Designing Modular and Redundant Cyber Architectures for Process Control: Lessons learned

Paulo Verissimo Alysson Neves Bessani Miguel Correia
Nuno Ferreira Neves Paulo Sousa
Universidade de Lisboa, Faculdade de Ciências - Portugal
{pjv, bessani, mpc, nuno, pjsousa}@di.fc.ul.pt

Abstract

An architecture was recently proposed to protect the power grid, in the context of a European project. The design of the architecture, guided by an analysis of the evolution of critical information infrastructures, tried to be as generic as possible, with a view of possibly serving as a reference cyber architecture for process control infrastructures.

The need for a new architecture is explained by the fact that cyber architectures for process control, despite being basically physical processes controlled by computers interconnected by networks, exhibit a potentially huge cost of failure in socio-economic terms, thus bringing extremely demanding requirements, which have not been previously found together in a same class of computer-based systems.

In this paper we wish to report the lessons learned in the development, analysis and evaluation of the proposed cyber architecture for process control.

1. Introduction

An architecture [5] was recently proposed to protect the power grid, in the context of a European project. The design of the architecture, guided by an analysis of the evolution of critical information infrastructures, tried to be as generic as possible, with a view of possibly serving as a reference cyber architecture for process control infrastructures.

Some years ago those systems were highly isolated, mostly proprietary, and hence, secure against most threats and reasonably robust against accidental faults. During the last years these infrastructures have undergone significant interconnection and computerisation, which created an enormous progress in the efficiency and effectiveness of their management. However, this web of critical information infrastructures also became greatly exposed to cyber-attacks coming from the Internet [2, 3]. On the other hand, their complexity increased by way of the added computer and network machinery, increasing the likelihood of accidental computer-generated faults pervading the control system.

However, this scenario has not relieved any pressure with regard to the introduction of more computer and network-based services. On the power grid side new challenges loom, such as distributed generation or smart metering. The same is expected to happen with other control system sectors. It is unthinkable for this evolution occur without heavy incorporation of ICT (information and communication technologies) at large, and most specially, interconnected computers.

We are witnessing the accelerated mutation of control system infrastructures to computer-electrical or cyber-physical systems. Where wanted or not, it is as inevitable as a preceding mutation, more than twenty years ago, of old POTS infrastructures toward computer-telephone systems. Systems are no longer closed, proprietary; they are connected to the Internet and often use common operating systems. As such, the risks they incur may drastically increase, if the problem is not tackled with the adequate weapons.

1.1 Why do we need a new architecture?

Cyber architectures for process control are basically physical processes controlled by computers interconnected by networks [1]. However, the value of cyber-physical infrastructures to society is incommensurably larger than that of common ICT systems (commercial, finance, etc.), and the socio-economic impact of their failure can correspondingly be huge. Progressively, modern societies discovered that threats against computers and control computers could have as devastating effects as attacks on the physical infrastructures themselves.

A word of caution is thus in order. The requirements on cyber architectures for process

control infrastructures are that they are dependable and secure against cyber-attacks and computer-generated faults. It is expected that they do so in an unattended way for a large extent of the architecture, and operate perpetually, in some cases literally non-stop, in all others, with very short unavailability periods. Furthermore, they must exhibit resilience against unexpected situations and/or when stressed beyond the design-time envelope.

These requirements impose constraints that mix the dynamics, adaptability and uncertainty seen in common ICT infrastructures, with the rigidity and predictability of typical smaller-scale critical control systems. These requirements have not been previously found together in a same class of computer-based systems. Moreover, many conventional security and protection techniques can bring serious problems, when directly applied to controlling devices, by preventing their effective real-time operation. In fact, we predict [5] that most of the remedies that are (and well) currently being put in place to bring cyber architectures for process control at least at the same level of security and dependability of common ICT systems (e.g., conventional firewalls, intrusion detection, or vulnerability scanners), will barely sustain the next impacts of cyber-attackers, who are becoming everyday more effective.

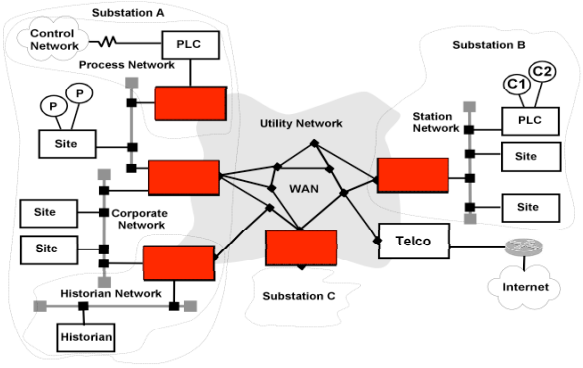

Figure 1. Process control system cyber architecture

These were the challenges we have tried to meet, hence the proposal of a new architecture. In this paper we wish to report the lessons learned in the development, analysis and evaluation of the proposed cyber architecture for process control.

2. Cyber architecture

Pretty much as any large-scale control system infrastructure, the power grid is formed by facilities (power transformation substations, corporate offices, etc.) interconnected by a wider-area network (WAN). In [5], we propose to represent facilities through protected LANs, all interconnected by a WAN, leading to a WAN-of-LANs architecture, as depicted in Figure 1. Then, we propose mechanisms to make this WAN-of-LANS interconnect incrementally resilient: LANs can be assigned different levels of trust, and be endowed with protecting machinery that offers different levels of trust (versus, obviously, different levels of cost and complexity).

Our claim that this should be the primordial approach to resilience of cyber architectures for process control, or in fact, of critical information infrastructure (CII) architectures in general, is based on the following observations:

Perimeter security is not sufficient in modern threat scenarios, which include insider intruders --- this architecture offers the right modularity by defining the LAN as the unit of trust, and of intrusion thereof;

Securing individual components (e.g. controllers, industrial PCs) is important, but does not solve the problem if one cannot assert the security of the overarching system architecture --- this architecture puts the first order security and dependability assertions at the level of information flow between LANs, e.g. corporate to SCADA, Internet to corporate, SCADA1 to SCADA2, etc.

2.1 Meeting the requirements

Whilst we do not claim this is the complete solution, by using such architecture the problem of protecting the power grid (or similar critical infrastructures) is in a first instance reduced to the problem of protecting LANs from the WAN or other LANs, making it easier and more effective to draft further protection measures, like for example, ultra-resilient in-LAN individual controllers for critical parts of the physical control system.

In the cyber reference architecture each LAN is connected to the WAN through a special interconnection and filtering device, let us call it CIS, a gateway implementing the CIS Protection Service, which ensures that both the incoming and outgoing traffic satisfy the security policy defined to protect the infrastructure.

A CIS is hence a kind of firewall, so what distinguishes our architecture from recent proposals for secure cyber architectures? In essence, the characteristics that fulfil the challenges enumerated earlier, that is: *dependability and security* against cyber-attacks and computer-generated faults, in an *automatic* and *unattended* way; *perpetual* operation or at most very short unavailability periods; *resilience* against unexpected or overstress situations.

Firstly, the CIS firewall works at the application layer, offering a richer semantics than for example TCP/IP level ones, and is distributed, as can be seen in Figure 1. This offers better intrusion prevention than alternative lower-level and centralised (perimeter-based) approaches.

Secondly, intrusion prevention, the workhorse of current approaches to security, even in common ICT, will not be enough to achieve our purpose. The CIS firewall is made *intrusion tolerant*, thanks to replication [6]. Replication is used in order to guarantee system correct operation in an automatic way, when some replica is compromised, since the spares will take over the operation. Since the attacker will attempt at repeating any successful replica intrusion in the other replicas, diversity is used across replicas (hardware, operating system architecture, binary layout, etc.) to oblige him to start anew.

Thirdly, whenever applicable, system defences are enhanced by guaranteeing *trusted-trustworthy* operation through architectural hybridization [8], an architectural paradigm whereby some special-purpose components exhibit very high resilience to attacks and faults, so that they can provide a small set of services improving the intrusion tolerance of the system. They are constructed in such a way that their resistance to faults and hackers can justifiably be trusted. This concept is in line with, but richer than, recent technological concepts like trusted computing or trusted platform modules.

Fourthly, even if intrusion tolerant, under stressing situations such as continued production of attacks or faults, the system will inevitably fall down, regardless of the number of replicas. Given a quorum *f* of allowed faults/intrusions, it is only a matter of time and effort from the attacker to exceed it. CIS resilience against this is achieved thanks to replica recovery for self-healing [7]. Rejuvenation is used periodically or on demand, to remove the effects of malicious attacks that may have compromised some replicas, and thus replace the quorum of fault/intrusion tolerance of the system.

Together, the automation provided by intrusion tolerance and the resilience provided by self-healing secure the objective of unattended operation in the presence of attacks and faults, so important in large-scale control systems with remote devices.

2.2 Architecture details

The infrastructure architecture is modelled as a WAN-of-LANs [5], as shown in Figure 1. All the Information and Communications Technology (ICT) parts necessary for the control of the whole power grid are logically grouped in substations and finally in local area networks (LANs). LANs are interconnected by a global interconnection network, called WAN. The WAN is a logical entity owned and operated by the critical information infrastructure operator companies, which may or not use parts of public network as physical support. All traffic originates from and goes to a LAN, so packets are switched by the WAN through the several CIS.

CIS collectively act as a set of servers providing distributed services aimed to control both the command and information flow among the ICT parts of the critical infrastructure, securing a set of necessary system-level properties. This set of servers must be intrusion-tolerant, prevent resource exhaustion providing perpetual operation, and be resilient against assumption coverage uncertainty, providing graceful degradation or survivability. An assumed number of CIS can be corrupted; in consequence, a logical CIS is implemented as a set of replicated physical units (CIS replicas) according to fault and intrusion tolerance needs. Likewise, CIS are interconnected with intrusion-tolerant protocols, in order to cooperate to implement the desired services. CIS is the substation gateway interfacing a protected LAN with the WAN, as shown in Figure 2.

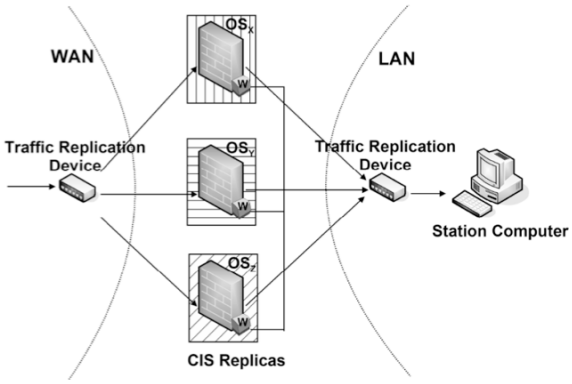

Figure 2. CIS intrusion tolerant hybrid architecture

In order to be intrusion-tolerant, the CIS is replicated (with diversity) in *n* machines and follows its specification as long as at most *f* of these machines are attacked and behave maliciously, both toward other replicas and toward the station computers in the protected LAN. Given the nature of attacks being malicious, an intrusion-tolerant firewall requires at least *3f+1* servers in order to tolerate *f* intrusions. Both the incoming and outgoing traffic is managed by "Traffic Replication Devices" that behave like Ethernet hubs: when they receive a packet from a port, they broadcast it to all the other ports; hubs are simple transceiver electronics which we assume are not

attackable. This way, the traffic received by the CIS from the WAN is spread to all the replicas, and the traffic generated by each replica is spread to all the other replicas and to the protected LAN.

For added intrusion tolerance, the CIS is implemented using a hybrid architecture [8], so it is composed by two parts: the payload and the wormhole. The payload is the main system where applications and protocols are executed. As shown in Figure 2, the wormhole (*W*) is a small secure sub-system in each replica, connected to the other local wormholes through a secure control channel, isolated from other networks, through which the CIS can exchange information despite attacks on the payload network and CIS parts.

The CIS resilience is enhanced by rejuvenating CIS replicas through recoveries. In order to guarantee system availability despite the unavailability of recovering replicas, the number of replicas in the system is set to $n>=2f+1+k$, where f is the maximum number of allowed faults/intrusions, and k is the maximum number of replicas allowed to recover in parallel without jeopardizing either the availability or the intrusion tolerance of the system. This way, the system is able to tolerate at most f compromised replicas and recover k replicas simultaneously. The reasoning behind this formula will be detailed in the next section.

3. Analysis and evaluation

Identifying applications security-related vulnerabilities and collecting real data to learn about the tools and strategies used by attackers to compromise target systems connected to the Internet is a necessary step in order to be able to build critical infrastructures and systems that are resilient to malicious threats. Furthermore, we also have been validating the new mechanisms introduced, in order to assess whether they meet the desired resilience objectives.

3.1 Approaches

We followed two complementary approaches: vulnerability discovery by attack injection, and quantitative assessment by modelling.

The first one concerns software vulnerabilities identification based on attack injection. We propose attack injection with monitoring capabilities as a method for detecting vulnerabilities. This methodology tries to detect software bugs as an attacker would, i.e., trial and error, by consecutively attacking its target. We have developed a methodology and a tool for injecting attacks in order to reveal residual vulnerabilities in the applications and software components developed for the architecture [10]. The tool is being used in particular to locate security vulnerabilities in network servers and software components of the cyber architecture and CIS. Attack injection does not depend on a database of known vulnerabilities, but it rather relies on a generic and exhaustive set of tests. This allows the discovery of known and unknown vulnerabilities in an automated fashion. Likewise, it also allows validating the assumptions used in the intrusion-tolerant protocols. Vulnerability removal can be performed both during the development and operational phases. A form of intrusion prevention, it reduces the power of the attacker [4], making the life of the adversary increasingly harder.

The second approach comprised a set of evaluation campaigns. An overall evaluation of the several architectural alternatives as discussed in the previous section was performed, in order to assess and justify the design choices made, showing the tradeoffs obtained in the reliability of the system, through a metrics of percentage of failed time. Concentrating on a CIS as providing a core protection service (firewall-like), we will show, in the next section, several interesting results. Firstly, that the baseline CIS without any intrusion tolerance (simplex firewall) exhibits a very modest level of reliability for a unattended operation. Secondly, that only the incremental application of the mechanisms we have proposed brings the percentage of failed time down to really comfortable levels given our objective, stated in the beginning: resilient, automatic, unattended and perpetual operation, in the presence of cyber-attacks and computer-generated faults.

In a more specialized campaign, followed in cooperation with other members of the project team, the analysis of the redundant architecture of the CIS (Figure 2) was performed, evaluating how effective is the trade-off between the two styles of recoveries followed by the CIS: proactive (every T instants) and reactive (on demand, e.g. upon detection of an intrusion), identifying the relevant parameters of the architecture and finding the best parameter setup. Several typical dependability and availability measures of interest were used in this campaign [9]. Proactive recoveries rejuvenate the replicas in predefined instants of time, without being based on any fault detection. This means that proactive recoveries treat all the faults, including also the latent and hidden ones, which cannot be treated in other way, but they recover also correct replicas, weakening the availability of the system. On the other side, reactive recoveries are triggered only on replicas detected or suspected of being faulty; replicas not detected or

suspected of being faulty are never recovered, even if they are actually faulty, potentially weakening the dependability of the system. In particular, we have shown that increasing the detection coverage of intrusions has conflicting effects on both dependability and availability measures, and that these effects depend also on the behaviour of invalid or omissive intrusions. We are currently refining these conclusions.

3.2. Evaluation results

This section presents the results of evaluating, through simulation, the different architectures for building a firewall that have been discussed in the previous sections. We used the Möbius [11] tool to build a model of the different firewall architectures and to simulate the models.

We used one metrics in our simulations: *percentage of failed time*. As the name implies, this metric shows the amount of time the firewall is failed, during a period of unattended mission. A firewall is failed when a certain number of its servers is failed and/or compromised. In the case of a simplex firewall, i.e., composed by a single server, then the firewall is failed when that server fails. If the firewall is intrusion-tolerant, i.e., composed by a set of n servers and capable of operating if no more than f ($< n$) of them become compromised, then the firewall is failed when more than f servers are compromised.

The simulations used the following parameters:

- Maximum execution time (*met*): defines the maximum mission time of the firewall. For simulation purposes, we needed to define an upper bound on the execution time of the firewall.

- Minimum inter-failure time (*mift*): defines the minimum time interval between successful attacks. In each successful attack, the adversary randomly compromises one (firewall) server.

- The value of *met* was 10.000 hours (about one year) and it was constant over all simulations. The value of *mift* was varied in order to simulate different adversarial power.

3.2.1 Simplex firewall. In the first experiment, we measured the percentage of failed time of a firewall composed of a single server. The results are presented in Figure 3. As expected, the percentage of failed time grows when the interval between successful attacks is smaller.

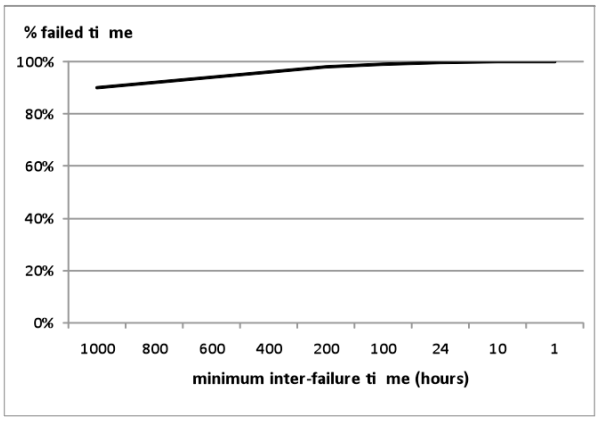

Figure 3. Simplex firewall

The main conclusion is that the percentage of failed time is substantially high even when the interval between successful attacks is moderate (1000 hours correspond to more than one month). This happens because the firewall is composed by a single server and so an adversary only needs to compromise one machine to bring the firewall down or corrupt its behavior.

3.2.2 Intrusion-tolerant firewall without hybridization. The goal of the second experiment was to show the advantages of adding intrusion tolerance capabilities to a firewall. As we have explained in Section 2.2, an intrusion-tolerant firewall without hybridization requires at least $3f+1$ servers in order to tolerate f intrusions. We simulated two different configurations of an intrusion-tolerant firewall without hybridization: one with 4 servers capable of tolerating one intrusion and another with 7 servers capable of tolerating two intrusions.

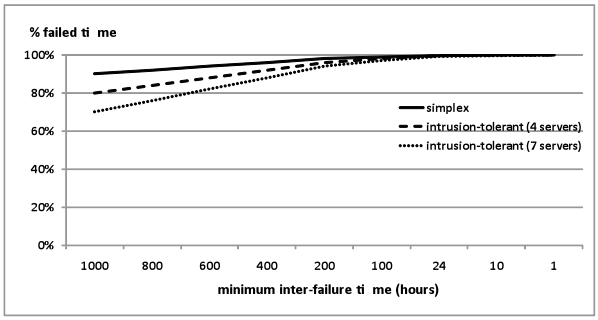

Figure 4. Intrusion-tolerant firewall without hybridization

Figure 4 shows that intrusion tolerance pays off when the minimum inter-failure time is above one day. This happens because an adversary needs to compromise more than one server in order to corrupt the firewall as a whole. This is not an easy task, given

that the servers are all different, do not share the same set of vulnerabilities, and so the adversary needs to restart his work after compromising one of the servers.

Although it is clear that an intrusion-tolerant firewall offers better protection than a simplex one, note that, for long unattended missions (10.000 hrs.), such an important device is failed more than 60% of the time even when the interval between successful attacks is over one month.

3.2.3 Intrusion-tolerant firewall with hybridization. In the third experiment, we measured the percentage of failed time of an intrusion-tolerant firewall that makes use of hybridization, i.e., each server has a set of trusted components. This type of firewall requires *2f+1* servers in order to tolerate *f* intrusions.

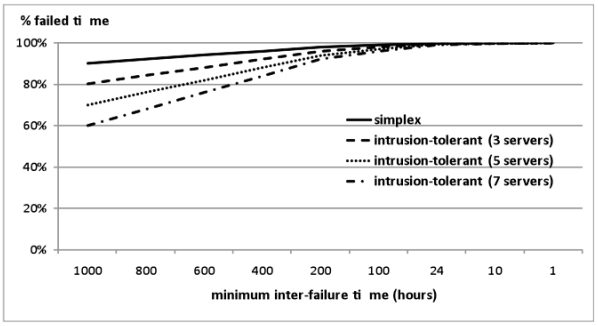

Figure 5. Intrusion-tolerant firewall with hybridization

Figure 5 shows again that intrusion tolerance pays off when the interval between successful attacks is above one day. The main advantage of using hybridization is that we save resources, i.e., servers, and we are in this way able to tolerate more intrusions with a fewer number of servers. For instance, 7 servers are now capable of tolerating three intrusions and this allows to reduce the percentage of failed time by 10% when the minimum inter-failure time is about one month.

Hybridization allows improving the protection of an intrusion-tolerant firewall, but we still have to solve the problem that the firewall is failed more than half of the time when the interval between successful attacks is over one month.

3.2.4 Self-healing & intrusion-tolerant firewall. In the fourth and last experiment we evaluated the impact of adding self-healing to an intrusion-tolerant firewall. With self-healing, the firewall servers run periodic rejuvenation procedures that not only remove the effects of intrusions, but also modify the servers in a way that an adversary cannot compromise a server through an attack that had success in the past. A self-healing and intrusion-tolerant firewall requires *2f+k+1* servers in order to tolerate *f* intrusions between rejuvenations and assuming that at most *k* servers rejuvenate at the same time. Typically, *k=1* and so *2f+2* servers are required to tolerate *f* intrusions between rejuvenations.

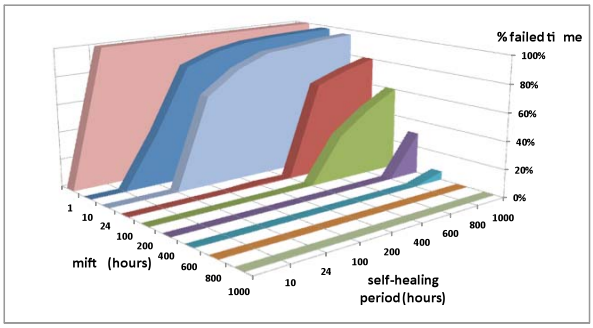

Figure 6. Self-healing firewall with 4 servers

Figure 6 depicts the percentage of failed time per self-healing period and minimum inter-failure time (*mift*) of a firewall composed of 4 servers and thus able to tolerate one intrusion between rejuvenations. We have chosen a 3D representation in order to clarify the presentation of the simulation results. A small self-healing period means that the firewall rejuvenates more often and therefore it is more difficult for an adversary to cause many intrusions. A small *mift* means that the adversary is able to perform successful attacks more quickly.

In one extreme, a firewall capable of rejuvenating once per hour (this is feasible given that in our lab we have a firewall rejuvenating each 10 minutes) is capable of resisting successful attacks with an interval of down to one hour. This means that an adversary that wants to break the firewall, will need to compromise two different servers (with a different set of vulnerabilities) in less than one hour. In the other extreme, even if rejuvenations are only done once per month (~800 hours), doing them largely reduces the percentage of failed time. For instance, when *mift*=400 hours, one rejuvenation per month guarantees 0% of failed time, while an intrusion-tolerant firewall with the same number of servers and no self-healing is failed 80% of the time (see Figure 5).

These results show that a self-healing firewall with 4 servers offers a good protection level. This protection level can still be increased if we add two more servers to our deployment. With a total of 6 servers, the self-healing firewall is capable of withstanding two intrusions between rejuvenations, this way doubling its protection power. Figure 7 shows exactly this given that for each combination of *mift* value and self-healing period, the firewall exhibits a smaller percentage of failed time.

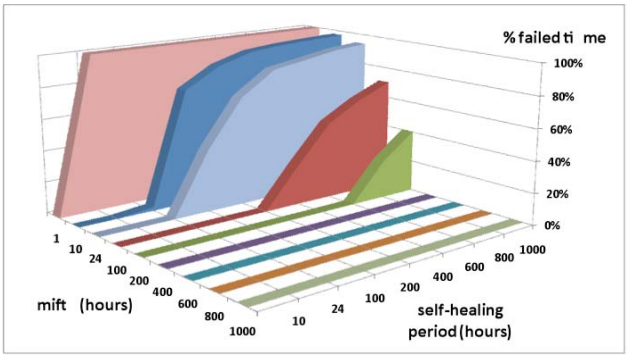

Figure 7. Self-healing firewall with 6 servers

Note also that we have assumed in our experiments that the maximum execution time of the firewall is 10.000 hours, while an actual unattended firewall may have a substantially higher execution time. Note that higher execution times would not affect the performance of the self-healing firewall (with 4 or 6 servers), but would increase the percentage of failed time of the simplex and intrusion-tolerant firewall without self-healing.

To summarize, Figure 8 compares the percentage of failed time of the different types of firewalls evaluated in this section. The advantages of self-healing are clear even with a rejuvenation period of one day.

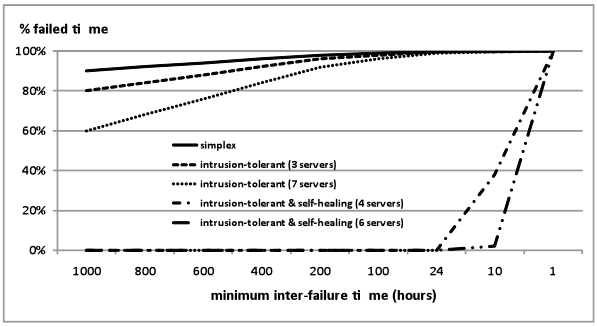

Figure 8 - Comparison of the different types of firewalls

4. Conclusions

An innovative cyber architecture for process control was recently proposed, in the context of a project related to the power grid. The design of the architecture is as generic as possible, with a view of possibly serving as a reference cyber architecture for process control infrastructures.

The need for a new architecture is explained by the fact that cyber architectures for process control have extremely demanding requirements, whose importance we have evaluated in this paper: *dependability and security* against cyber-attacks and computer-generated faults, in an *automatic* and *unattended* way; *perpetual* operation or at most very short unavailability periods; *resilience* against unexpected or overstress situations.

We reported some of the lessons learned in the development, analysis and evaluation of the proposed cyber architecture for process control, which look very promising in terms of usability of the concepts in real-life systems. Namely, through several simulations, we have shown the incremental power of the several mechanisms used to enhance the operation of the CIS, the core component of the architecture, modelled as a firewall.

At a time where it is believed by many stakeholders that it suffices to provide cyber architectures for process control infrastructures with well designed firewalls, in order to protect them against cyber-attacks, we hope to have shown that, for the expected level of threat, classical (simplex) firewalls in unattended operation mode exhibit modest resilience.

References

[1] Madani, V., Novosel, D.: Getting a grip on the grid. Spectrum, IEEE 42 (2005) 42–47.

[2] Dawson, R., Boyd, C., Dawson, E., Gonzalez Nieto, J.: SKMA: a key management architecture for SCADA systems. In: ACSW Frontiers '06: Proceedings of the 2006 Australasian workshops on Grid computing and e-research, Darlinghurst, Australia, Australian Computer Society, Inc. (2006) 183–192.

[3] Wilson, C.: Terrorist capabilities for cyber-attack. In: Dunn and V. Mauer, editors, Int. CIIP Handbook volume II, CSS, ETH Zurich (2006) 69–88.

[4] Verissimo, P., Neves, N. F., Correia, M.: Intrusion-Tolerant Architectures: Concepts and Design. In: R. Lemos, C. Gacek, & A. Romanovsky, editors, Architecting Dependable Systems, vol. 2677 of Lecture Notes in Computer Science, pp.3-36, 2003.

[5] Verissimo, P., Neves, N. F., Correia, M.: The CRUTIAL reference critical information infrastructure architecture: a blueprint. In: International Journal of System of Systems Engineering, vol. 1, n. 1/2, pp 78-95, 2008.

[6] Bessani, A. N., Sousa, P., Correia, M., Neves, N. F., Verissimo, P.: Intrusion-Tolerant Protection for Critical Infrastructures. Technical Report DI/FCUL TR-07-8, 2007.

[7] Sousa, P., Bessani, A. N., Correia, M., Neves, N. F., Verissimo, P.: Resilient Intrusion Tolerance through Proactive and Reactive Recovery. In: Proceedings of the 13th IEEE Pacific Rim Dependable Computing Conference, pages 373—380, 2007.

[8] Verissimo, P.: Travelling through wormholes: a new look at distributed systems models. SIGACT News 37 (2006) 66–81.

[9] Daidone, A., Chiaradonna, S., Bondavalli, A., Verissimo, P.: Analysis of a Redundant Architecture for Critical Infrastructure Protection. In: R. Lemos, F. Di Giandomenico, C. Gacek, H. Muccini & M. Vieira, editors, Architecting Dependable Systems, vol. 5135 of Lecture Notes in Computer Science, pp.78-100, 2008.

[10] Neves, N. F., Antunes, J., Correia, M., Verissimo, P., Neves, R.: Using Attack Injection to Discover New Vulnerabilities. In: Proceedings of the International Conference on Dependable Systems and Networks (DSN), pages 457-466, 2006.

[11] Deavours, D. D., Clark, G., Courtney, T., Daly, D., Derisavi, S., Doyle, J. M., Sanders, W. H., Webster, P. G.: The Mobius framework and its implementation. In: IEEE Transactions on Software Engineering, 28(10):956–969, 2002.