# An Analysis of Two New Directions in Control System Perimeter Security

Ludovic Piètre-Cambacédès[1,2], Pascal Sitbon[1]
[1]Electricité de France, [2]TELECOM ParisTech
France
ludovic.pietre-cambacedes@edf.fr, pascal.sitbon@edf.fr

**Abstract**: Traditional IT firewalls are used in control systems to provide a security perimeter and offer important protection. However, two new directions in control system perimeter security are now available and claim to provide a higher level of security: namely, one-way communication technologies and control system protocol deep packet inspection. In this paper, we will present these technologies and purported benefits, analyze their effectiveness against a variety of attacks, and consider architecture options where these new classes of security products may offer the most benefits. Finally, the paper discusses how this new technology can be extended and further tailored for the control system security challenge.

**Keywords:** Diode, One-way Communications, Deep Packet Inspection, Firewall, Intrusion Prevention Systems (IPS)

## 1 Introduction

The last years have seen tremendous evolutions in the field of Industrial Control Systems (ICS); from closed, isolated, and proprietary architectures, they are now becoming more and more interconnected, relying in large part on numerous standardized technologies that enable new functionalities and cost-effectiveness. Those evolutions have also introduced new vulnerabilities, which combined with a highly-evolving threat landscape have turned ICS cybersecurity into a top priority and a permanent challenge. Fortunately, traditional cyber security solutions, used in the corporate context, have found their way into industrial environments. Firewalls, DMZ's, and appropriate organizational structures are now commonly put into place. Nevertheless, the specificities of ICS call for complementary or adapted security approaches in certain situations. This article presents and discusses two of them. The first approach, presented in Section 2, deals with one-way communication restrictions in general and physical data diodes in particular. Commercial solutions and reliable technologies are already available. The second approach, discussed in Section 3, considers deep packet inspection for industrial protocols and process-logic awareness for security decisions. This approach is still experimental but is promising. Of course, none of these approaches can ensure an appropriate cybersecurity posture on their own and have to be considered in a global and consistent approach.

# 2 One-way Communications

Considering the high sensitivity, in terms of integrity and availability, of specific critical industrial control systems, their connection to other systems and networks are often deemed acceptable only if "one-way communication" is ensured. Unfortunately, this requirement is not precise enough and, in fact, covers a large scope of implementations, from regular firewalls to physical data diodes, enforcing different levels of security. This section exposes the underlying stakes and implementation possibilities before providing some targeted insights and considerations for diode solutions.

## 2.1 When One-way Is Not Always One-way…

Communication flows between two domains with unequal security levels or requirements have to be restricted. Several kinds of security controls are usually enforced dealing with the nature, the direction, and the innocuousness of the exchanged data; we focus here only on communication direction restrictions. On this topic, different graded enforcements are possible:

a. A first approach is based on the initiation of the communications, which can only be started on the initiative of one domain towards the second domain.

b. A complementary measure is often added to the initiation restriction, controlling the direction of the transmitted content. In addition to the fact that the communication is initiated by a given domain, the data, in terms of content and useful payload, flows only from one domain to the other one.

c. A last approach strengthens the restriction to a maximum, allowing only a strict one-way communication from one domain to the other one, preventing even a single bit, including signaling, from going in the other direction.

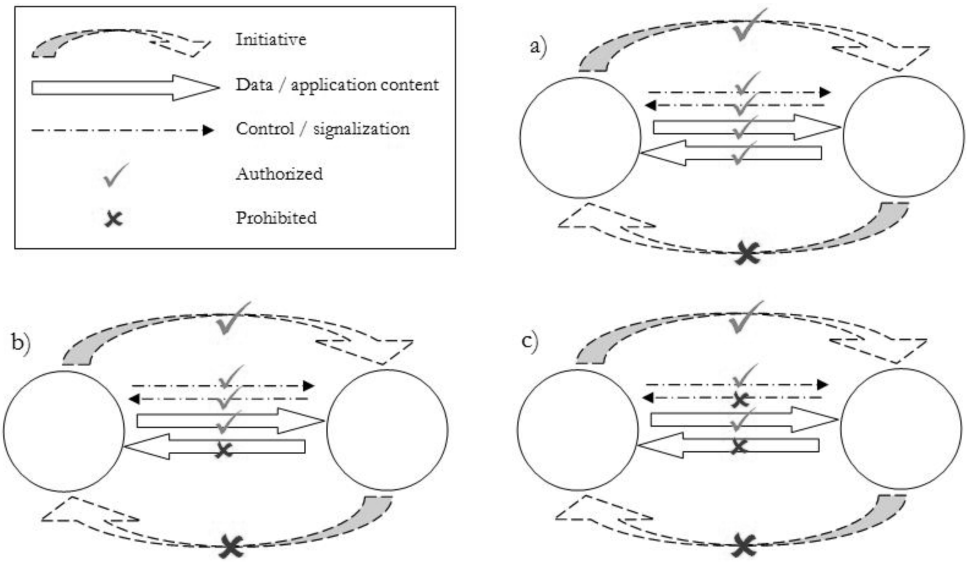


*Figure 1 – Different Communication Flow Restrictions*

Despite their differences, it is unfortunately rather common to find them labeled under the same designation of one-way communication. They may even be misleadingly represented by the same symbol: a diode. However, the nature and strength of protection provided by each of them differ to a great extent. This confusion should be avoided at all costs, as it could lead to a dangerous illusion of security. We propose to name them, respectively:

a. initiative-restricted policy

b. content one-way policy

c. strict one-way policy

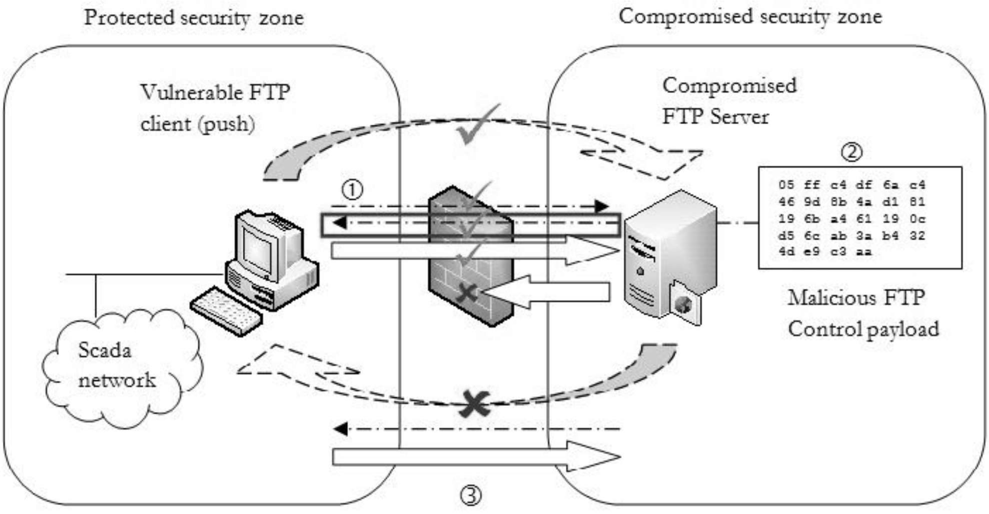
Only the first, initiative-restricted policy can be enforced in a straightforward fashion with a regular firewall. This approach is typically adopted between untrusted domains, such as the Internet and corporate systems.

Content one-way communications can be adopted to protect specific domains within an organization-wide global network. The segmentation between business networks and SCADA/industrial control systems would be a particular declination. At this point, a first warning should be made: logical one-way policy may be tricky to implement, as application layer intelligence is needed. In fact, the distinction between control information, such as acknowledgements or error signaling, and application data has to be clearly identified by the filtering device. This requires specific capabilities, commonly available for classical internet protocols, like FTP, but not common for industrial protocols.

The difference in the level of protection between a) and b) is relatively easy to capture. If the first case prevents a direct connection attempt by an attacker towards the protected zone, the attacker can still wait to piggyback on legitimate communication to get across the barrier. The content one-way policy makes the attacker's life much harder, as the policy prevents him from directly introducing malicious content though regular mechanisms.

The difference between b) and c) may be less obvious, and the content one-way policy is sometimes considered to be a maximum security policy. Even if it can indeed provide a high level of protection, network attacks are in fact still possible, as control and signaling data remains authorized to enter the protected zone, and to be inserted into it. This is the case for TCP connections in particular, TCP being a non one-way protocol by nature.

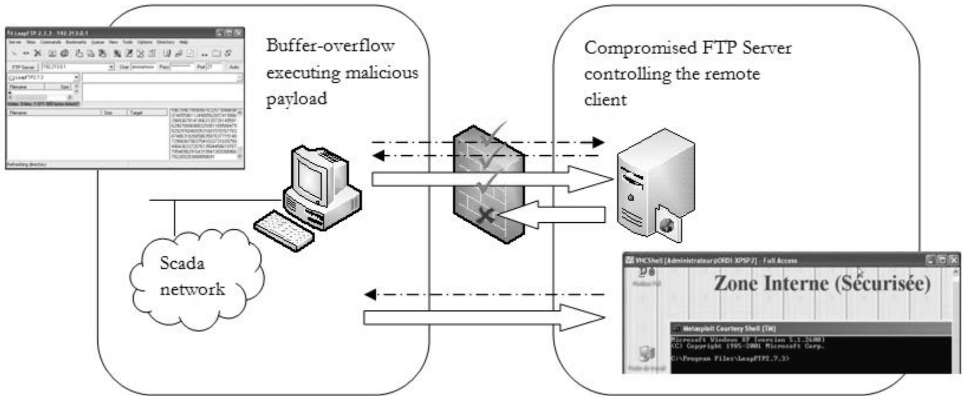
While this difference may be clear to the reader, a simple example may be useful to help the understanding of a non-expert audience. Ms. Smith, recently appointed CSO of the ACME Power Utility, has declared to her board of directors, "our SCADA system is 100% secure, as only one-way communications are permitted with the rest of the company." In fact, the diode in question is a classical firewall controlling a content one-way policy approach for outgoing FTP connections, used to export SCADA historian data to the business zone. A simple proof of concept may be useful to break such a myth. Let us consider the scenario represented in Figure 2.


Protected security zone
Compromised security zone
Vulnerable FTP client (push)
Compromised FTP Server
①
②
Scada network
Malicious FTP Control payload
③

```
05 ff c4 df 6a c4
46 9d 8b 4a d1 81
19 6b a4 61 19 0c
d5 6c ab 3a b4 32
4d e9 c3 aa
```

*Figure 2 – A Reverse FTP Attack with a "One-way" Configured Firewall*

The firewall is configured with a content one-way policy for FTP. Only FTP "push" from the client to the server outside the protected zone is allowed. Passive mode is configured so that the data and control FTP connections flow in the same direction. Inward-bound TCP connections are not permitted. Nevertheless, FTP signaling is allowed in both directions between the client and the server in order to make this "push" possible. FTP control data can enter the zone, and will be interpreted by the client (cf. ① in Fig. 2). This could be enough to gain access to the client. In step ② from Fig. 2, the malicious server crafts the malicious ftp control messages as a reply to the client to exploit a buffer-overflow vulnerability in the remote client. The firewall policy is respected, but the client is compromised. Depending on the firewall and its inspection mechanisms, complete, malicious connectivity can potentially be established between the two zones as shown in Figure 3. Of course, all of this requires the attacker to have knowledge of an exploitable vulnerability in the client software. This said, even if server vulnerabilities are numerous and historically the most commonly exploited, clients are also vulnerable. In fact, a quick search on classical security archives leads to tens of client-side vulnerabilities and associated exploits.


Buffer-overflow executing malicious payload
Compromised FTP Server controlling the remote client
Scada network
VNCShell [Administrateur@ORDI-XPSP2] - Full Access
Zone Interne (Sécurisée)
Metasploit Courtesy Shell (TM)

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Program Files\LeapFTP2.7.3>
```

*Figure 3 – Exploiting the Control Channel of a FTP Connection*

*Stricto sensu*, only strict one-way policy implementations should be called diodes and can provide the implied level of protection. All TCP-related connections or, more broadly, all protocols implying bi-directional signaling, including acknowledgements, are ineligible for a strict one-way designation. A firewall enforcing a strict one-way policy and acting only on non-connection-oriented/unidirectional protocols such as UDP can be considered to be a data diode. However, firewalls may be vulnerable or misconfigured; in critical environments, the one-way functionality may have to be ensured physically, by dedicated hardware called physical data diodes.

## 2.2 General and Historical Considerations on Physical Data Diodes

Physical data diodes are not new and, although they are currently gaining interest in the SCADA security community for their integrity protection capabilities, the military world has used them for a long time, especially for classified environments. To the best of our knowledge, the first public discussion of this usage appeared in 1988 [4]. Since then, the best source of information comes from declassified technical reports or non-classified work from military organizations. The Austrian DSTO (Defense Science and Technology Organization) have in particular released three reports of prime interest in this area: a generic report called "Data Diodes" in 1995 [5] a more focused report investigating the use of UDP for such devices in 1998 [6], and a last report from 1999 describing a simple optical data diode implementation based on UDP [7]. The NRL (Naval Research Laboratory) from the US Department of Defense has also contributed to the public discussion, mainly through development and research around their concept of a "Pump". Different from a pure diode, especially regarding its acknowledgement and reliability capabilities, it was initially presented in 1993 [8] and discussed through different papers and studies throughout the years [9-13]. Its development has recently been summarized by their authors in "The Pump, a decade of covert fun" [14]. The French DGA (Delegation Générale pour l'Armement) has also communicated on physical data diode developments in 2006 [15], releasing an open source implementation of an adapted file transfer protocol in addition.

Grounded in this history, physical data diodes have first been used mainly for confidentiality protection, and are used for integrity protection in a sort of reversed way. In the first case, traditionally found in military and governmental environments, the diode ensures that data can only be sent from a network with a higher classification level to a lower one, whereas in the second case, more typical of control systems, the diode prevents data from being sent from a lower security zone to a higher security zone. In fact, physical data diodes can play a role in protecting each of the C.I.A aspects of information security:

- Confidentiality: preventing data from being sent from a system with a higher classification level to a system with a lower classification level.
- Integrity: preventing data from being sent from a less protected system to a more protected system.
- Availability: preventing flooding and other denial of service events, malicious or not.

SCADA and ICS cyber security challenges have lead to a resurgence of interest for this "old" technology. The next pages will delve deeper into its implementation alternatives before providing some critical and architectural considerations.

## 2.3 Technical Approaches

### 2.3.1 RS232

Customized RS232 serial links have been used over the years to implement physical diodes in a simple fashion, despite their limited bandwidth. Report [16] presents details of several alternatives that can be used to realize such data diodes, explaining cabling and necessary modifications.

### 2.3.2 AUI Ethernet

Another approach to implementing strict one-way on a physical basis is to modify a point-to-point Ethernet link at the physical layer. The deprecated AUI standard provides a very simple way to realize such a diode. Figure 4 indicates the modification preventing write operations on the low side of the cable, where the transceiver is represented.

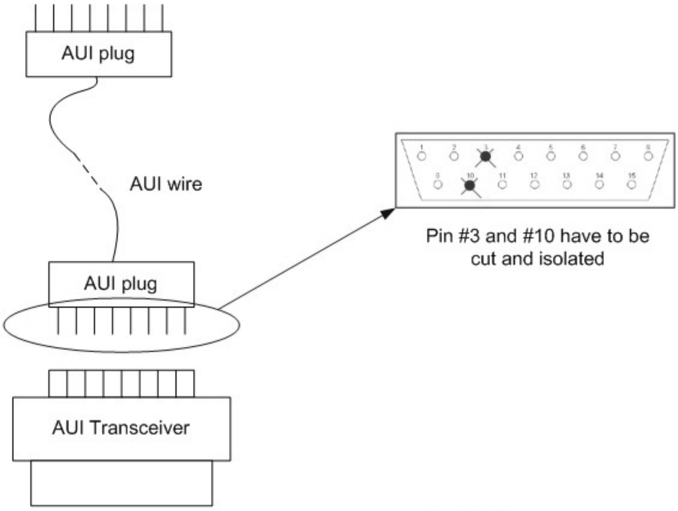


*Figure 4 – AUI "No-write" Modification*

### 2.3.3 Optical Fiber

The most common design relies on Ethernet optical fiber-based standard connectivity. A simple design involves two dedicated transceivers connected as shown in Figure 5.

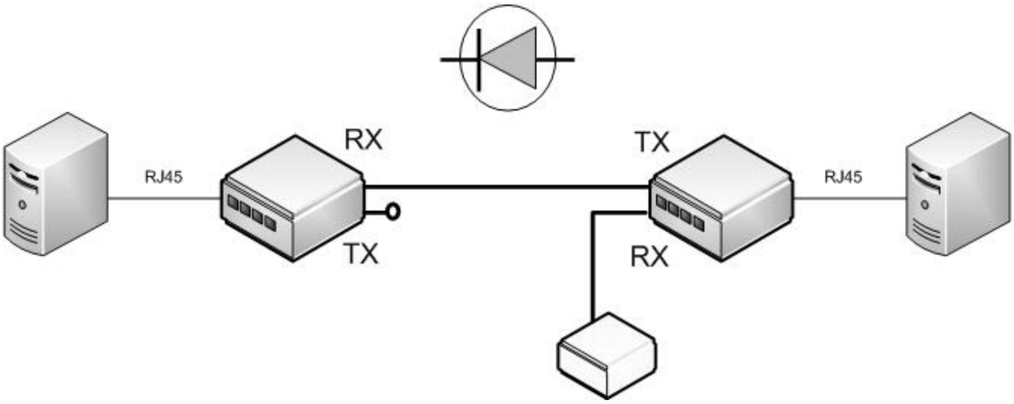


*Figure 5 – A Classical Optical Diode Design*

An equivalent design can be made with two optical network cards embedded directly in the servers. In both cases, when standard components are used, the receive links (RX) need to be connected, as standard card or transceiver firmware expects to receive activity to consider the link as alive. This is not a problem on the emitting side but may require artificial connectivity on the receiving side [7,15]. The transceiver represented in the lowest part of Figure 5 ensures this function.

The possibility of using standard equipment, the bandwidth capabilities, the natural fail-secure behavior, and the TEMPEST attack resistance (at least on the optical link) are some of the reasons why optical fibers appear as the best basis for implementing physical data diodes.

## 2.4 Solutions and Functionalities

Numerous providers propose commercial solutions based on optical fiber point-to-point links, such as the link described in Section 2.3.3. Table 1 gives a non-exhaustive overview of some commercially-available solutions. Most of them implement file transfer capabilities, email transfer, and raw data forwarding.

| Provider (country) | Core Product | Characteristics & Functions | Certification | Remarks |
|---|---|---|---|---|
| Owl (USA) | Data Diode Network Interface Cards (DDNIC)<br>[17,18] | 55, 155 MB/s or 2.5 Gb/s (ATM) (5, 15 or 270 Mb/s user throughput)<br>File Transfer, raw UDP forwarding, multiplexing, TCP handling (proxy). | Common Criteria EAL4 [19]<br>"Approved-to-Operate" in the US DoD and Intelligence community [20] | Based on modified ATM, not Ethernet (from Sandia Labs [21,22]). Differs from Fig 3.<br>Claimed as widely deployed (>650) in US agencies [18,20]. |
| Tenix (Australia) | Interactive Link Data Diode (IL-DD<br>[23] | 1Gb/s or 100Mb/s<br>File Transfer, raw UDP forwarding, SMTP, keyboard switches | ITSEC E6 [24]<br>Common Criteria EAL7+ [25,26] | From the DSTO Starlight products (> 5000 units), widely deployed in Australian gov. [27] |
| Thales (France) | ELIPS-SD<br>[28] | 100 Mb/s (20 Mb/s user throughput)<br>File Transfer, raw UDP forwarding, Email (SMTP) | Secret-Défense approval for use in French Defense/State environments. | Claimed as widely used in French military environments. |

| Provider (country) | Core Product | Characteristics & Functions | Certification | Remarks |
|---|---|---|---|---|
| Waterfall (Israel) | Waterfall One-Way<br>[29,30] | 100 Mb/s<br>File Transfer, raw UDP forwarding Video, TCP handling, Scada protocols (Modbus, PI, OPC, Profibus) | N.A. | Proposes a SCADA targeted offer [31] |
| Fox-IT (Holland) | Fort Fox Hardware Data Diode (FFHDD)<br>[30,31] | 100BaseT (servers) and 1 Gb/s (diode) (40 Mb/s user throughput)<br>File Transfer, SMTP, Raw UDP forwarding, CIFS (SMB), RS232. | State-Secret Classification in. Holland<br>Accredited up to NATO Secret level [31]<br>AMSG 720B | Under Common Criteria EAL4+ evaluation. |

*Table 1 – Physical Data Diode Commercial Solutions*

As stated before, the table above does not pretend to provide an exhaustive analysis of the available offerings; other solutions may exist and the reader is invited to consult the indicated references and associated websites to obtain deeper information on the presented ones. In particular, the certification column may deserve some complementary considerations.

In the physical data diodes landscape, to the best of our knowledge, only two solutions have been evaluated along the Common Criteria: the Owl and the Tenix solutions. Some others claim to be in-process (Fox-IT), others may choose not to attempt the certification. Generally speaking, to appreciate the added value of such certifications and possibly compare them, one should not forget to carefully analyze the Target of Evaluation (ToE) considered, and the associated Security Target (ST), defining the security claims and assurances of the evaluated products. For the record, numerous Microsoft operating systems have been certified. For example, Windows 2000 is EAL4+ [32]. This Windows 2000 ST was assuming non-malicious environment. Another example lies in the "certification" of a firewall where the ToE was only dealing with the administration module, and not the filtering capabilities.

Owl and Tenix products have been approved EAL4 and EAL7+, respectively. It is interesting to notice that the Owl ToE includes the physical and the driver parts of its optical network cards whereas the Tenix ToE is only concerned with the physical hardware part. These differences are explained by the way the strict one-way policy is enforced by each product. The Security Targets also have differences regarding the considered environment and threats. These aspects can explain and lessen the EAL difference. Finally, evaluation assurances levels above EAL4 do not benefit from the same global international agreement as lower levels [33]: an EAL7+certificate delivered in the United States is not recognized as such in France while an EAL4 certificate would be. The important message here is that one solution should not be considered better or worse than another only because of a simple comparison of their EAL. Common Criteria evaluations do provide a lot of valuable information, but require thorough analysis. Furthermore, the absence of Common Criteria certification does not imply that a given product does not compare to the certified ones.

## 2.5 The Reliability Question

A major technical question regarding strict one-way policy can be raised regarding transmission reliability. As no signaling, acknowledgement, and other connection-oriented mechanisms are allowed, how can all of the mentioned functionalities be ensured with a sufficient quality of service?

First, all of those mechanisms are usually put in place to compensate for harsh transmission conditions: shared media, multi-hopes routes, perturbations and so on. In fact, it has been proved that a non-connection-oriented protocol like UDP can be quite reliable in small and dedicated networks [4]. Physical data diode solutions use one-way protocols in ideal conditions: dedicated point-to-point link, no collisions, no or limited interferences. Nevertheless, aspects other than just the transmission channel have to be considered. For example, the receiving and the transmitting processes have to be managed in a very specific manner considering the impossibility of feedback information from the receiver to the emitter. There is no obvious solution for the emitting process to be sure that the receiving side is ready to handle the transmitted data, which can then be dropped. Buffers may be overloaded, or the CPU could be used for other tasks and the receiving side may simply not run fast enough to treat the flow. Several measures can be implemented to help, including the following:

- On the receiving side, network buffer sizes and treatment process priorities are specifically dimensioned to cope with peak emissions from the emitting side, which benefits from lesser capabilities.
- Sending delays between packets can be configured.

As losses can still happen, complementary mechanisms are put in place to compensate:

- Data is divided, numbered, and sent several times.
- The repetitions may involve a different order of transmission.
- Hash, CRC, and equivalent mechanisms are added at different levels.
- Keep-alive and status messages are sent to the receiver.

Specific headers or footers are typically used to pass such information from the emitter to the receiver in order for it to undertake the appropriate actions. If the emitter is blind, the receiver can still use these mechanisms to obtain a clear view of the status of the communication (complete, in process, erroneous, etc.). Overall, an excellent level of reliability can be achieved.

### 2.6 Architectural Discussion

As discussed before, a physical data diode can be used to protect highly sensitive networks from lower security zones. This implies that such a division is clearly established in the considered environment. Guidance such as ISA99 [34], but also CIGRE's works [35,36] for power utilities or the future IAEA computer security guide for nuclear facilities [37] can help in this task, as they all discuss zone distinction and graded approaches to security. Note that such diodes can protect a complete zone but can also be considered in specific cases for single systems; in any case, the diode should of course be a choke point. Functionally speaking, Table 1 gives some indications of possible classical applications: file transfer capabilities can be leveraged in many situations, from historian data to maintenance data exports; UDP raw transfer is also quite generic, used in applications ranging from syslogs and SNMP traps to video-surveillance applications. Many others are possible, and considering the diversity of industrial architectures and systems, there is no single perfect solution and recommendation. However, three generic "principles" may be useful for the security architect:

- The "security Ockham's razor": if two situations provide the same level of security, choose the simplest. In our case, when considering strict one-way communications between two entities, the designer should always check if the two communicating entities could and should not be put into the same security zone. The simpler the better.
- The "One-way vs. zero-way debate": is one-way connectivity absolutely needed for highly sensitive systems? Would no communications at all ("Air Gap") and out-of- band solutions (with human operators) be acceptable?
- The "Maginot line syndrome": physical diodes provide such a high level of assurance regarding network connectivity that one could be tempted to give them magical powers… and forget that other communications exist (e.g., removable memory storage)! More generally, putting in place a diode should not prevent enforcement of other complementary and defense in depth security measures.

## 3 Deep Packet Inspection

In this chapter, we consider the value of deep packet inspection as an opportunity to enhance the defense of industrial control systems (ICS). There have been numerous studies on this theme, some of which are discussed here, and we also present a new direction that could bring ICS security a step further.

## 3.1 State of the Art

### 3.1.1 Protection and OSI Layers

The history of protection of network communications follows the OSI layers, see Tables 2 and 3. It started with filtering routers, using Access Control Lists (ACL) at the Network layer. The filtering is made on a packet basis, which means that the decision is taken regardless of what has been received before. The classical pitfall of such a technique is that a packet with the flag "established" coming from the external interface is accepted by the router, no matter if it is related to a real established session.

Then stateful firewalls arrived. They permit keeping track of the state of network connections to ensure that packets related to existing connections are authorized.

Finally, deep packet inspection and application proxies have completed the toolbox to cover the OSI stack, and are now commonplace in corporate IT. Application proxies like HTTP or FTP can be used to filter commands and data.

| Layer | Device | Example |
|---|---|---|
| 3 | Router with ACL | permit tcp any 80<br>permit tcp any 80 established |
| 4 | Stateful firewall | iptables –A OUTPUT –p tcp –dport 80 –j ACCEPT<br>iptables –A INPUT –m state – state ESTABLISHED –j ACCEPT |
| 7 | Application firewall | Permit specific Modbus functions, like "read register", eventually only on a specific range, for specifics values |

*Table 2 – Protection and OSI Layers*

For the industrial context, in 2003-2004 Cisco CIAG developed a firewall prototype [38] for Modbus/TCP, one of the most commonly used industrial protocols. This firewall has been released as open-source [39] as a module for Netfilter. Presently, Byres Security Inc. and MTL Instruments propose a commercial product called "Tofino Modbus TCP Enforcer Loadable Security Module" [42], which performs detailed analysis and filtering of all Modbus TCP messages, and is certified by Modbus-IDA. In November 2008, Checkpoint has announced the support of content inspection for Modbus/TCP on their UTM-1 Edge Appliances [51].

| Layer | Benefit | Limits |
|---|---|---|
| 3 | Restrict the allowed ports | Can be fooled by forged packet with ACK or RST flag |
| 4 | Maintain a table of allowed connections | No protection at the application level |
| 7 | Check the compliance to protocol definition (for example compute the length field and compare to the packet size)<br>Can restrict traffic based on layer 7 fields (function ID, data range) | Static rules, no knowledge of the industrial process logic |

*Table 3 – Benefit and Limits of Existing Security Devices*

Router ACL's, for example, only perform a basic verification of session initiation based on the presence of the TCP flags ACK and RST. In particular, there is no verification if the packet belongs to an existing TCP session, or if there is IP address consistency [44].

Layer 7 inspection, already implemented by available commercial products, is the state-of-the-art of ICS protection. It gives the possibility to restrict the traffic based on layer 7 characteristics such as function types (read, write, diagnostic), memory, or data range. Some protocol compliance checks are also implemented in different commercial products, taking advantage of the available SCADA Intrusion Detection Systems (IDS) signatures [52], by some general vendors like Tipping Point or more specific vendors like Industrial Defender.

### 3.1.2 Specifics of ICS

An industrial process is well known to the asset owner and has rather explicit bounds. In addition to the fact that the topology is static, the traffic is regular for a given state of operation, and is made of simple protocols - ICS are usually used and tailored for specific purposes. These purposes are highly dependent and are shaped by the overall facility mission, electricity production for example.

It is, in fact, possible to describe the normal technical conditions associated with this industrial process, thanks to the limited number of possible commands, the restricted nature of the data, and the determinist profile of communication, which gives pertinence to methods such as state graph modeling, formal verification, or model checking [43][49]. [50] introduces already-available commercial products called Network Based Anomaly (NBA) solutions that can observe traffic and compare it to a normal baseline on layer 3 and layer 4 characteristics, including IP addresses, ports, number of packets, total size of flow, and TCP flags. These solutions could be smoothly integrated into a control system network but should be very carefully implemented, especially by avoiding bridging together different security zones. We also have to take into account real-life constraints, for example the fact that there are few operators with advanced computer skills on industrial sites. The solutions must also be compatible with legacy equipment and respect the consideration of industrial priorities: safety first, then performance and security. In particular, we must pay attention to possible antagonisms between "fail-secure" and "fail-safe" situations.

### 3.1.3 Deep Packet Inspection for ICS

Deep packet inspection can be done by active or passive devices. Industrial contexts require us to be very careful when introducing such security measures: the top priorities are different from traditional business environments. Safety and process continuity are paramount. When introducing active devices, we must be sure that only 100% reliable preventive measures are put in place when safety and process continuity are at stake; passive detective measures are often preferred in these contexts.

Active devices are typically firewalls and Intrusion Prevention Systems (IPS); passive devices are Intrusion Detection Systems (IDS) in stealth mode. The technological progress made for one type of device is usually applicable to the others.

In both cases, two different strategies are possible. The most secure is to define a "white list" of what is explicitly authorized, assuming that process communications are globally predictable. All activity outside this "white list" is considered to be dangerous. The other way is to define what is explicitly known as harmful as a "black list" of patterns to reject. To establish such a black list, two approaches are used. The first one is based on the hypothesis that an intrusion requires an abnormal usage of a system. Thus, the problem is resolved by the detection of behavior deviation from a known baseline. The other one is based on direct detection of a harmful behavior [47].

The main underlying technologies are based on:

- Statistical models
- Expert systems
- Neuronal networks
- Pattern matching

Papers like [43] introduce the use of model-based IDS for SCADA Networks. The models are constructed in order to describe the expected behavior of the system.

## 3.2 One Step Beyond: "Layer 8"

Similar to studies performed in other papers [41][43], we propose to go a step further by defining an expected behavior of ICS exchanges on the network.

### 3.2.1 Architecture Issues

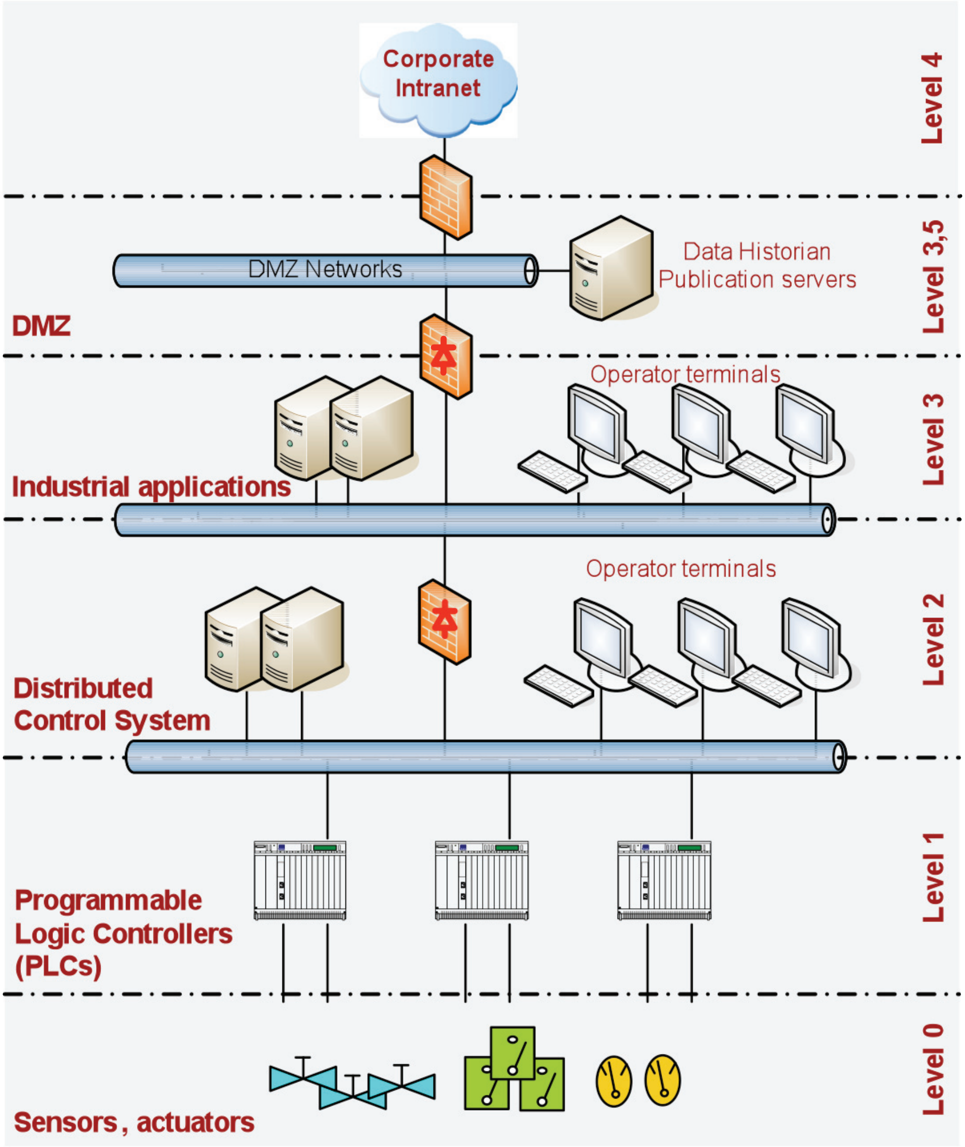


*Figure 6 – Classical ICS Architecture*

The natural location for deep packet inspection devices is at zone boundaries. Many ICS focus the initial security effort at the boundary between the process and business zones. This may not be the most interesting place for deep packet inspection as business communications are usually not authorized to enter ICS networks, and this communication rarely involves control system protocols. The number of zones, including for remote maintenance purposes, is exploding; this introduces several pertinent positions for deep packet inspection components. In addition, it may also be interesting to use security tools inside the process network itself, for example between two independent industrial sub-processes, such as environmental sensors and vibration monitoring, or between a supervisory network and control or safety networks.

Looking at a classical ICS architecture (Figure 6), the succession of security devices is not itself sufficient to obtain better security. If the protocols, operating systems and software versions used are the same, a vulnerability present between levels 3 and 3.5 would also be present between levels 3.5 and 4. As a result of the "defense-in-depth" principle, it should be required to use different software, operating systems, and protocols when moving through one level to another. It should also be required to use a rebound on a proxy and prohibit direct flows between two zones of different levels of trust. In Figure 6, level 4 is IT (corporate or business network) and levels 0 to 3 are ICS. The level "3.5" is shared between IT and ICS in order to abide with both security policies.

### 3.2.2 "Layer 8": Application Logic and Dynamic Layer

We propose to work now on layer 7 and "layer 8". Layer 7 is composed by application services but is not sufficient if we want to filter along the process itself. We have called "layer 8" the application logic, including data and industrial process logic, and content and application security.

| Layer | Device | Example |
|---|---|---|
| 8 | Proof of concept (real device TBD) | Permit Modbus / write register R if related values and states are coherent |

*Table 4 – The Layer "8"*

Layer 8 security is mainly based on industrial process awareness. Security decisions are made thanks to criteria like the operation mode of the installation (e.g. production, maintenance, idle, emergency stop), the values of other ICS variables (e.g. PLC registers), device states and measures (e.g. alerts, sensor network), and the date and repetition of actions. Reference [7] explicitly refers to the operation mode as an important input for determining the security posture of a facility.

In order to provide a global security coherence, [43] proposes a solution to detect Man-in-the-Middle (MITM) attacks based on correlation of traffic, see Figure 7. It is obvious that once all the information has been collected, it should be easy to compare data and check for consistency. Despite the interest and benefits of this solution, it should be implemented very carefully because of the possible impact on the segmentation of the different zones. In Figure 7, we see that the operator station could be used as an unauthorized rebound between the application server and the PLC.

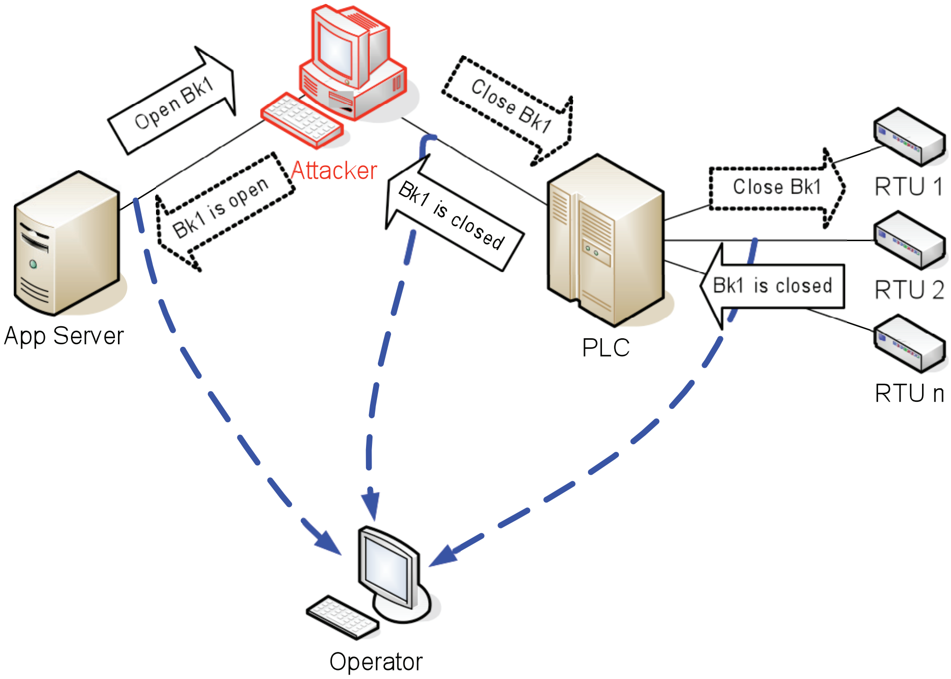


*Figure 7 – Protection Against MitM Described in [43]*

The application dynamic should permit decisions to be made based on historic considerations. Has this happened before? Is it consistent with what has been done before?

| Layer | Device | Example |
|---|---|---|
| 7 | Data range checking protecting from fuzzing on layer 7 | -50°C < t° < +50°C |
| 8 | Protecting from actions outside the industrial process logic.<br>Using heuristics to decide if an action is dangerous. | open_a_valve_allowed = fct°(Pressure, historic, t°, context)<br>if Pressure and t° are normal and we are in operation mode "running state", then accept the "open the valve" command |

*Table 5 – Layer 7 and Layer 8*

### 3.2.3 Proof of Concept

We have first developed a library handling Modbus/TCP requests and responses. This has enabled us to set up an elaborated protection scheme for these environments, taking into explicit account sequence consistency, chronology, operation mode, etc. Such a protection can address not only known attacks but also unknown ones.

The platform used to develop the proof of concept is based on Linux 2.6.24. Starting from 2.6.14, Netfilter uses a new architecture that enables interaction with userspace programs through NFnetlink, see Figure 8. Three new libraries are available from the userspace: libnetfilter_queue for userspace decisions, libnetfilter_log for logging, and libnetfilter_conntrack for connection tracking handling. We have chosen to use the NFQUEUE bindings through libnetfilter_queue, in order to delegate the decision to the userspace [47]. This library provides access to packets queued by the kernel packet filter.

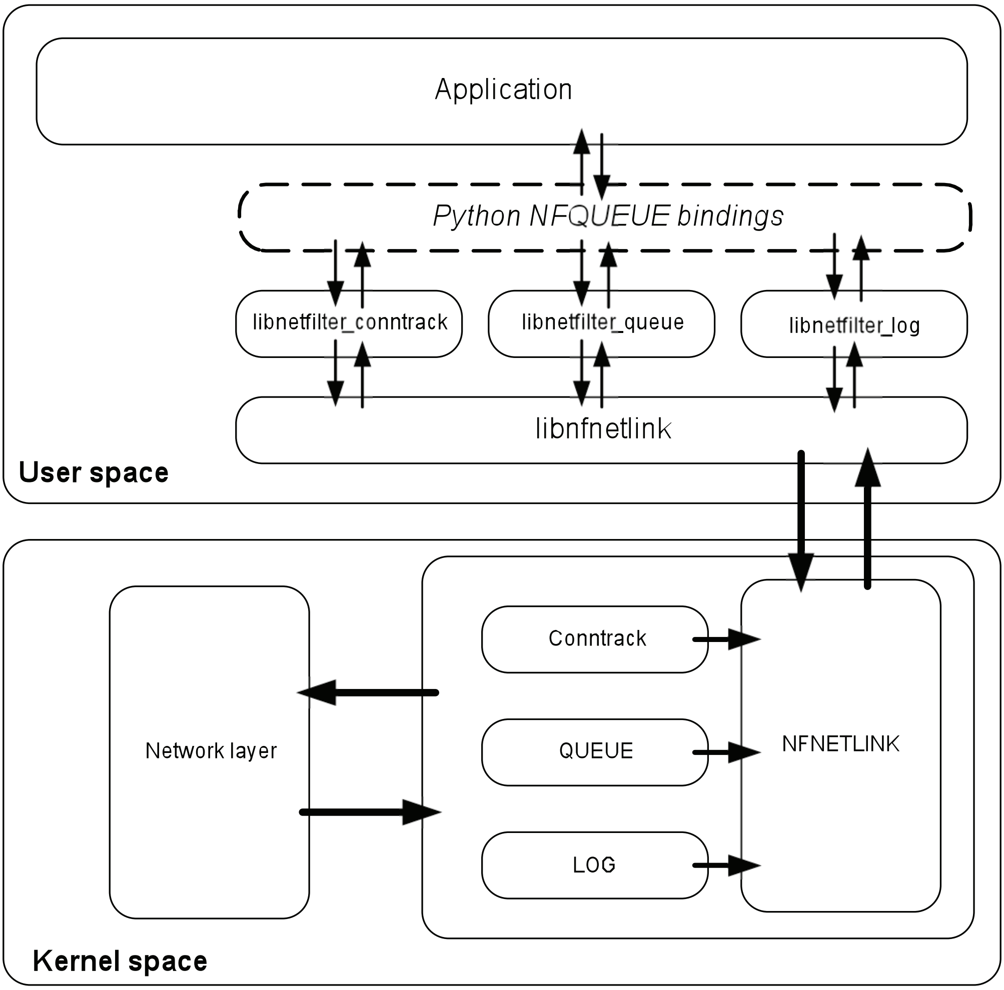


*Figure 8 – Netfilter Architecture from Kernel 2.6.14*

The decision can be to DROP or ACCEPT the packet, but the packet can also be modified.

We have used Scapy, a powerful packet manipulation program [46], to develop our Modbus/TCP implementation in accordance with the Modbus specification [45][46]. The power of Scapy and Python leverage the comprehensiveness and reusability of our development.

The main purpose of Scapy is to decode packets and let the user interpret and manipulate the data. Thus, packets can be interpreted one by one, with context, such as values of other registers, or with history and states. It allows very easy implementations of advanced protocols and tools.

The examples presented here have been developed with Modbus, the “Hello World program” equivalent for SCADA protocols.

```
class ModbusTCPReq(Packet): # This class defines ModbusTCP request packets
  name = "ModbusTCPReq"
  fields_desc = [                                   # common fields
                 ShortField("transaction_identifier",1),
                 ShortField("protocol_identifier",0), # must be 0 for Modbus
                 ShortField("length",6),   # total number of following bytes
                 ByteField("unit_identifier",1),
                 ByteEnumField("function", 3, modbus_functions),
```

The specific fields of function 3 “request packet” (read holding registers) are as follows:

```
# Specific fields - function 3 "Read holding registers"
ShortField("starting_address",0) # address 0 corresponds to register #1
ShortField("quantity_of_registers",0) # quantity_of_registers must be between 1 and
125 (0x7D)
```

The answer packet should have these fields:

```
 FieldLenField("byte_count",None,"register_value","B") # byte_count = 2*N
 FieldListField("register_value",None,ShortField,count_from=lambda pkt:
pkt.byte_count/2),
```

Example of packet manipulation:

```
>>> pkt = ModbusTCPReq()
>>> pkt
<ModbusTCPReq |>
>>> pkt.function = 6
>>> pkt.register_value = 666
>>> pkt
<ModbusTCPReq function=Write single register register_value=666 |>
>>> pkt.show()
###[ ModbusTCPReq ]###
 transaction_identifier= 1
 protocol_identifier= 0
 length= 6
 unit_identifier= 1
 function= Write single register
 register_address= 0
 register_value= 666


# send a crafted Modbus packet (assuming 3-way handshake has been performed)
>>> sr1( Ether()/IP(dst=”192.168.0.6”)/TCP(dport=502)/pkt)
```

Example of packet, as seen through our tool:

```
# function 6, request packet
<ModbusTCPReq transaction_identifier=24 protocol_identifier=0 length=6
unit_identifier=1 function=Write single register register_address=0 register_value=333
|>
# function 6, respond packet
<ModbusTCPRes transaction_identifier=24 protocol_identifier=0 length=6
unit_identifier=1 function=Write single register register_address=0 register_value=333
|>
```

```
# function 3, request packet
<ModbusTCPReq transaction_identifier=25 protocol_identifier=0 length=6
unit_identifier=1 function=Read holding registers starting_address=0
quantity_of_registers=10 |>
# function 3, respond packet
>>> Packet.show()
###[ ModbusTCPRes ]###
 transaction_identifier=25
 protocol_identifier=0
 length=23
 unit_identifier=1
 function=Read holding registers
 byte_count=20
 register_value=[333, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Once the Modbus library was developed, we have used it to easily filter packets based on their characteristics. The first step was to configure iptables to send packets to the queue:

```
# iptables –A FORWARD –p tcp –source 192.168.20.20 –dport 502 –j NFQUEUE
# iptables –L
(…)
Chain FORWARD (policy DROP)
Target   prot     opt      source            destination
NFQUEUE      tcp      --       192.168.20.20     anywhere                          tcp
dpt:502 NFQUEUE num 0
```

Developing the code for accepting only read_register functions is then obvious, using the libnetfilter_queue bindings:

```
# if not a ModbusTCP packet, then drop
if not packet.haslayer(ModbusTCP):
  payload.set_verdict(nfqueue.NF_DROP)
# if not a "read register" function, then drop
elif modbus_functions[packet[ModbusTCP].function_id] != “Read holding registers”:
  payload.set_verdict(nfqueue.NF_DROP)
# else accept !
else:
  payload.set_verdict(nfqueue.NF_ACCEPT)
```

Based on this very basic filtering function (accept only “read register” commands), we have developed our tool to implement high-level layer 8 filtering capabilities, taking into account the industrial process logic. We could also have easily used it as a “black-box recording machine” of all industrial communication going through the device.

The main pitfall of our implementation is that it is not suited for high performance requirements, but that is not a concern for the proof of concept. A development based on C and libnetfilter_queue interaction would be more appropriate if performance is required.

There are multiple advantages of using a userspace decision:

- Simpler to develop (no dependency on the kernel).
- More reliable (a crash in userspace is preferable to a crash in the kernel!).
- Enables existing library use (like packet manipulation, regular expression, etc.).

At the time Cisco had designed Modbus Firewall, there was no such possibility to do it in user space [39].

Using LOG instead of ACCEPT or DROP, we should be able to detect the suspect packets without interfering directly. If we want to process the packet without necessarily dropping it, we could process it through the Python program and return the ACCEPT decision through libnetfilter_queue.

Each device (PLC, RTU, etc.) could then have specific rules defined on a “need-to-do” basis, with processing to be defined along what the most applicable and secure scheme is at the moment, and adapted to the operation mode of the facility.

Another way to develop such proof-of-concept is to use L7-Filter. L7-Filter [48] is a packet classifier based on application layer matching. It enables the reconnaissance of protocols at the application layer and not (only) with the port numbers in use, thus making it possible for example to differentiate P2P using port 80 against classical web browsing. L7-Filter has a version working in user space. Since, for our proof of concept, we give priority to comprehensiveness and reusability of our developments, we choose not to use it.

### 3.2.4 Benefit in the Proof of Concept: from Layer 7 to Layer 8

The benefit of our proof of concept deep packet inspection firewall lies in its multiple approaches of protection, each having its value. Those are gradual measures; the operator must decide which should be used, depending on the context and the level of protection required.

The possible security functions are:

- Check compliance to Modbus specifications (including packet lengths, etc.).
- Restrict to expected sub-functions.
- Check arguments for valid range or values (static).
- Filter responses according to requests (reading and response on same register address).
- Context-based decisions, taking into account the states of devices, the history, the mode, and the dynamic of the industrial process.

Of course, if each control enhances the level of protection, it also requires particular attention for configuration.

### 3.2.5 Use of the Proof of Concept

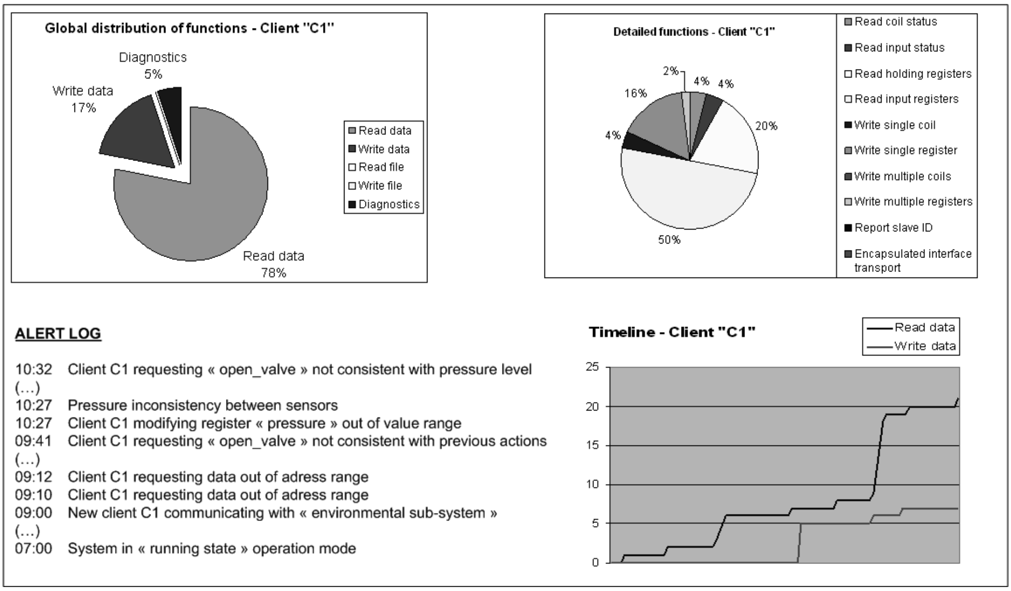
Our proof of concept implementation can enable several technical applications like:

- **Generic proxy for ICS protocols**. In particular, such a proxy could for example turn a "subscribe" type protocol into a "request/response" protocol. This could also help to control the direction of information flow
- **Blackbox recording machine** for the history of values, maintenance, or logging
    - This could monitor and record all "write" operations on critical PLC's
    - As a passive and transparent tool, it would easily be integrated into legacy architectures
    - Optionally, it could warn about misbehaviors, using the same principle as described in [50] but in a way much closer to the process logic
- **"IPS-MC"** (IPS with Memory and nodes Collaboration)

- **Transparent firewall**
    - Compatible with legacy equipment
    - Filtering requests and the corresponding responses based on process logic expectations
    - Useful to confine communication to a sub-process (for safety)
    - If more efficient when fully aware of the process, it may not necessarily watch all of the traffic to efficiently accomplish its mission (zone isolation issues)

Note that this new device would be independent from the other nodes, like PLCs or operators' desktops.

One of the most interesting uses in a production environment would probably be the set-up of a blackbox connected to a mirrored port and recording all traffic at the application level. The information collected would contain detailed process information as commands or data, enabling later analysis; this emphases the need for high-volume data analysis and representation capabilities. Figure 9 represents a typical dashboard that could be generated by such a blackbox application. In fact, a comparison with what has happened before could help to detect unexpected events, for example that a new host is communicating on the ICS network or that a new command is trying to be performed. A decision could also be made if a specific sequence of commands is performed, in coherence with process logic.



ALERT LOG

```
10:32  Client C1 requesting « open_valve » not consistent with pressure level
(…)
10:27  Pressure inconsistency between sensors
10:27  Client C1 modifying register « pressure » out of value range
09:41  Client C1 requesting « open_valve » not consistent with previous actions
(…)
09:12  Client C1 requesting data out of adress range
09:10  Client C1 requesting data out of adress range
09:00  New client C1 communicating with « environmental sub-system »
(…)
07:00  System in « running state » operation mode
```

*Figure 9 –Blackbox Device Generated Dashboard*

Generally speaking, the reactions of a layer 8 security device will be different depending on the facility mode of operation. For example, in maintenance mode, suspicious traffic could be blocked directly and raise an alert. In emergency or critical modes, with critical or safety stakes, the same suspicious activity could only trigger high-priority alarms, without active reactions like blocking traffic, if hazardous consequences were possible for such reactions.

We are currently developing and testing such a layer 8 security application on a mock-up dedicated to ICS cyber-security testing at EDF R&D facilities. They are not yet used in production environments, the security being based on a set of other measures. Nevertheless, they all appear to be promising prototypes, paving the way for more industrialized solutions to be deployed in the future in our architectures.

## 3.3 Perspectives

As stated before, our tools are still under development. More precisely, we are still currently working on the global layer 8 filtering capability, including data and application logic, in our mock-up. Developing heuristics is still needed; in particular, we need to go deeper into logic programming, interference making, and interaction modeling to describe the industrial process. This should help to sort alerts along priorities and also to assist an operator to decide if abnormal conditions require further investigations.

Other investigations and improvements have to be made to make those tools easily usable in a production environment. For this, it is necessary to help the users to model equipment as objects with attributes and to characterize their interactions. This could be done at the engineering level, ensuring that the process logic between what is on the field and what is implemented in the defense tools are the same, see Figure 10. In any case, the reference model should be based on an open format definition, XML for example, and inheritance should be established with supervisory and security levels.

We are currently extending our proof of concept implementation to other industrial protocols, some of which can be far more complex than Modbus. IEC-60870-5-104 ([53]) is presently being integrated and, in the future, we also plan to support industrial protocols like IEC61850 ([54]) or others.

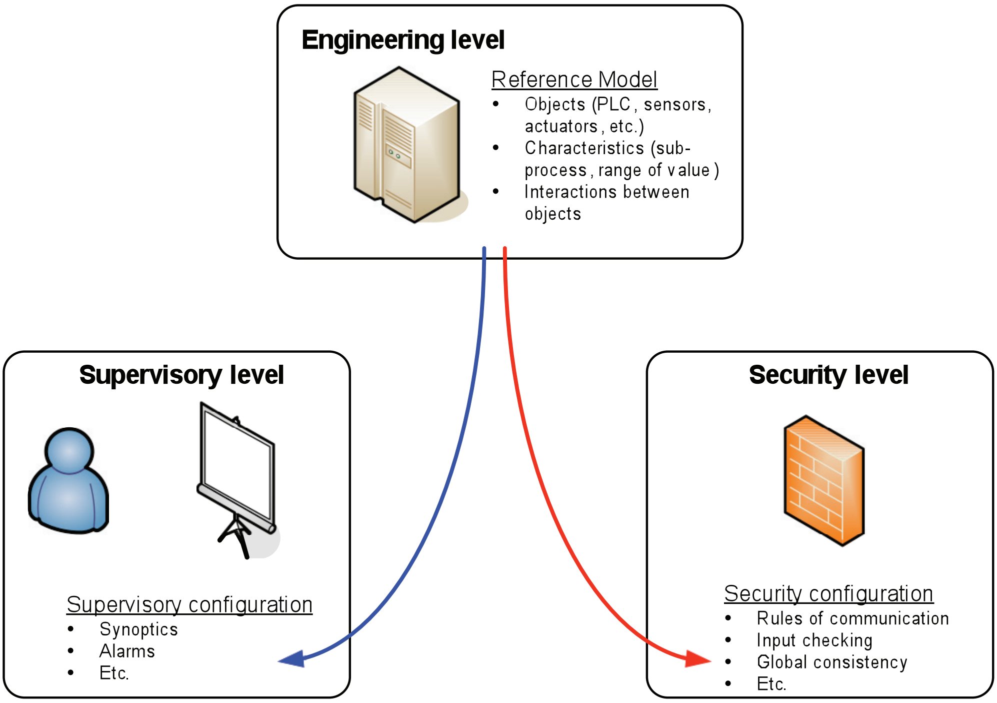


*Figure 10 – Reference Model Inheritance Designed at the Engineering Level*

Measurement methods are needed to evaluate the technical efficiency, especially to see if our prototype resists typical evasion techniques (fragmented IP packets, overlapping or reordered TCP streams, Unicode obfuscation). Performance issues should also be studied.

Finally, the applicability of such tools for real-life process controls must also be assessed through close collaboration with personnel in charge of the industrial process.

## 4 Conclusion

Physical data diodes, deep packet inspection for industrial protocols, and layer 8 security approaches are all relevant and promising security measures. Nevertheless, none of these approaches can ensure sufficient cyber-security protection alone; they have to be considered in a defense-in-depth perspective, and used in a global and consistent approach. In any case, skilled teams and appropriate organizations and procedures remain absolute prerequisites for the efficient use of any security technologies.

**About the Authors** – Ludovic Piètre-Cambacédès works as a research-engineer on computer security at Electricité de France (EDF). He is particularly interested in the architecture of and technical recommendations for the cyber security of EDF industrial critical infrastructures. He contributes as an expert to several normative and collaborative international initiatives fostering cyber security for power utilities (including for the International Atomic Energy Agency, the Comité International des Grands Réseaux (CIGRE), and the International Electrotechnical Committee).

Pascal Sitbon, CISSP and ISO27001 Lead Auditor, graduated from ENSIMAG, a French Grande Ecole (the equivalent of a MS in Computer Science). He has 10 years of experience in IT, including 7 years in IT security. His main interests are IT and Industrial Control Systems security, including technical and organizational as well as management themes. He is currently working at EDF R&D as a cyber security expert, researcher, and project manager.

## References

[1] "The Metasploit Project"; http://www.metasploit.com/.

[2] F. Cohen, "Designing provably correct information networks with digital diodes," Computer & Security, vol. 7, 1988, pp. 279-286.

[3] M. Stevens et M. Pope, Data diodes, Australia : Defence Science and Technology Organisation, 1995.

[4] J.D. Yesberg et M.W. Klink, An Investigation into the Reliability of User Datagram Protocol Reception for a Data Diode, Australia: Defence Science and Technology Organisation (DSTO), 1998.

[5] M. Stevens, An Implementation of an Optical Data Diode, Australia: Defence Science and Technology Organisation (DSTO), 1999; http://www.dsto.defence.gov.au/publications/scientific_record.php?record=4386

[6] M.H. Kang et I.S. Moskowitz, "A pump for rapid, reliable, secure communication," Proceedings of the 1st ACM conference on Computer and communications security, Fairfax, Virginia, United States: ACM, 1993, pp. 119-129; http://portal.acm.org/citation.cfm?id=168604&dl=GUIDE&coll=GUIDE&CFID=9094569&CFTOKEN=86930921.

[7] J.N. Froscher et coll., "Improving inter-enclave information flow for a secure strike planning application," Proceedings of 11th Computer Security Applications Conference, 1995, pp. 8--9.

[8] M. Kang, I. Moskowitz, et D. Lee, "A network version of the Pump," Security and Privacy, 1995. Proceedings., 1995 IEEE Symposium on, 1995, pp. 144-154.

[9] M.H. Kang, I.S. Moskowitz, et D.C. Lee, "A Network Pump," IEEE Trans. Softw. Eng., vol. 22, 1996, pp. 329-338.

[10] M. Kang et coll., "A case study of two NRL Pump prototypes," Computer Security Applications Conference, 1996., 12th Annual, 1996, pp. 32-43.

[11] M. Kang, A. Moore, et I. Moskowitz, "Design and assurance strategy for the NRL Pump," Computer, vol. 31, 1998, pp. 56-64.

[12] M. Kang, I. Moskowitz, et S. Chincheck, "The Pump: a decade of covert fun," Computer Security Applications Conference, 21st Annual, 2005, p. 7 pp.

[13] P. Lagadec, "Diode reseau et ExeFilter: 2 projets pour des interconnexions securisees," Rennes, France: 2006.

[14] D.W. Jones, RS-232 - Data Diode Tutorial and Reference Manual - Draft, University of Iowa, 2006; http://www.cs.uiowa.edu/~jones/voting/diode/RS232tech.pdf.

[15] J. Menoher, "Owl Computing Product Overview," 2007; http://www.owlcti.com/docs/Owl_product_overview.pdf.

[16] J. Menoher, "Coalition Warrior Interoperability Demonstration (CWID) 2007: A Case Study in International Cross-Domain Network Communications secured by strategic deployment of one-way data transfer systems," 2007; http://www.acsac.org/2007/casestudies/Menoher.pdf.

[17] "Validated Product - Owl Computing Technologies Data Diode Network Interface Card Version 4 (EAL4)," The Common Criteria Evaluation and Validation Scheme, 2007; http://www.niap-ccevs.org/cc-scheme/st/vid10208/.

[18] "Frequently Asked Questions - Is Owl product certified and accreditable?," Owl Computing Technologies Inc., 2008; http://www.owlcti.com/products/faqs.html#5.0.

[19] C.A. Nilsen, "Method for transferring data from an unsecured computer to a secured computer"; http://www.freepatentsonline.com/5703562.html.

[20] N. Garcia, "One-way network link keeps systems secure," Sandia National Laboratories - LabNews, 2002; http://www.sandia.gov/LabNews/LN04-05-02/key04-05-02_stories.html#owl.

[21] "Tenix America Web Page - Interactive Link Product Suite - Description and datasheets"; http://www.tenixamerica.com/products.html.

[22] "Interactive Link Data Diode Device - ITSEC E6 Certification," Austrian Government - Department of Defence, 1999; http://www.dsd.gov.au/infosec/evaluation_services/epl/network_security/Tenix_InterLinkDataDevice.html.

[23] "Validated Product - Tenix Interactive Link Data Diode Device Version 2.1 (EAL7+)," The Common Criteria Evaluation and Validation Scheme, 2005; http://www.niap-ccevs.org/cc-scheme/st/vid9512/.

[24] "Validated Product - Tenix Interactive Link Data Diode Device, Gigabit Variant (EAL 7+)," The Common Criteria Evaluation and Validation Scheme, 2006; http://www.niap-ccevs.org/cc-scheme/st/vid9513/.

[25] "Achievements in E-government - Securing Systems with Starlight," The Australian Government Information Management Office Archive, 2003; www.agimo.gov.au/archive/publications_noie/2003/06/transform/defence.

[26] "Thales EILPS-SD White Paper - Solution d'interconnexion des réseaux sensibles via une liaison monodirectionnelle (v6)," 2006; www.afina.fr/upload/Fournisseur/Thales/Thales_WP_ELIPS-SD_FR.pdf.

[27] A. Turniansky, "Unidirectional Connectivity - A Novel and Robust Method for Absolute Protection of Process Control Systems"; https://www.pcsforum.org/events/2008/documents/unidirect.pdf.

[28] L. Frenkel, "A Realistic Approach for Connecting SCADA/DCS Networks to Administrative or Less Secure Networks," 2008; http://www.waterfall-solutions.com/UserFiles/File/Entelec%202008%20-%20Synopsis.pdf.

[29] "Waterfall SCADA Monitoring Enabler - Product Brief"; http://www.waterfall-solutions.com/UserFiles/File/Waterfall%20SME%20Product%20Brief.pdf.

[30] "Fort Fox Data Diode - A Preferred Solution For High-Security Real-time Electronic Data Transfer Between Networks," 2008; http://www.datadiode.eu/whitepaper.

[31] "Fort Fox Data Diode - Product Overview," Fort Fox Data Diode Website; http://www.datadiode.eu/product.

[32] "Validated Product - Windows 2000 Professional, Server, and Advanced Server with SP3 and Q326886 Hotfix (EAL 4+)," The Common Criteria Evaluation and Validation Scheme; http://www.niap-ccevs.org/cc-scheme/st/vid4002/.

[33] "Arrangement on the Recognition of Common Criteria certificates in the field of information Technology Security," 2000; http://www.commoncriteriaportal.org/files/operatingprocedures/cc-recarrange.pdf.

[34] "The ISA99 Committee Web Page - Industrial Automation and Control System Security"; http://www.isa.org/MSTemplate.cfm?MicrositeID=988&CommitteeID=6821.

[35] Åge Torkilseng et S. Duckworth, "Security Frameworks for Electric Power Utilities – Some Practical Guidelines when developing frameworks including SCADA/Control System Security Domains," Electra, To Appear. ; http://www.cigre.org/gb/electra/electra.asp.

[36] Andrew Bartels, S. Ludovic Pietre-Cambacedes, et Stuart Duckworth, "Security Technologies Guideline - Practical Guidance for Deploying Cyber Security Technology within Electric Utility Data Networks," Electra, To Appear. ; http://www.cigre.org/gb/electra/electra.asp.

[37] "Nuclear Security Series - documents in preparation," The International Atomic Energy Agency Website, 2008; http://www-ns.iaea.org/security/nuclear_security_series_forthcoming.htm.

[38] "Cisco CIAG Netfilter extension for Modbus/TCP"; http://www.cisco.com/web/about/security/security_services/ciag/research/CRP_netfilter_extensions.html

[39] "Linux based firewall for Modbus/TCP"; http://modbusfw.sourceforge.net/

[40] Hirschmann EAGLE mGuard Firewall; http://products.hirschmann.de//index.php?sessiontoken=20081109012547-11111111&manid=4&CFID=&CFTOKEN=&pbparam=tid=996,spid=2,maid=1,view=1,pid=9291

[41] "Idaho National Laboratory Supervisory Control and Data Acquisition Intrusion Detection System (SCADA IDS)"; Jared Verba and Michael Milvich

[42] TOFINO; http://www.byressecurity.com/pages/products/tofino/

[43] "Using Model-based Intrusion Detection for SCADA Networks"; S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, A. Valdes, S4 2007

[44] "Filtering with Cisco IOS" (in French); http://www.hsc.fr/ressources/breves/cisco-acl.html.fr

[45] Modbus Application protocol specification V1.1b; Modbus-IDA, December 2006, http://www.modbus-ida.org

[46] Modbus messaging on TCP/IP implementation guide V1.0b; Modbus-IDA, October 2006, http://www.modbus-ida.org

[47] Intrusion Detection methods and tools (in French); Ludovic Mé , Supélec Rennes

[46] Scapy; http://www.secdev.org/projects/scapy/

[47] NFQUEUE bindings; http://software.inl.fr/trac/wiki/nfqueue-bindings

[48] L7-Filter; http://l7-filter.sourceforge.net/

[49] Integrating Model Checking with the Industrial Design of Interactive Systems; K. Loer and M. Harrison, University of York, UK

[50] Modeling Flow Information and Other Control System Behavior to Detect Anomalies; Brian Moran, Rick Belisle, S4 2008

[51] Checkpoint upgrade to UTM-1 Edge Appliances announce, http://www.checkpoint.com/press/2008/utm-1-edge-upgrade-111808.html

[52] SCADA Network Intrusion Detection Systems (IDS) signatures, http://www.digitalbond.com/wiki/index.php/SCADA_IDS_Signatures

[53] IEC 60870-5-104, http://webstore.iec.ch/Webstore/webstore.nsf/0/27D9ACDA8C3BBBC8C125727F00581447

[54] IEC 61870, http://webstore.iec.ch/Webstore/webstore.nsf/artnum/030525?opendocument