we may have a layer of *non-essential features*, to be placed in the outer-most layer, which can be offer integration with existing and future business information systems – e.g. access to maintenance and operation data - while being completely isolated from the core process. Any system must also from the beginning be open for new extras, without impairing the design, through standard interfaces such as OPC or XLM/SOAP-based protocols.

These abstract different layers must be reflected in the network design. They should be implemented on separate networks, with strict access controls. Increasingly, all layers will be based on IP/Ethernet technology, with proprietary field-buses going out of fashion. While separating them logically (e.g. via VLANs) - or even physically - requires extra devices to act as gateways between separated network, these devices are now commonplace in all network environments and ever easier to deploy. Following this principle means applying *defense-in-depth*. A starting point is separating the organization's network (almost) completely from the Internet, via *perimeter* security controls, but we must also make additional barriers between the general business network and dedicated process networks, possibly even isolating individual applications or processes.

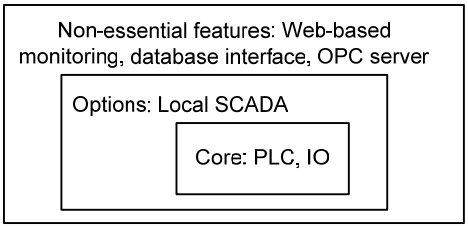

Figure 1 - A layered system architecture

User management

Designing a control system means identifying users and user roles. Typical usage scenarios must be described using *use-cases*, and depending on this, user privileges should be adapted. A user assuming a specific role (e.g. operator) may have different privileges depending on his physical location. Switching on and off equipment may require physical on-site presence, whereas reading values can be done from anywhere. The objective is to prevent deliberate attacks but also user errors. In some systems, it is possible to define *restricted* privileges on remote web-based process panels, which have *full* functionality when used locally. When attributing user rights, care must be taken to organize for integrity and confidentiality of the system. It means implementing security paradigms such as *least privilege* and *need-to-know*: General operators cannot change process parameters (it is the exclusive right of the process engineers) and non-authenticated network users do not have authorization to accessing sensitive information such as production formulas.

Having established the user rights model and a layered architecture, it would be possible to define a multidimensional *access control matrix*: *who* can do *what*, *where*, and *when*. An essential part of the subsequent implementation would be locking down the process control system, so that users *do* effectively have limited privileges. Too often, an application may restrict users' access to data, only to let them access it (let alone delete it) directly via the operating system. New systems offer increasingly sophisticated facilities to prevent the user from straying outside his *sandbox* (his authorized area) but the resources and time needed to set this up – and verify the correctness of the configuration – are often neglected in the project plan. A related problem is that suppliers implementing process control systems lack the necessary qualified personnel.

Access control

It is normally good practice to define rigid *physical* access controls in a process environment, based on the different users and their needs. It serves to protect both the process but just as well the users: in a physics lab, without access restrictions people could inadvertently approach a radioactive source. This is well covered by legal requirements but *logical* access controls are far less well defined and applied today. Some environments define a *single* user for network access to the process without any access restrictions. Technical solutions are now available to manage individual users and map them to the access roles previously defined, and these solutions should be deployed. Still, *generic accounts* – which do not relate to an individual user - may be needed in some environments. Think of the *root* user in the UNIX operating system. Such accounts must not be accessible directly, but the user must first perform individual identification *before* escalating his privileges. Some suggest protecting generic

accounts by limiting them to read access. This may not be satisfactory, when it does not satisfy the confidentiality criterion. Finally, the correct authentication service should be selected based on the criticality of the process. Often a *password* is enough, but increasingly it is vulnerable to *cracking*, *sniffing* and *social engineering* attacks. Since people in a process control environment often need badges to obtain physical access, it would seem a natural step to extend this to logical access control. Fortunately, systems that control network access via a badge or *smart card* are becoming commoditized. Networked equipment where access cannot be strictly controlled must not be integrated directly on a big network, if it is not protected from anonymous users or devices. This applies to PLCs, which have rudimentary and vulnerable networking implementations. Such systems – often deployed to assure the availability and integrity of the process – must be part of the core layer, and only available via firewalls or proxy services and gateways, which *do* offer authentication services.

To offer the highest level of assurance, the authentication services should be based on systems already deployed in the company, such as for the general-purpose administrative and office networks. Integrating the process control computers in these infrastructures means leveraging existing systems and avoiding local, distributed account and password management. The down-side is that when the network is down or the services otherwise unavailable, access may be limited to the process control system. Due to the criticality of the networks and associated services, there are strong pressures to make them ever more robust and highly available. If some level of process intervention is still required in the unlikely event that network services are down, a redundant or degraded operation interface must be available in the core layer, which is physically accessible. In most situations, the advantages of managing users and privileges rigidly with central services outweigh the limited risks of unavailability of non-core functionality.

Management and non-technical issues

More important than most *technical* security controls, is making people within process control aware of the risks they face when incorporating application software, operating systems and networks, and including the top management. These modules are just black boxes, and new functionality is often incorporated at the expense of security, which only becomes apparent over time. Users must understand that the up-front cost of deploying these new solutions is only a small part of the life-time cost of operating the systems, something that the vendors are all too aware of. End-users, system designers and developers need education and awareness. So do project managers and decision makers. When incidents do happen, it is often difficult to see who to blame, but ultimately upstream decisions may have impaired the process quality. These decisions fall with the management, who should act proactively and try reducing risks where process quality is important. They must show due diligence and assure that IT security takes into account the problems and concerns specific to the process control environment. The management structure must be in place to assure that the corporate wide security policies are available, which will provide guidelines when systems are designed, operated and evolve.

CONCLUSION

IT security management within the domain of process control is emerging and not yet mature. Governments have expressed concerns over the vulnerability of critical infrastructures, e.g. electricity distribution, to hostile cyber-attacks [1]. However, inside the organizations that assume responsibility for these systems, the full action is not yet taken. It is not for lack of will, but legacy control systems are a lot more difficult to change than run-of-the-mill IT systems (that can be difficult enough). A process control system traditionally lives along with the process equipment, which can last several decades. The philosophy was: "*Why change it, if it is not broken?*". Now, many systems are potentially broken, and need remedies to mitigate computer security risks. To control IT security risks, it is ultimately necessary to consider carefully to which extent and how IT components are integrated with the process, and to evaluate the benefits with the risk and cost over the life time of the system. We can only hope that action is taken before the lights go out.

REFERENCES

[1] United States Computer Emergency Response Team, Control Systems Cyber Security Awareness, July 2005, http://www.us-cert.gov/reading_room/Control_System_Security.pdf

[2] Digital Bond, Control Centre Protection Profile, submitted to NIST Process Control Security Requirements Forum, http://www.digitalbond.com/SCADA_security/Control_Center_PP.htm