




# PENTESTING ICS 101

Arnaud SOULLIE
Alexandrine TORRENTS

WAVESTONE

# Who are we?



## Arnaud Soullié

Manager and security auditor

@arnaudsoullie

## Interests

- **Windows Active Directory**: Can a Windows AD be secured ? JSSI 2014 (French, sorry)
- **SCADA** (BHEU14, HIP15, BruCon, BSLV15, DC24)
- **Wine tasting / Motorbike riding** (we're not going to talk about it today)
- **Sorry for the French accent**



# Who are we?



## Alexandrine Torrents

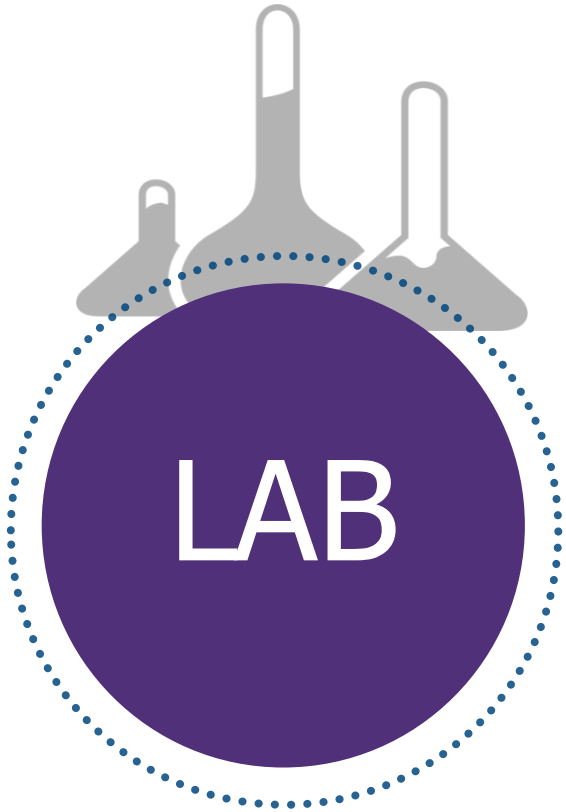
Security auditor

@DrineTorrents

### Interests

- **SCADA** (BruCON), and securing ICS, especially in the scope of the French military law, enforcing companies offering a vital service to the nation to comply to security rules
- **Penetration testing**, including IoT, hardware hacking and security assessment on ICS ;)

# LAB PREREQUISITE
## What's in the lab VM?




### KALI LINUX

### ADDITIONAL TOOLS

- ModbusPal
- Mbtget
- Plcscan
- Snap7
- …

### SCRIPTS

- PCAP samples
- Scripts skeletons
- …

**tinyurl.com/dc26-ics101**
**The VM is also available on USB stick**

# AGENDA

/ 01 Introduction to ICS

/ 02 What's wrong with ICS security?

/ 03 Programming PLCs — Hands-on !

/ 04 Pentesting PLCs — Hands-on !

/ 05 Capture the flag ! — Hands-on !

/ 06 Securing ICS





# ICS Introduction

# Where do we find Industrial Control Systems ?









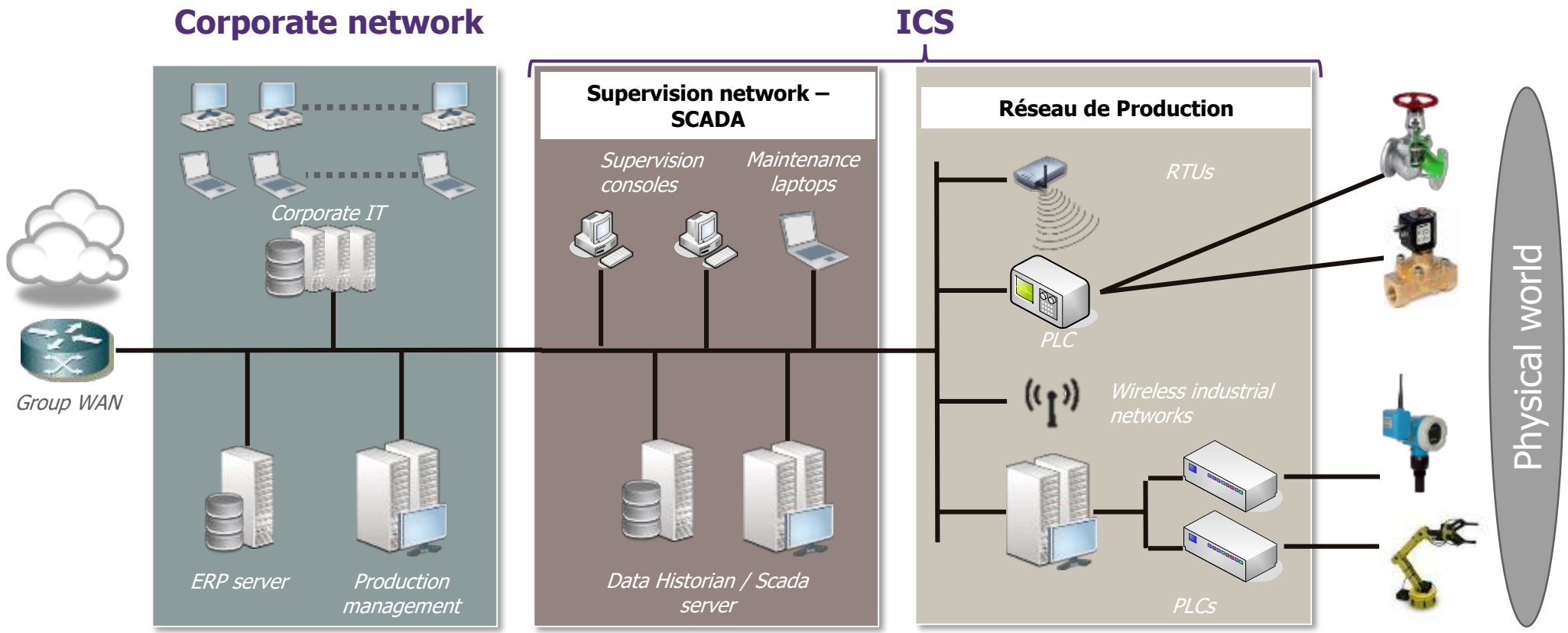


Manufacturing plants
Food

Power plants
Building automation systems (AC/HVAC/..)
Water treatment
Pharmaceutical manufacturing
Chemical plants

But also…swimming pools, building heating system, dams, etc.

# What is an Industrial Control System (ICS)?




Corporate IS handle data

≠

ICS handle interfaces data with physical world (cyber-physical systems)

# What about IoT / smart stuff ?

## Fit the definition of « cyber-physical » systems

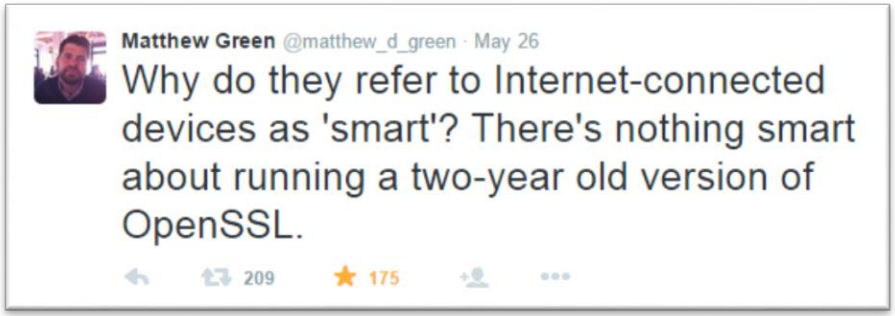
/ Cardio-meter for your smartphone

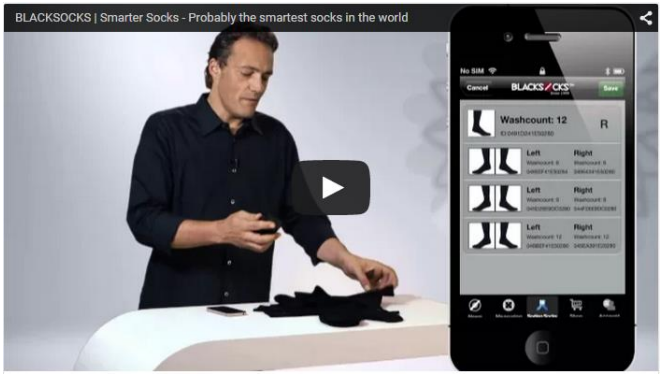
/ « Smart » electrical plugs

http://weputachipinit.tumblr.com/



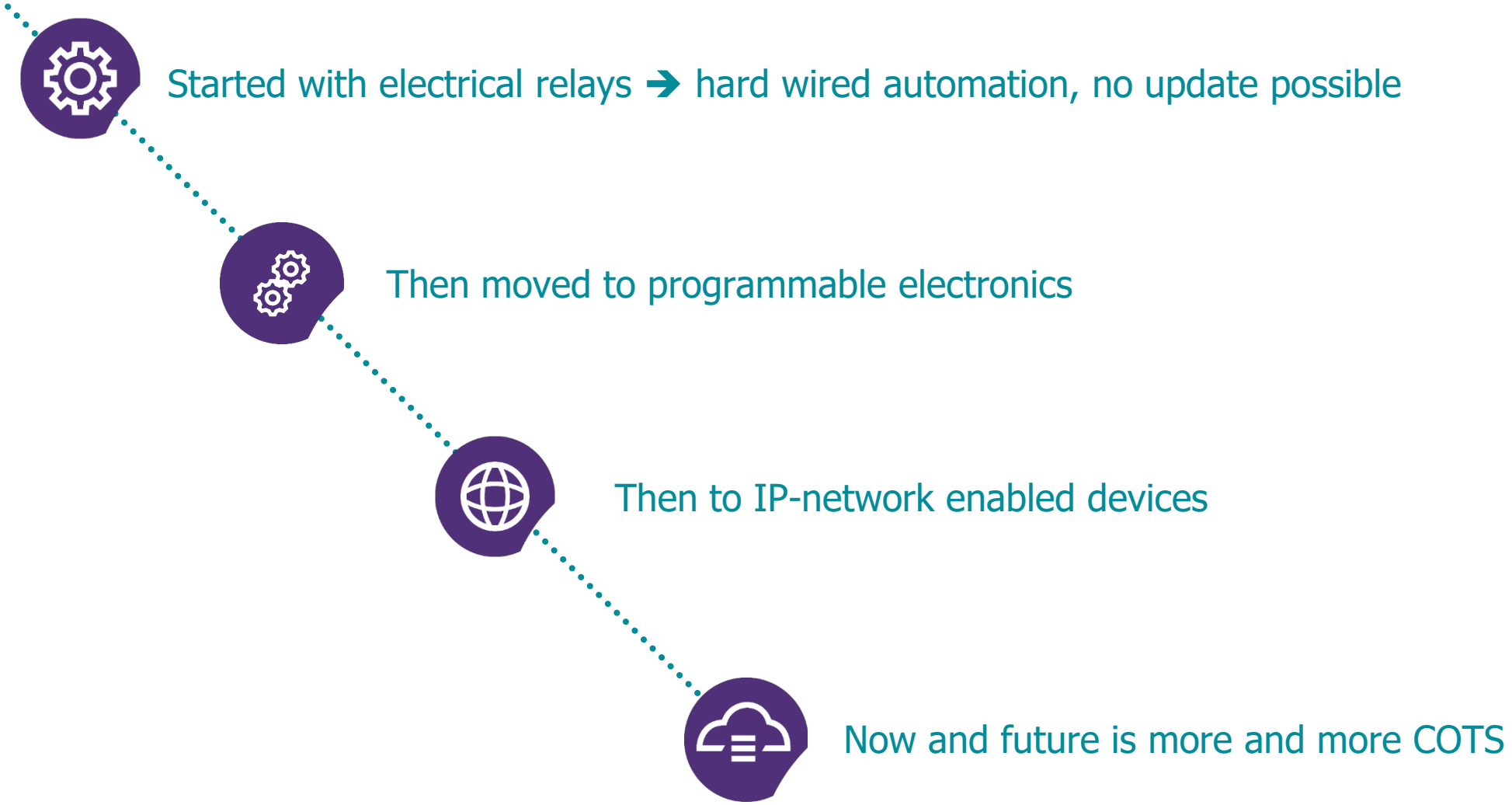

## By the way, « smart » clearly isn't the right word



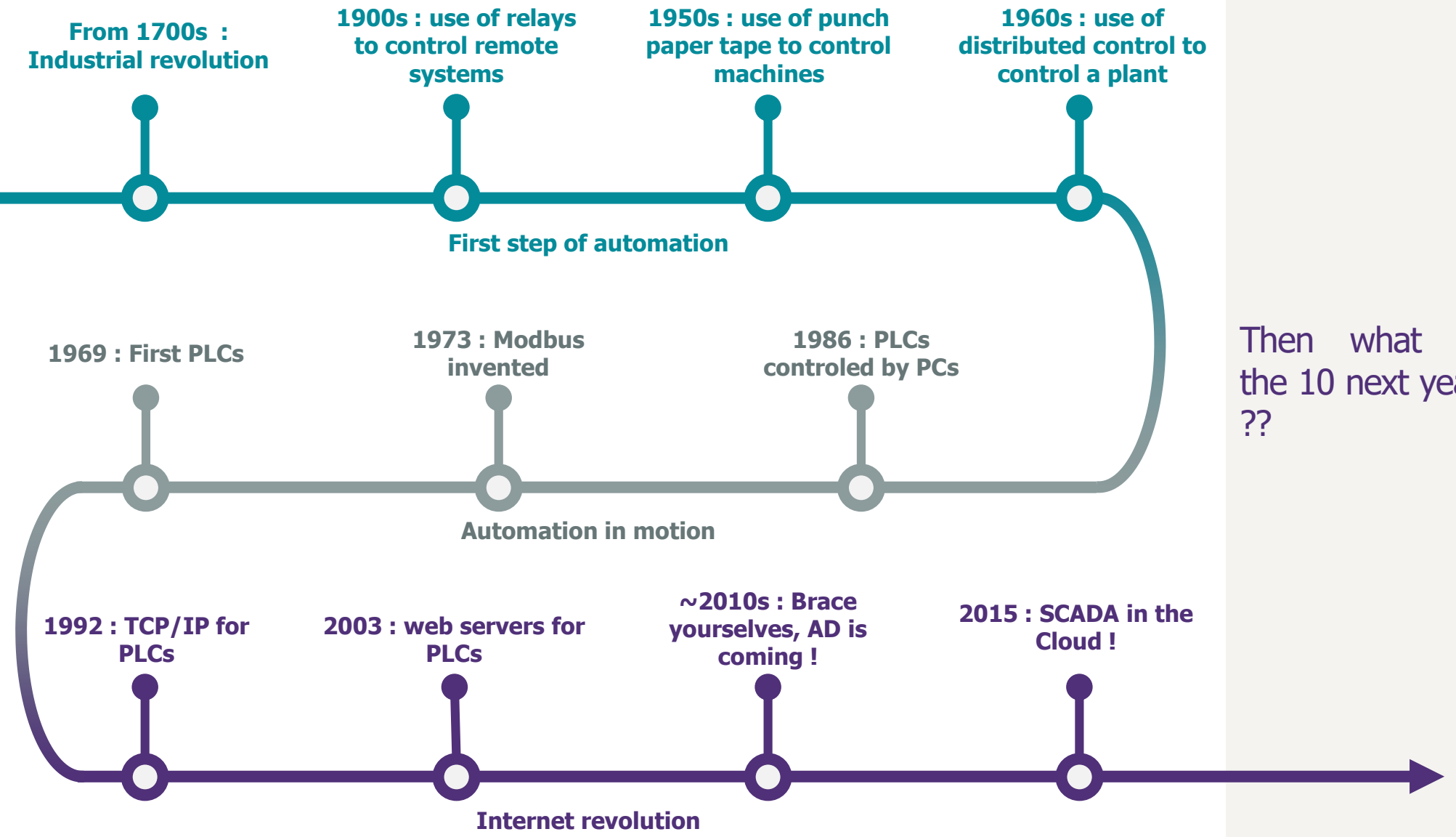




➔ Not in the scope of this training

# Evolution of ICS



# ICS evolution timeline




Then what for the 10 next years ??

# Future of ICS (well at least, current trends)

Now and future is more and more COTS

Active Directory domains

Soft-PLCs : CodeSYS under Windows computers



**CODESYS Control RTE: Real-time SoftPLC under Windows**

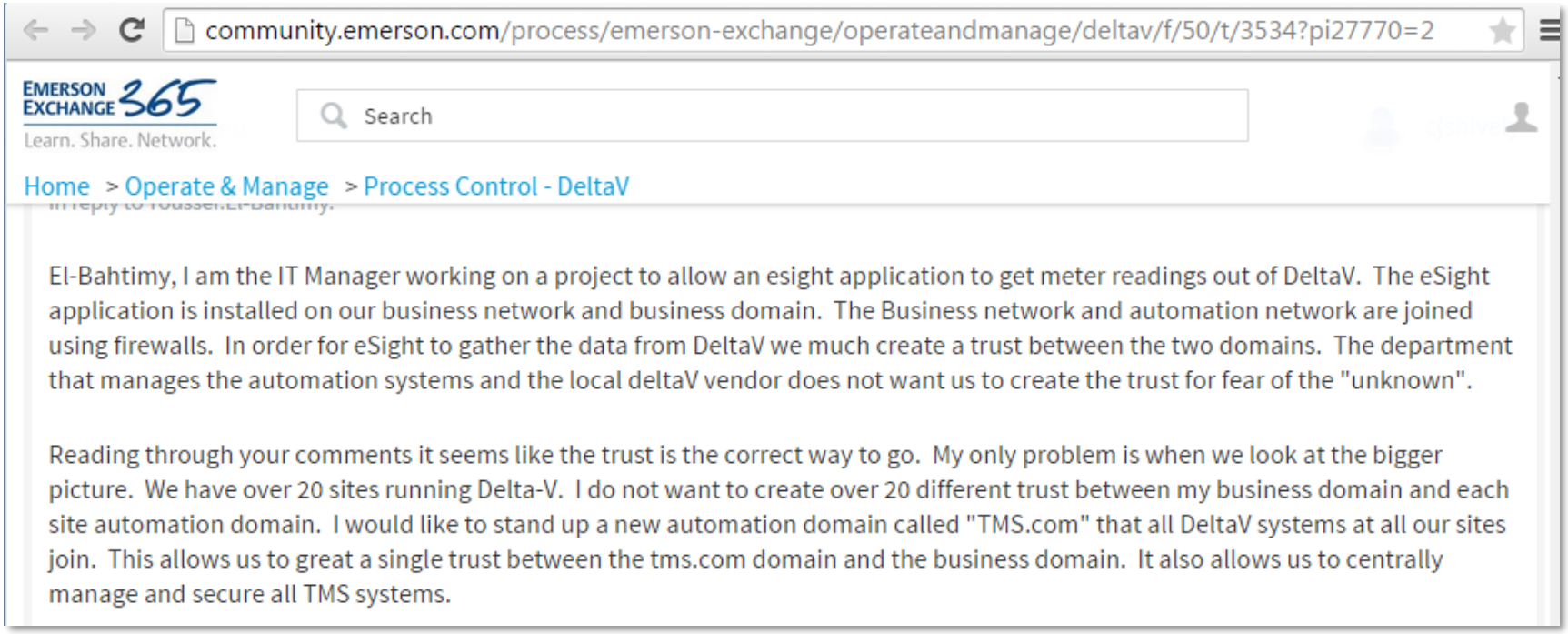
CODESYS Control RTE: Real-time SoftPLC for PC-based industrial controllers (PLCs) under Windows, programmable with the IEC 61131-3 Development System CODESYS

Overview | Howtoget | Download

**CODESYS Control RTE**

- Executable under Windows XP/Embedded/7/8 (32/64 Bit) with single or multi core CPUs
- Has its own real time kernel: deterministic behavior with jitter values in the µs region without additional hardware components or operating system extensions
- Supports several interfaces to inputs and outputs: Discrete I/O cards or standard fieldbus cards
- Supports the configuration of the standard fieldbuses directly in the IEC 61131-3 Development System, no external tools required
- Adaptable with a special toolkit (CODESYS RTE Runtime System Toolkit): API driver interface for customized I/O drivers and C functions and SoftPLC-specific configurations
- Supports remanent data: hard disk, flash card, battery buffered SRAM card

# Future of ICS (well at least, current trends)




*This is what happens when you start using too much IT in OT*

# A bit of vocabulary

**ICS** (Industrial Control System)

=

**IACS** (Industrial Automation and Control Systems)

≃

**SCADA** (Supervisory Control And Data Acquisition)

≃

**DCS** (Distributed Control System)

Nowadays, people tend to say “SCADA” for anything related to ICS

# SCADA vs DCS

## In theory …

- SCADA : event / data acquisition driven
- Used across several sites (even at the country scale)
- Can work even when offline
- Low response time

- DCS : process driven
- Limited to local process monitoring
- DCS works as standalone system
- Dedicated products by the vendors for specific industries / process

## In reality …

In the real world, you'll find some PLCs even when DCS is used

Today, SCADA manufacturer tend to have DCS functionality, while DCS systems response time is lowering to be comparable to traditional SCADA response time

# SCADA vs DCS

## *Siemens vision (1/3)*

| Characteristic | PLC | DCS |
|---|---|---|
| Market Introduction | 1960s | 1975 |
| Replacement of . . . | Electromechanical Relays | Pneumatic & Single-Loop Controllers |
| Products Manufactured . . . | "Things" | "Stuff" |
| Classic Application | Automotive | Refining |
| Type of Control | Discrete | Regulatory |
| Redundancy | "Warm" Backup | "Hot" Backup |
| Engineering Mindset | "Programming" | "Configuration" |
| Operator Interaction | Exception Basis | Man in the Loop |
| Operator Interface | Simple Graphics | Sophisticated Graphics |
| Size/Footprint | Compact | Large |
| Up-front cost | $$ | $$$$ |
| System | "Open" | "Closed" (Proprietary) |

*Table 1 Classic Stereotypes of PLC vs. DCS*

*From « DCS or PLC? Seven Questions to Help You Select the Best Solution" http://w3.siemens.com/mcms/process-control-systems/SiteCollectionDocuments/efiles/pcs7/support/marktstudien/PLC_or_DCS.pdf*

# SCADA vs DCS

## *Siemens vision (2/3)*

| PLC | DCS |
|---|---|
| ❑ Manufacturing or assembly of specific items (aka "Things") | ❑ Involves the combination and/or transformation of raw materials (aka "Stuff") |
| ❑ Product is visible as it moves through the process | ❑ Often impossible to visually see the product as it moves through the process |
| ❑ High-speed logic control (such as motors) | ❑ Regulatory/Analog (loop) control |
| ❑ Simple Batch control | ❑ Complex Batch Control |
| ❑ Value of the individual component being manufactured is relatively low | ❑ The value of a "batch" can be very high (either in raw material cost or market value) |
| ❑ Downtime mainly results in lost production | ❑ Downtime not only results in lost production, but can result in dangerous conditions |
| ❑ Downtime does not typically damage the process equipment | ❑ Downtime can result in process equipment damage (product hardens, etc.) |
| ❑ Return to steady state production after an outage is short and relatively straightforward | ❑ Return to steady state production after an unplanned outage can be long, expensive, and difficult |
| ❑ Typically, the heart of the system is the controller | ❑ Typically, the heart of the system is the HMI |
| ❑ The operator's primary role is to handle exceptions | ❑ The operator's interaction is typically required to keep the process in its target performance range |
| ❑ Status information (On/Off, Run/Stop) is critical information for the operator | ❑ Faceplates and analog trends are critical to "see" what is happening to the process |
| ❑ Exception-based alarming is key information for the operator | ❑ Alarm management is key to safe operation of the process and for responding effectively during plant upset conditions |
| ❑ Manufacturing might be able to run "lights-out" | ❑ Failure of the HMI could force the shutdown of the process |

# SCADA vs DCS

## *Siemens vision (3/3)*

| PLC | DCS |
|---|---|
| ❑ Fast logic scan (approx. 10ms) is required to perform motor or motion control | ❑ Control loops require deterministic scan execution at a speed of 100 to 500 ms |
| ❑ Redundancy may not be cost justified | ❑ System redundancy is often required |
| ❑ System can be taken offline to make configuration changes | ❑ Online configuration changes often required |
| ❑ Analog Control: Simple PID only | ❑ Analog Control: Simple to advanced PID control up to Advanced Process Control |
| ❑ Diagnostics to tell you when something is broken | ❑ Asset Management alerts you to what might break before it does |
| ❑ High level programming languages are available for creating custom logic | ❑ Custom logic created from existing function blocks |
| ❑ Customized routines usually required | ❑ Many algorithms (i.e. PID) are complex and do not vary among applications |
| ❑ Standard libraries considered nice features | ❑ Standard application libraries are expected (function blocks and faceplates) |
| ❑ Provisions must be available to integrate functions/ products into an integrated architecture | ❑ Entire system is expected to function as a complete solution |
| ❑ Program/configure individual components, integrate later (bottom-up) | ❑ Up-front design of complete system before implementation begins (top-down) |
| ❑ Desire customizable platforms to build upon | ❑ Looking for significant "out-of-the-box" functionality |
| ❑ System designed to be flexible | ❑ System designed to make it "easy" to engineer process applications |
| ❑ Solution is generic in nature, to be applied on a wide variety of applications | ❑ Use of pre-defined, pre-tested functions saves time |
| ❑ Use ladder logic to configure application | ❑ Use function block diagram to configuration application |

# ICS COMPONENTS

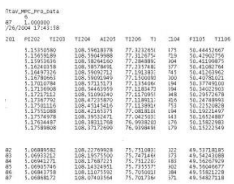
- **Sensors and actuators**: allow interaction with the physical world (pressure sensor, valves, motors, …)
- **Local HMI**: Human-Machine Interface, permits the supervision and control of a subprocess
- **PLC**: Programmable Logic Controller : manages the sensors and actuators
- **Supervision screen**: remote supervision of the industrial process
- **Data historian**: Records all the data from the production and Scada networks
- **RTU**: Remote Terminal Unit (standalone PLC)
- **IED**: Intelligent Electronic Device (smart sensor)



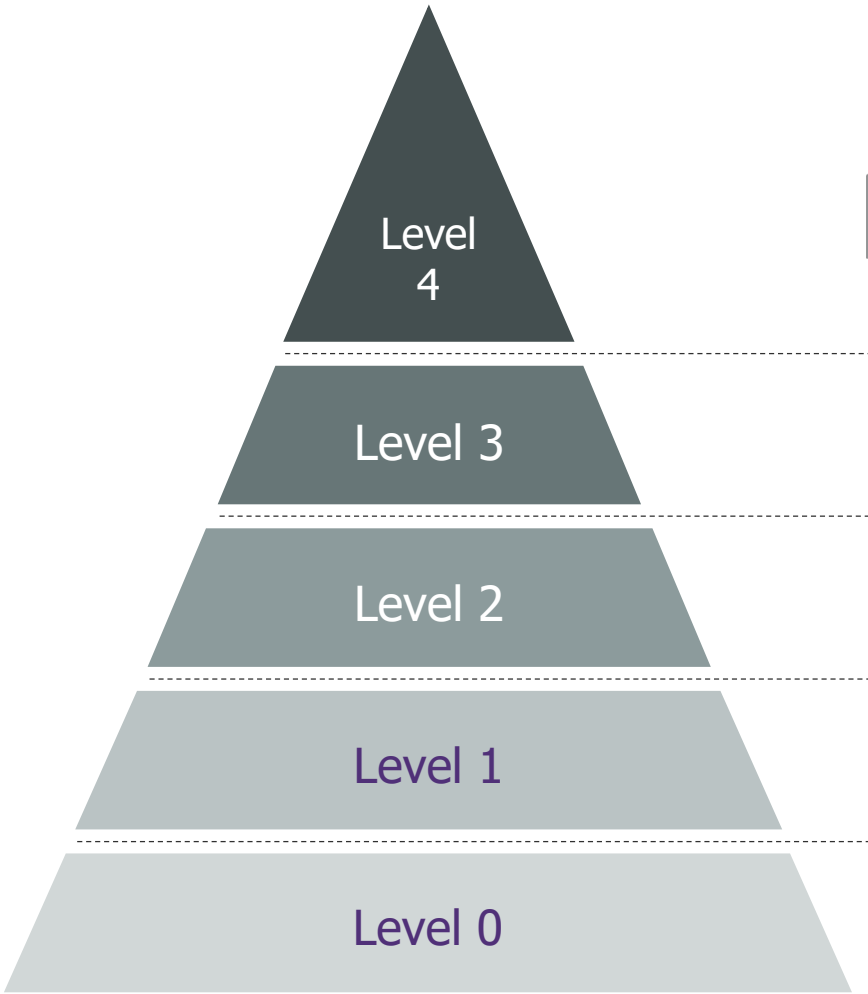






# CIM (Computer Integrated Manufacturing) pyramid






**Global planification (ERP)**
Orders and stock management, clients and accounting



**Production management (MES)**
Execution and control of manufacturing, scheduling

**SCADA**
Of an industriel process

**PLCs**

**Sensors and actuators**

# Industrial protocols

At the beginning, specific protocols on specific **physical layer** (RS232, RS285, 4-20 current loop)

Some protocols were **adapted to TCP/IP**, like Modbus, and other were developed to allow interoperability.

Currently, the most used seem to be :

- HART / Wireless HART
- Profibus
- Modbus
- Profinet / S7
- DNP3
- For the field protocols (used by the PLCs, and some « intelligent » sensors/actuators)
- OPC for data exchanged with the corporate network / Windows-based systems

# ICS vendors


EMERSON
Industrial Automation
Schneider Electric
ABB
OMRON
HIRSCHMANN
Rockwell Automation
PHOENIX CONTACT
eWON
MACHINES CAN TALK
SEMAPHORE
TBOX
aspentech
SIEMENS
YOKOGAWA
Honeywell

# IT vs OT




Lifetime of components span **over decades**



The essential criteria for ICS security is **availability**, not confidentiality



The use of COTS and standard protocols is relatively new

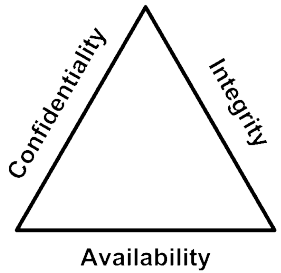


ICS were designed to be isolated, but today need to **communicate with the outside world**

No security awareness

# IT vs OT

## IT




## OT

**Availability** : My system is working correctly
**Integrity** : My system is working as expected

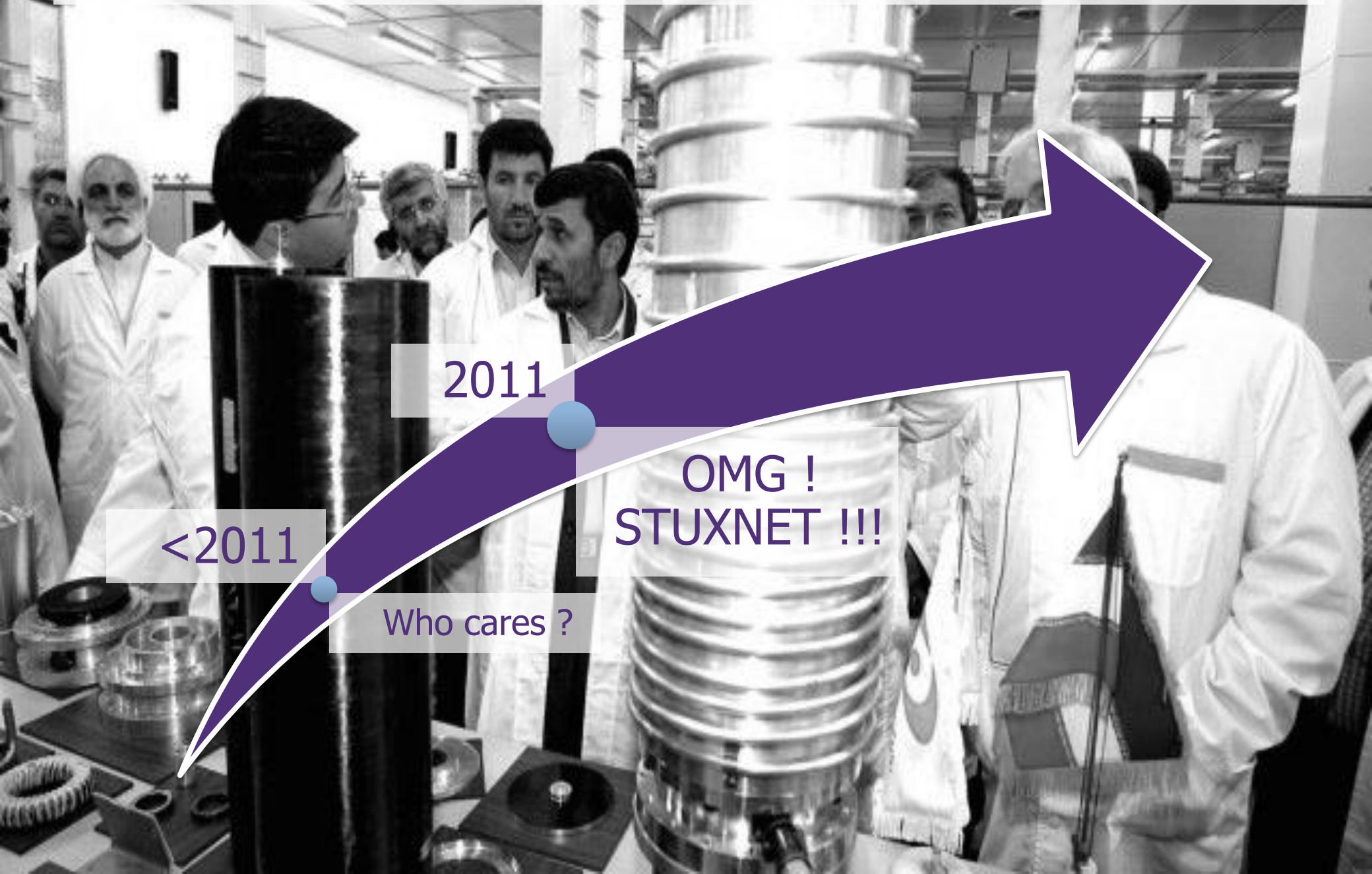
OT cares more about **safety**

| | | |
|---|---|---|
| Applied regularly on standards systems | Security patches | Only recently provided by vendors. Applied once a year tops |
| Deployed on all Windows machines, centrally managed | Antivirus | Sowly starting to show up. Some vendors used to / stil forbid AV usage (lack of support if AV is installed)<br>Not managed |
| Centralized, dedicated teams, standard operations and procedures | System Administration | Lack of local skills, heterogeneous environments, lots of different tools to use, vendor support is mandatory |
| Nominative accounts | IAM | Generic, shared accounts, no password policy |
| Service interuptions are OK, especially outside business hours | Availability | Real-time operations, downtime is unacceptable or very costly |
| Standard, TCP/IP protocols that include authentication and encryption | Protocols | Lot of vendor-specific protocol<br>no security built in |
| No people endangered | Impact | Possible impact of people, environnent, industrial gear |

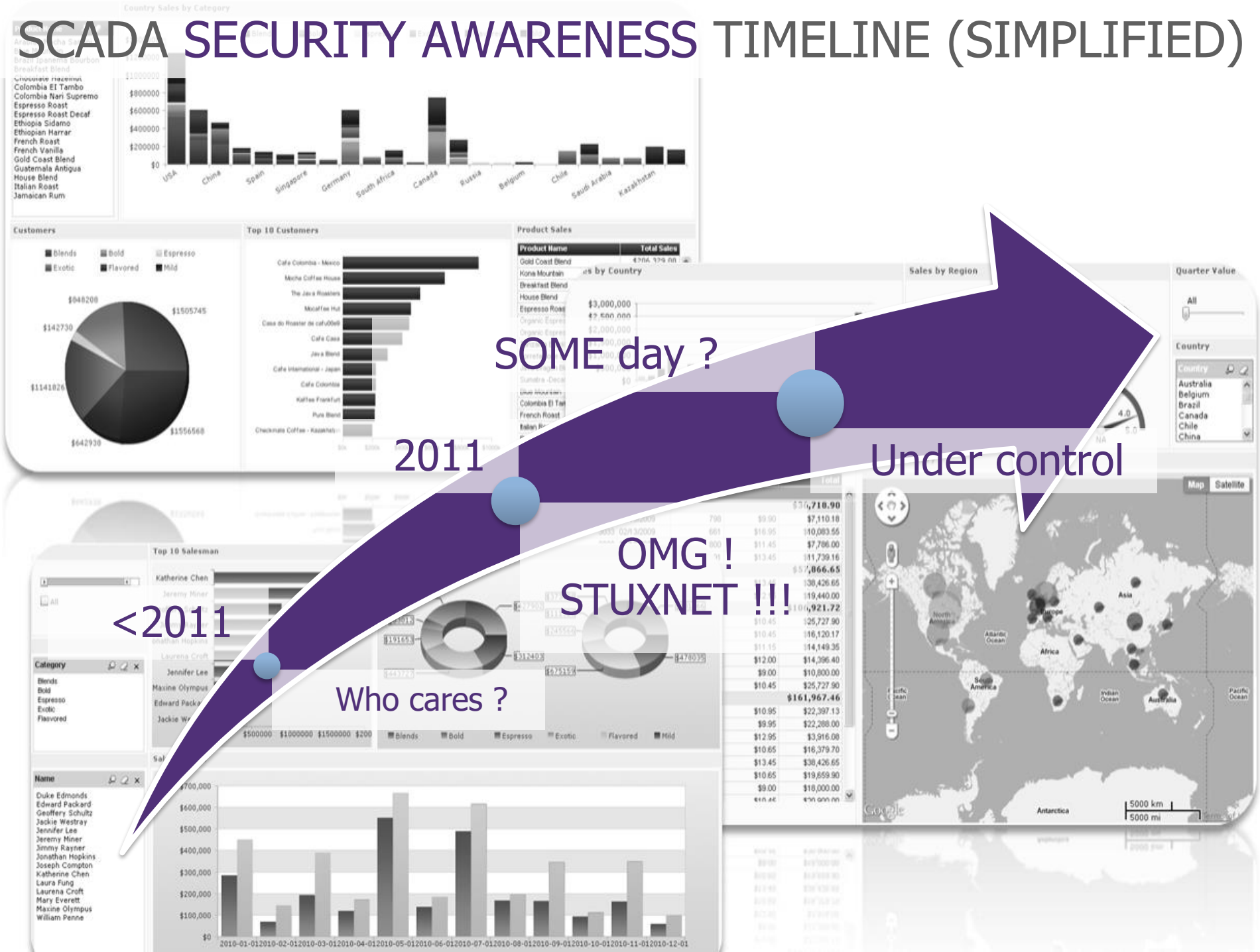
# SCADA SECURITY AWARENESS TIMELINE (SIMPLIFIED)



# SCADA SECURITY AWARENESS TIMELINE (SIMPLIFIED)



# SCADA SECURITY AWARENESS TIMELINE (SIMPLIFIED)



# ICS security awareness is growing

**At the government level** → Regulations

**In industrial companies** → Policies

**For the general audience** → Fear

© WAVESTONE

# Vendors Cyber-Security offer

Most ICS vendors have now understood the client's worries about security and offer different kinds of dedicated products and/or services

That DOES NOT mean that vendors' staff has the required knowledge and training about ICS security



## Cyber Security Services

Schneider Electric addresses an organization's compliance and cyber security challenges from analysis through to implementation and management. We start by providing expert consulting with a track record of global success to assess the current compliance situation. We then define an overall Cyber Security plan and remediation strategy encompassing processes, procedures, people, products, networks and applications.

The Cyber Security portfolio ensures regulatory compliance for:

- Power Generation: NERC CIP
- Nuclear: 10 CFR 73.54, NEI 08-09, Regulatory Guide 5.71
- Chemical and Water: CFATS
- Petroleum, Oil and Gas: Internal or Industry regulations

Our solution is unique because it provides Cyber Security compliance for a company's critical infrastructure and integrates seamlessly between manufacturing operations and corporate Information Technology networks.

**re aspects of Managed Security Service**

ecurity incident response and resolution guidance

nsite re-assessment of security posture, personnel raining and process enhancement

ontinuos monitoring of security and system status

roactive threat notification based on real-time global intelligence

Implementation of security updates to adress changing threat landscape





# What's wrong with ICS security?

# What is wrong with current ICS security?



# Organization & awareness



## No true ICS cybersecurity sector

/ ICS security does not have the same level of maturity as IT in general

/ You will often face situation were nobody is in charge of ICS security

/ Sometimes, there is even nobody in charge of IT (computers, switches)

/ Someone is in charge of ICS safety, but not security

/ ICS are often still out of the CISO perimeter

## No representative on the field

/ Very few awareness on cyber risks

/ No formation on information systems

## No budgetary line for ICS cyber-securing

**Misconceptions**

SAFETY != SECURITY

Our ICS are not connected

Proprietary protocols are safer

I have an Antivirus, I am safe

$BIG_VENDOR products are certainly secure

The safety systems will prevent anything bad from happening

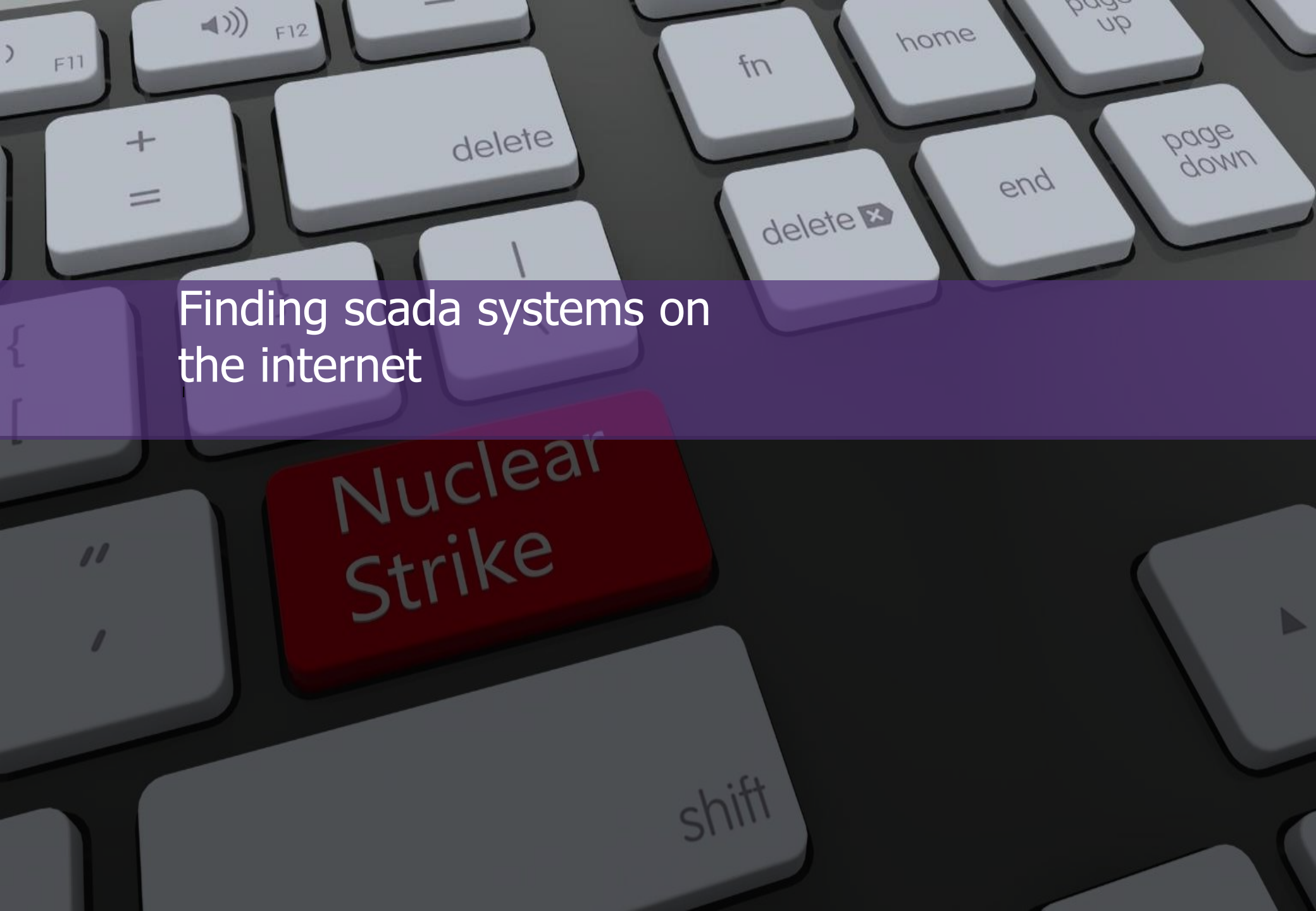
# Network segmentation

| Business needs | Security needs |
| --- | --- |
| Send information to the corporate network<br>/ Production supervision<br>/ Bills issuing<br>/ Etc.<br>Allow remote maintenance<br>➔ More and more interconnections with ICS | Unidentified! |

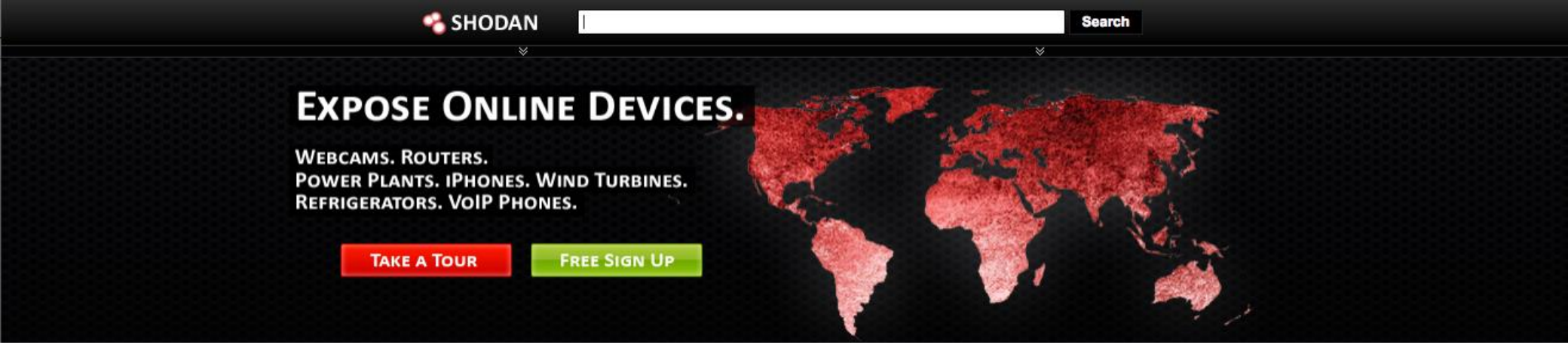
**Filtering often wrongly done, with non-dedicated equipment**
**A lot of "dangerous" flows accessible from the corporate network**

# Network segmentation

- Mostly a fail
- Nothing is air-gapped, ever (well, almost)
- « It is segmented because my laptop can’t connect to both networks at the same time »
- Often, poorly configured ACLs on routers
- Wide access to the ICS from the
  - Port 80 allowed to everyone
  - Spoiler alert : port 80 allows to do plenty of nasty things
- How good is your network segmentation if you directly copy files from the office network to the ICS network ?




# Finding scada systems on the internet




- Shodan is a search engine dedicated to find devices exposed to the Internet
- It regularly scans the whole Internet IPV4 range (~4,3 billions IPs)
- Results are partially free (you have to pay to export the results)

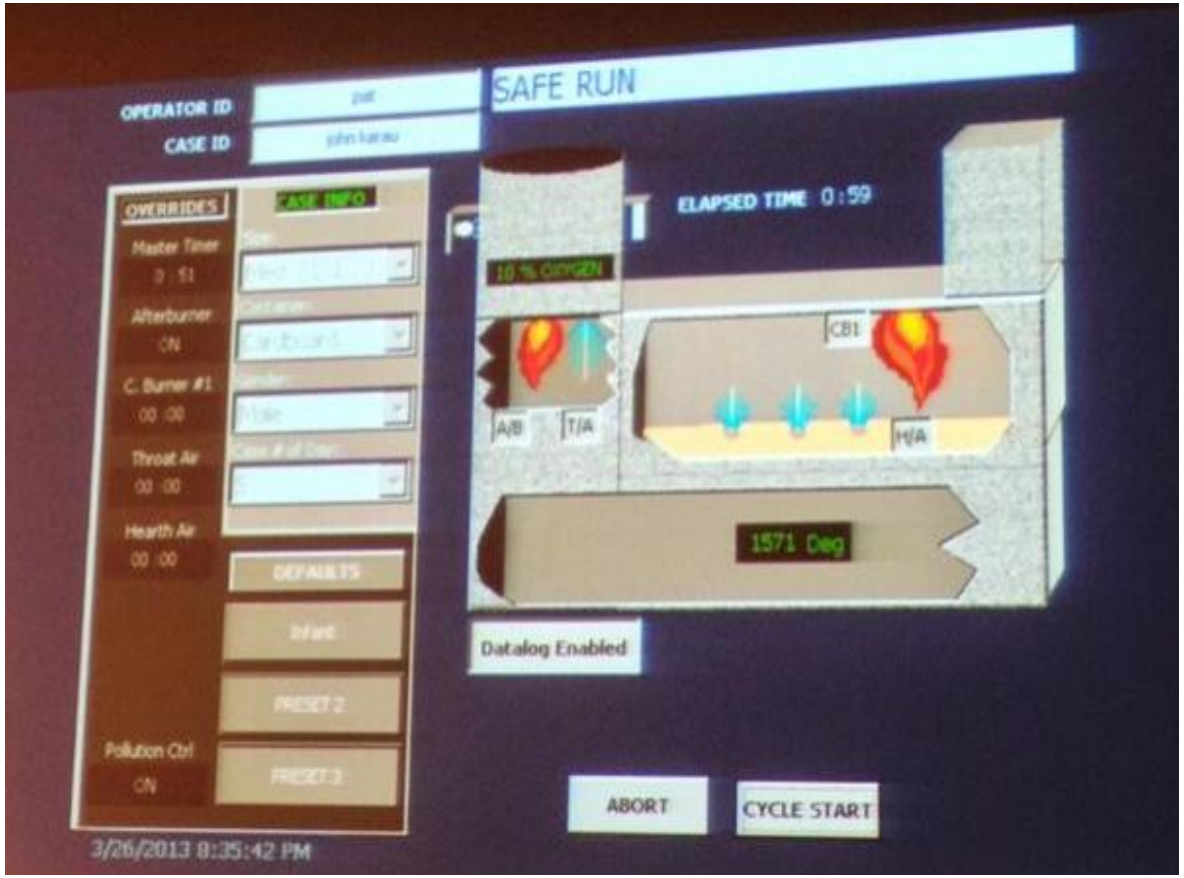
## What can you find?

- All kinds of connected devices
  - PLCs
  - Webcams
  - Smart-things (fridge, TV, …)
- Things you can't even imagine…
- Example ICS report: https://www.shodan.io/report/l7VjfVKc

## Alternatives?

- Scan the Internet yourself (Zmap, Massscan)
- Other online services/surveys

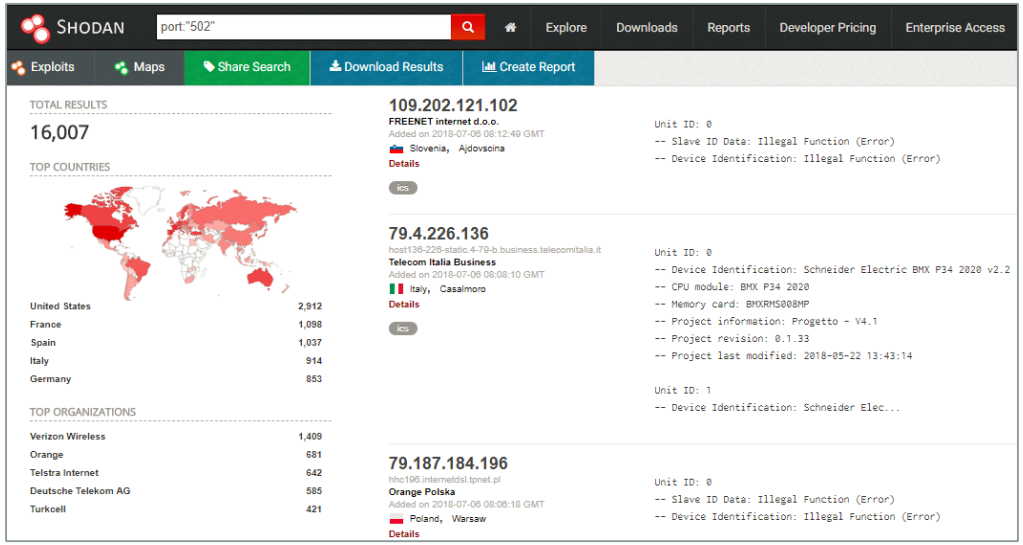
# FUNNy things you can find on teh interwebs
# It's not just webcams.

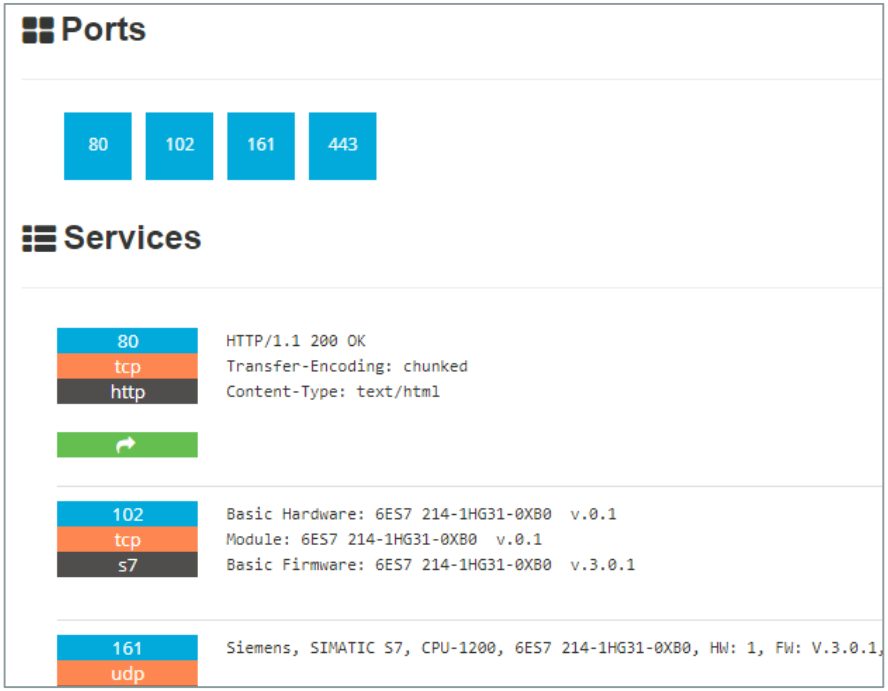





This is a crematorium.
On the internet.

# PLCs you can find on SHODAN




502
tcp
modbus

**BMX NOE 0100** Version: V3.10

```
Unit ID: 0
-- Device Identification: Schneider Electric BMX NOE 0100 V3.10
-- CPU module: BMX P34 1000
-- Memory card: BMXRMS008MP
-- Project information: Station - V4.0   EMA C:\USERS\UTILISATEUR\DOCUMENTS\CENTRALES DOSSIERS\CANCELEC\M340
-- Project revision: 0.8.124
-- Project last modified: 2018-04-18 11:41:49

Unit ID: 1
-- Device Identification: Schneider Electric BMX NOE 0100 V3.10
```

Research options:

- **IP address**
- **Object type**: PLC, HMI
- **Constructor**: Schneider, Siemens
- **Model**: M340, S7-1200
- **Protocol**: Modbus, S7
- **Port**: 502, 102

# Vulnerability management : a hard topic

## Impossibility to patch some components as it requires a stop and a restart

- ICS help industries make money. Most of the time, by producing something. The more « uptime » the ICS has, the more money you make. That is why a lot of ICS run 24/7
- Each production stop costs money, hence the difficulty to regularly apply security patches.

## Difficulty to implement a watch on vulnerabilities and security patches

- Non exhaustive or lack of cartography on installations
- Obsolete components which support is no longer assured

## No test environment to evaluate impact on production or safety of the security patches

## As ICS hardware is much more costly than VMs, sometimes security patches must be applied to production directly.

# Lack of security mechanism in equipment and protocols

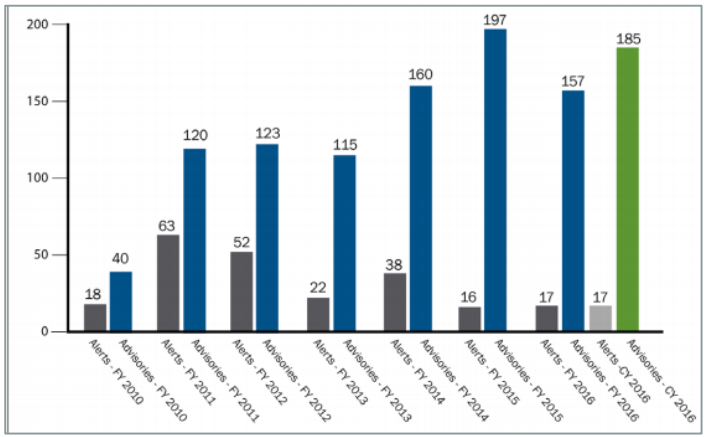
**Technologies axed on availability and longevity that do not take into account security concerns**

## Frenquent vulnerabilities on ICS protocols

- Information exchanged in clear-text
- Replay possibilities
- Lack of authentication

## Frenquent vulnerabilities on PLCs

- Weak authentication
- Default password
- « *Hardcoded* » password
  *(Schneider took 2363jd to provide an update correcting a hardcoded password, and still not on all PLCs…)*




**Alerts**

ICS-ALERT-17-341-01 WAGO PFC200, December 7, 2017

**Advisories**

ICSA-17-355-01 Moxa NPort W2150A and W2250A, December 21, 2017

ICSA-17-355-02 Schneider Electric Pelco VideoXpert Enterprise, December 21, 2017

ICSA-17-341-03 PHOENIX CONTACT FL COMSERVER, FL COM SERVER, and PSI-MODEM/ETH, December 7, 2017

ICSA-17-334-01 Siemens SWT3000, November 30, 2017

ICSA-17-334-02 Geovap Reliance SCADA, November 30, 2017

ICSA-17-332-01 Siemens SCALANCE W1750D, M800, and S615, November 28, 2017

ICSMA-17-332-01 Ethicon Endo-Surgery Generator G11 Vulnerability, November 28, 2017

Alerts and advisories released since FY 2010. FY and CY 2016 based on closed tickets.

# Lack of third party management

## Suppliers / editors are still too often in a strong position

- Remote maintenance often a requirement (non secured…)
- Unguaranteed products in case of security patch installation or even antivirus

## They often provide equipment in their default / non hardened configuration

## The introduction of malicious components is eased by the important volume of employees' in and out and the multiplication of suppliers



# Lack of security supervision

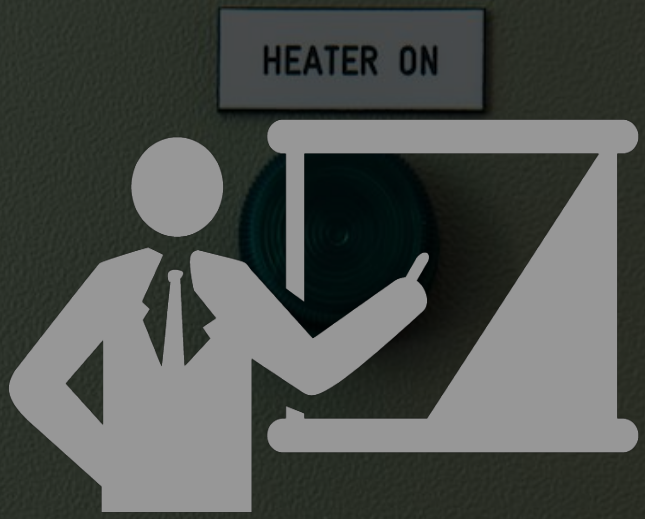


Supervision is at the heart of SCADA systems, it is even their primary purpose

However, security supervision is almost non-existent

Equipment do not have event logging or incident notification protocols

Because of interconnection requirements, a security operation center cannot be easily put together



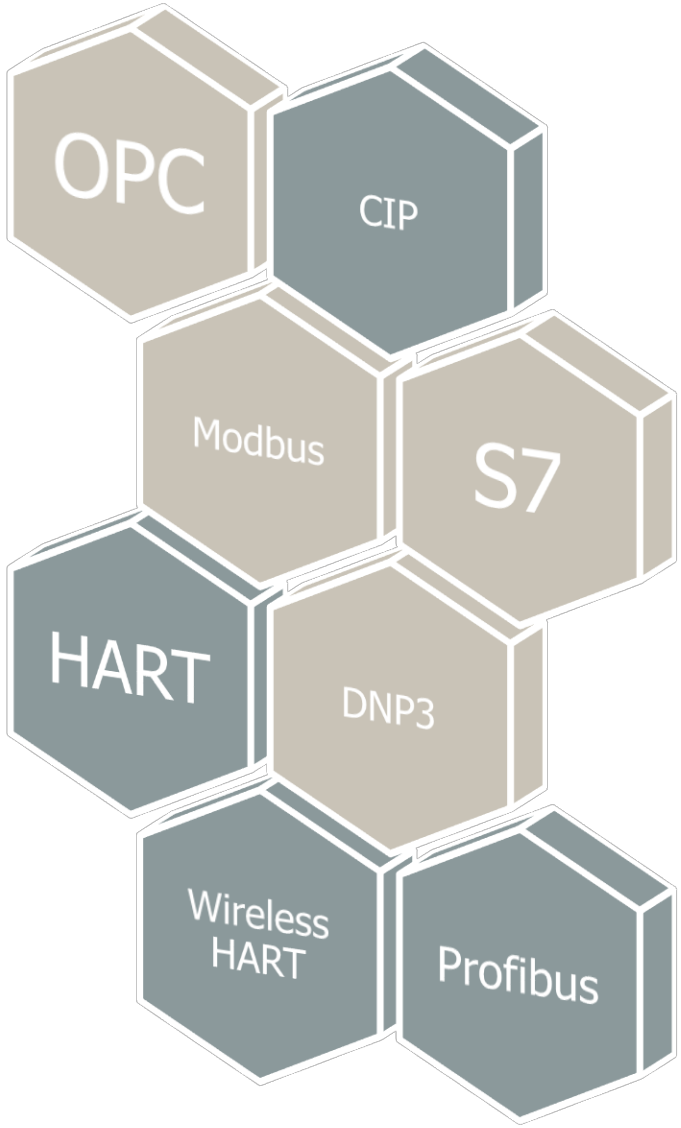

# ICS Protocols

# Security in protocols

ICS devices often use proprietary protocols, and there are also several standards

That is why on a given plant/factory ICS, you are likely to find several protocols in use

We will cover the most used ones



# Modbus protocol

- Serial communication protocol invented in 1979 by Schneider Electric
- Developed for industrial application
- Royalty-free
- Now one of the standards for industrial communications

## How it works:

- Master / Slave protocol
- Master must regularly poll the slaves to get information
- Modbus addresses are 8 bits long, so only 247 slaves per master
- There is no object description: a request returns a value, without any context or unit

## Security anyone?

- Clear-text
- No authentication



# Modbus protocol

- Modbus was originally made for serial communications
- However it is now often used over TCP (port 502)

## Modbus TCP/IP frame

- Transaction identifier set by the sender
- Protocol identifier set to 0 (default Modbus value)

| Transaction identifier | Protocol identifier | Length field | Slave address | Funtion code | Data |
|---|---|---|---|---|---|
| | | | | | Variable structure depending on the function |
| 2 bytes | 2 bytes | 2 bytes | 1 byte | 1 byte | N bytes |

# Modbus protocol

## Modbus functions

- The most common Modbus functions allow to read and write data from/to a PLC
- Other functions, such as file read and diagnostics functions also exist
- Undocumented Modbus function codes can also be used to perform specific actions

## COMMONLY USED MODBUS function codes

| Function name | Function code |
|---|---|
| Read coils | 1 |
| Write single coil | 5 |
| Read holding registers | 3 |
| Write single register | 6 |
| Write multiple registers | 16 |
| Read/Write multiple registers | 23 |

# Modbus protocol



| Function type | | | Function name | Function code |
|---|---|---|---|---|
| Data Access | Bit access | Physical Discrete Inputs | **Read Discrete Inputs** | 2 |
| | | Internal Bits or Physical Coils | **Read Coils** | 1 |
| | | | **Write Single Coil** | 5 |
| | | | Write Multiple Coils | 15 |
| | 16-bit access | Physical Input Registers | **Read Input Registers** | 4 |
| | | Internal Registers or Physical Output Registers | **Read Holding Registers** | 3 |
| | | | **Write Single Register** | 6 |
| | | | Write Multiple Registers | 16 |
| | | | Read/Write Multiple Registers | 23 |
| | | | Mask Write Register | 22 |
| | | | Read FIFO Queue | 24 |
| | File Record Access | | Read File Record | 20 |
| | | | Write File Record | 21 |
| Diagnostics | | | Read Exception Status | 7 |
| | | | Diagnostic | 8 |
| | | | Get Com Event Counter | 11 |
| | | | Get Com Event Log | 12 |
| | | | Report Slave ID | 17 |
| | | | Read Device Identification | 43 |
| Other | | | Encapsulated Interface Transport | 43 |

# S7 protocol

- Proprietary protocol by Siemens
- No security
- New version of the protocol available starting with version 4 and up of the PLC firmware : provides mutual authentication and communication encryption
- Security features analyzed by Quarkslab, see the talk from SSTIC https://www.sstic.org/2015/presentation/analyse_de_scurite_de_technologies_propritaires_scada

| | Couche OSI | Protocole |
|---|---|---|
| 7 | Couche application | S7 communication |
| 6 | Couche présentation | S7 communication |
| 5 | Couche session | S7 communication |
| 4 | Couche transport | ISO-on-TCP |
| 3 | Couche réseau | IP |
| 2 | Couche liaison | Ethernet |
| 1 | Couche physique | Ethernet |

- 3 steps to establish a connection with a Siemens PLC:
  - Connect to the PLC via TCP on **port 102**
  - Connect to ISO layer (COTP Connect Request)
  - Connect to the S7comm layer
- S7comm protocol rely on the following protocols:
  - **COTP** : Connection-Oriented Transport Protocol
  - **TPKT** : "ISO transport services on top of the TCP"
  - **TCP** : TPKT use TCP as transport protocol

# OPC protocol in general

- Standard protocol
- Used to exchange data between ICS and Windows devices
- Works on TCP/IP
- Several variants:
  - OPC-DA : Data access, used to gather data from the process control
  - OPC A&E : Alarm & Events
  - OPC HDA : Historical Data Access
  - OPC DX : Data Exchange, allow to exchange data between OPC servers
  - OPC Security
  - OPC XML-DA
  - OPC UA : Unified Architecture, aimed at replacing the others while using a more modern Service Oriented Architecture.
- Provides authentication and encryption, probably the future of ICS protocols

# OPC-UA

- Defined in IEC 62541 in 2015
- Designed to replace « DCOM »
- Open and non-hardware specific protocol
- Probably the future of ICS communications

## How it works

- Service-oriented architecture (client/server)
- A client can read and edit server nodes, as well as subscribe to them. It is then notified by the server when the node is modified.
- Thanks to the nodes hierarchy and names, it is possible to know what is controlled by the node.
- One server can handle several clients simultaneously.
- The protocol can use « binary/TCP » or « SOAP/HTTP »

## Security

- Several security levels: none, signature, signature and encryption.
- Compatible with X.509 certificates and Kerberos.
- Login/password connection
- Fine grained access rights for each node (read/write).

# DNP3 protocol



- Standard protocol, developed by GE in the 90's
- Slave/Slave: Information can be share at the initiative of any device
- Layer 2 protocol (just top of physical layer), ported to TCP/IP
- Integrity is verified using CRC
- Data reporting communication: only send the data that has changed (simplified), or at the initiative of the PLC/RTU
- Used for smart grids in the US
- Secure DNP3 introduced in 2007
  - Works on serial and TCP versions
  - Challenge/response to exchange a session key (using PSK)
  - Can be done at startup, every XX minutes, or only for sensitive actions (write requests for example)
  - There is also an aggressive mode, which can be exploited to perform replay attacks
  - Possible to use secure DNP3 over TLS

# Profibus protocol

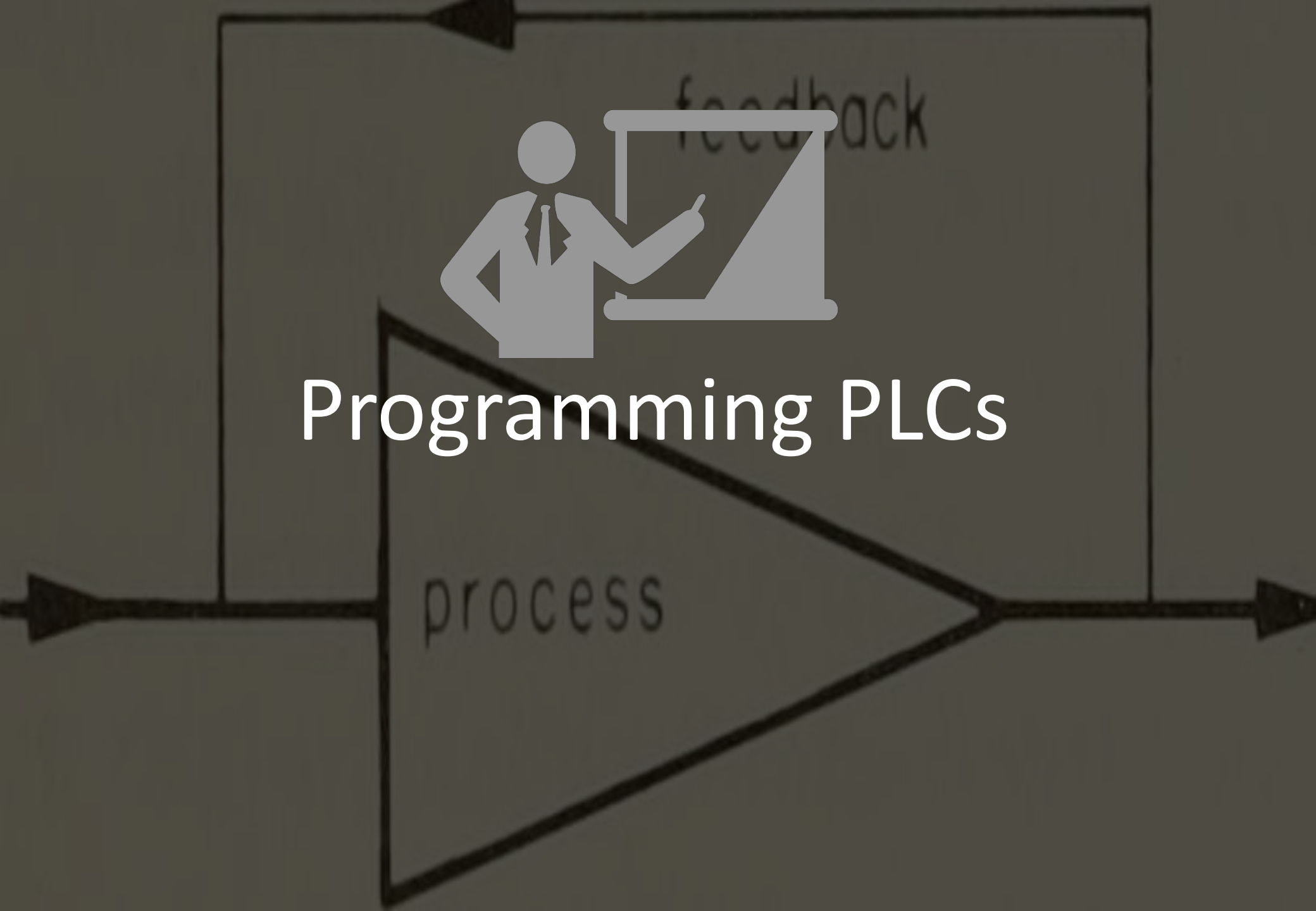


- Standard protocol
- Used for communication with field devices (sensors/actuators)
- Uses 4-20mA current loop

# #Foreverdays

#foreverdays is a term coined by @reverseics
Very important concept when talking about ICS
The highest vulnerabilities are not patched.

So it is really worth considering the effort of patch management of ICS equipment when you know

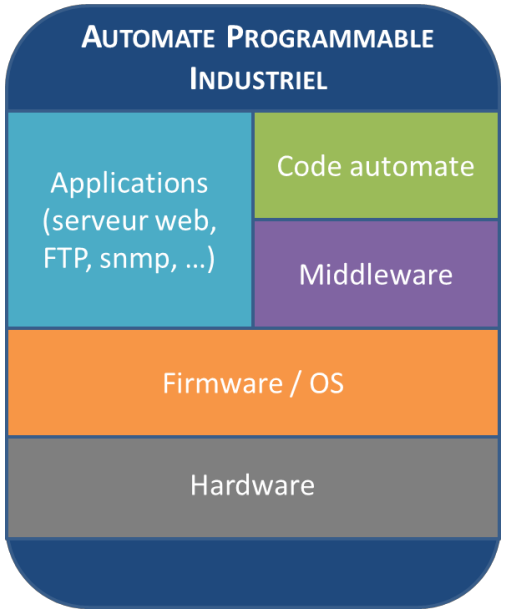



# Programming PLCs

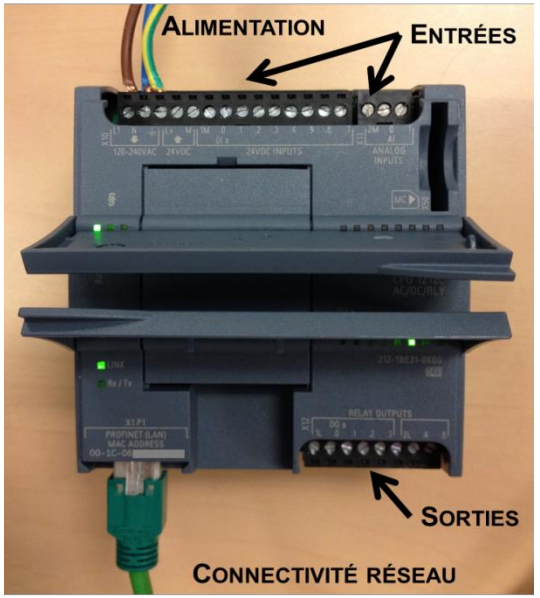
# What is a PLC?

- Real-time digital computer used for automation
- Replaces electrical relays
- Lots of analogue or digital inputs & outputs
- Rugged devices (immune to vibration, electrical noise, temperature, dust, …)

## What's inside?




## Siemens S7-1200



# A few pics of PLCs







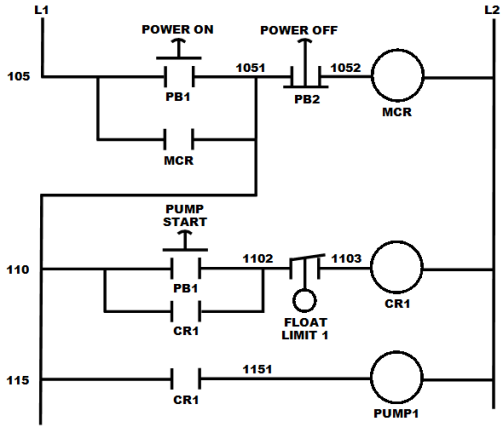

# PLC programming

"Ladder Logic" was the first programming language for PLC, as it mimics the real-life circuits

IEC 61131-3 defines 5 programming languages for PLCs

- **LD:** Ladder Diagram
- **FBD:** Function Block Diagram
- **ST:** Structured Text
- **IL:** Instruction List
- **SFC:** Sequential Function Chart

## Ladder diagram example




## Structured text example

```
(* simple state machine *)
TxtState := STATES[StateMachine];

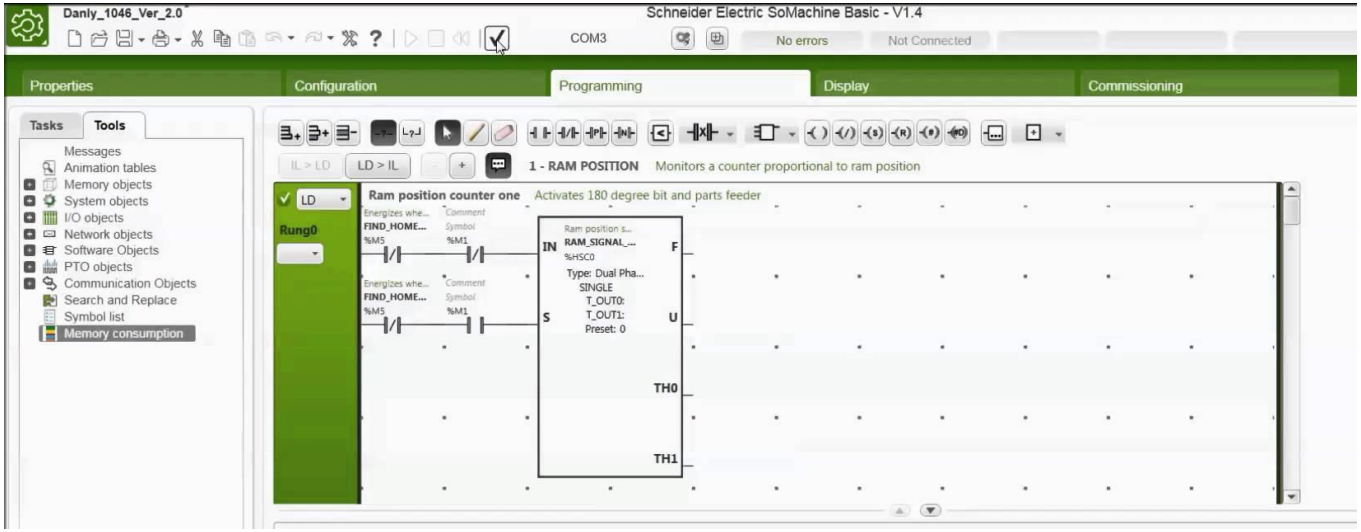
CASE StateMachine OF
   1: ClosingValve();
ELSE
    ;; BadCase();
END_CASE;
```

## Instruction list example

```
LD      Speed
          GT      1000
          JMPCN   VOLTS_OK
          LD      Volts
VOLTS_OK  LD      1
          ST      %Q75
```

# Programming with SoMachineBasic

- SoMachineBasic is the software provided by Schneider Electric to program the entry-level PLCs.
- PLCs used in big plants are usually programmed using Unity Pro, for which there is no free demo version.
- Fortunately, the way this software work is very much the same

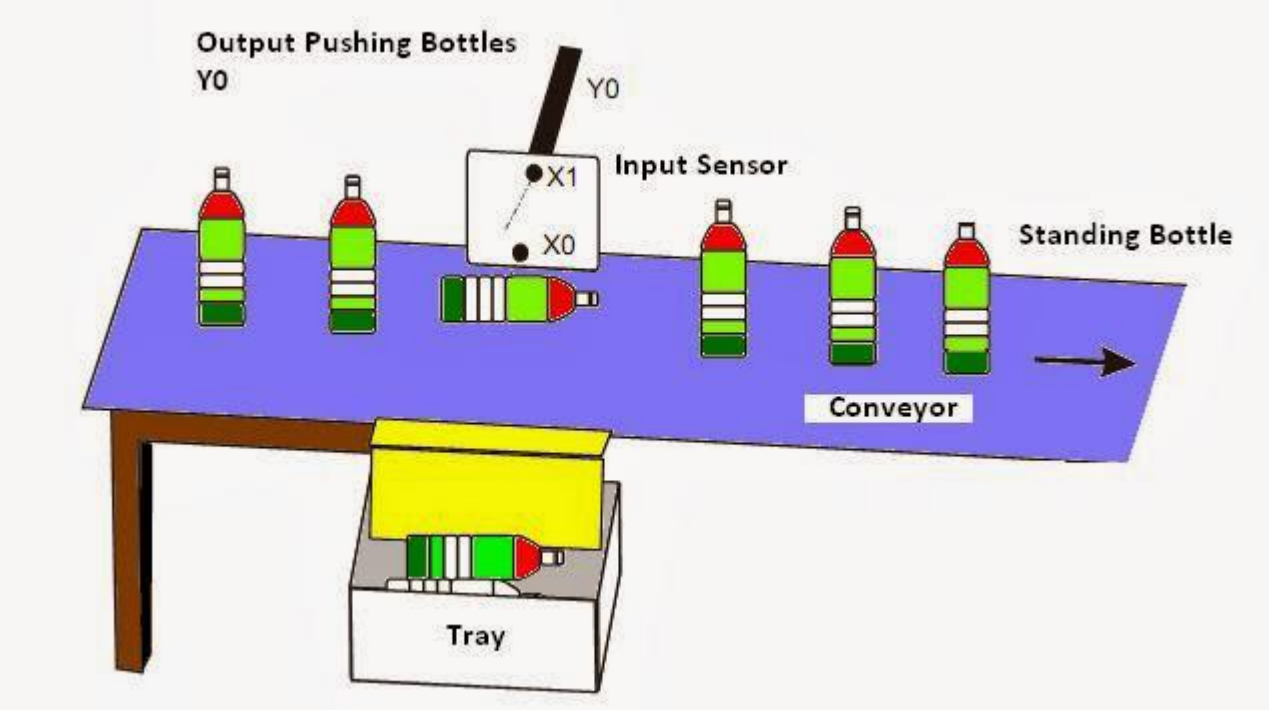



**PLC programming**

- Create a project
- Define the hardware setup
- Create variables
- Define the program
- Test
- Debug
- Push to PLC
- START

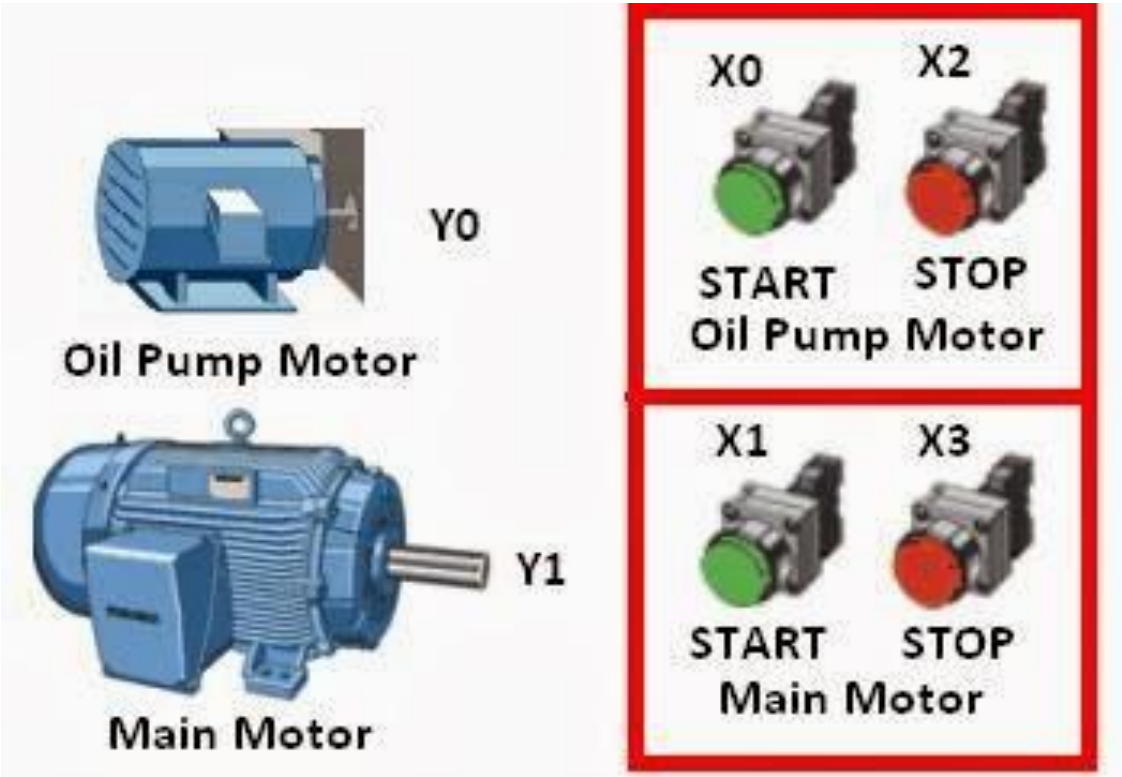
# PLC programming

- Production line
- Flipped-over bottles must be put in the tray



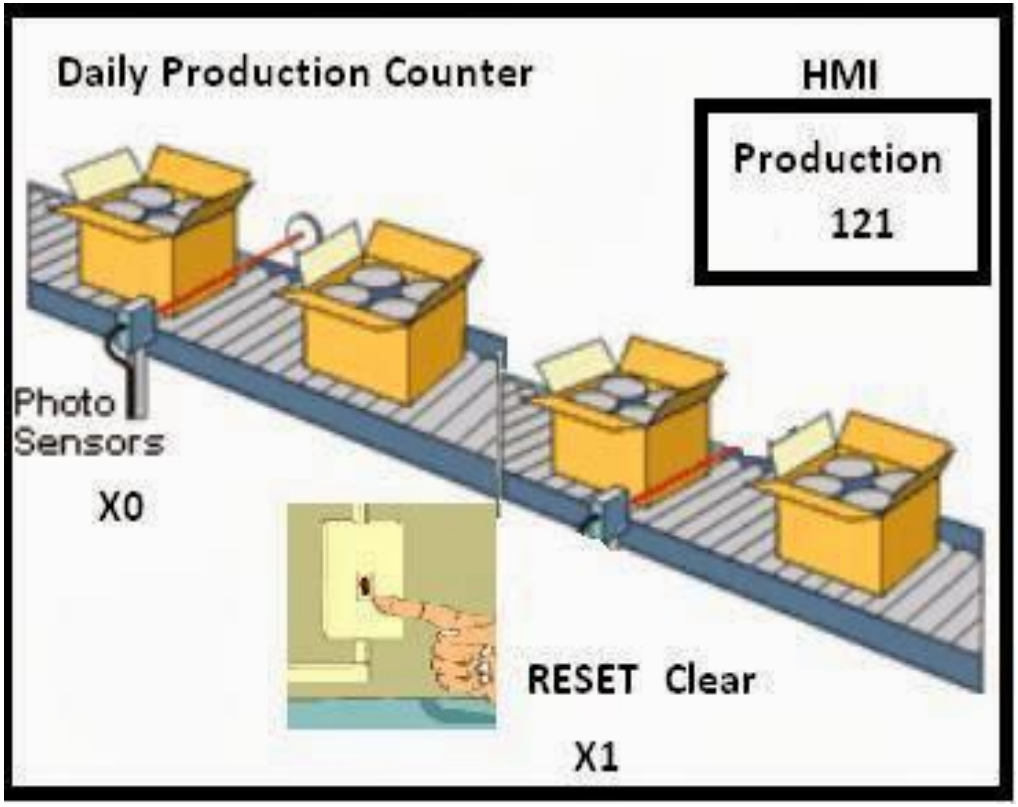
# PLC programming

- The main motor must only start if the oil pump is running
- Motor must stop is X3 is pressed



# PLC programming

- Another production line
- Display indicates when 10 packages
- A button allows resetting the display

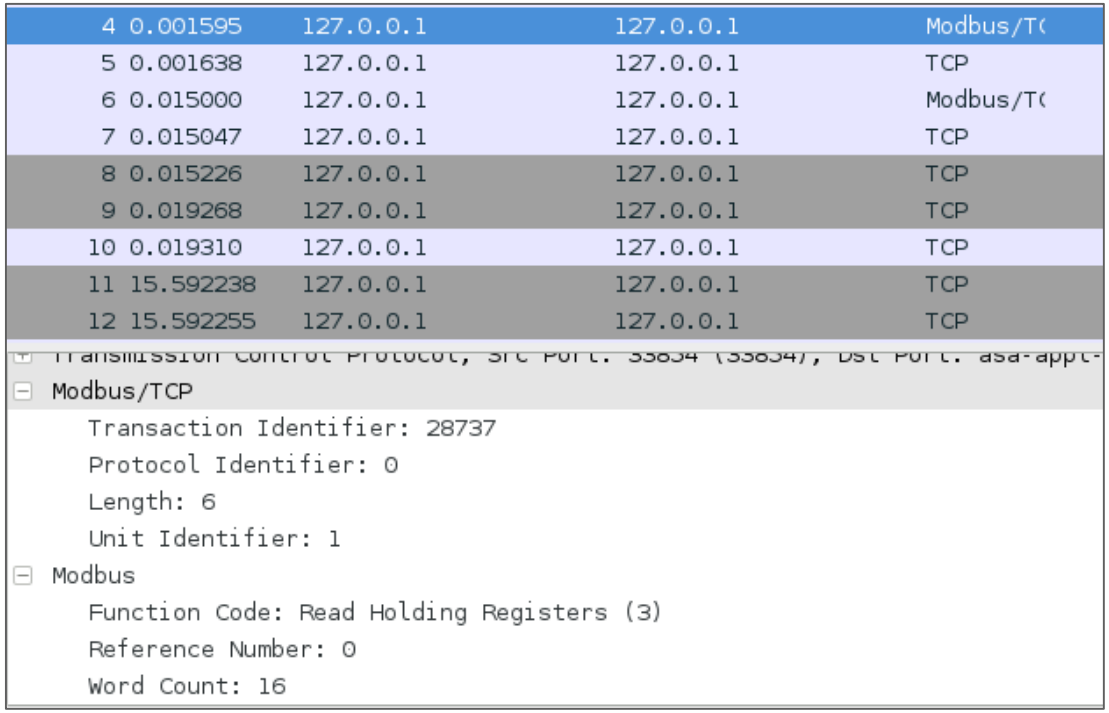




# PENTESTING PLCs

# Lab Session #1: Analyzing a Modbus communication with Wireshark

- Analyze a modbus communication with Wireshark
- Wireshark owns by default a modbus dissector



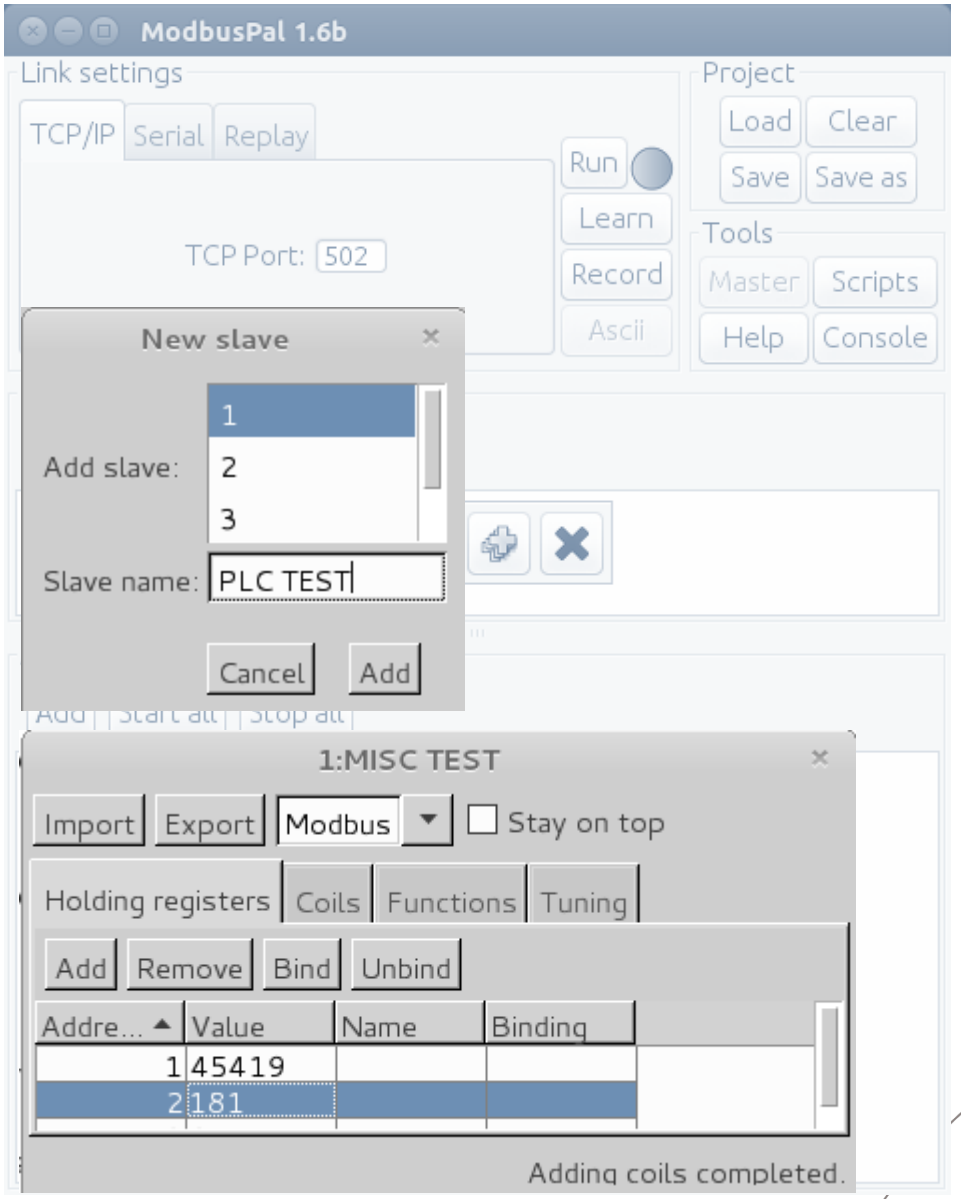

- Launch Wireshark
- Open « modbus1.pcap »
- Try to understand what's going on
  - Reading request
  - Writing request
  - PLC's answer
- What's the value of register #123 at the end?

# Lab session #2: ModbusPal

- Modbuspal is a modbus simulator

```
$ > cd /root/toolz/modbus
$ > java -jar ModbusPal.jar
```

- Add a modbus slade
- Set some register values
- Query it with:
  - MBTGET Perl script
  - Metasploit module
- Analyze traffic with Wireshark



# Lab session #2: ModbusPal + MBTGET




- Mbtget is a perl script to perform Modbus/tcp queries

```
$ > cd root/toolz/modbus/mbtget/scripts
$ > ./mbtget -h
```

- Read requests
  - Coils (1 bit)

```
$ > ./mbtget -r1 -a 0 -n 8 127.0.0.1
```

  - Words (8 bits)

```
$ > ./mbtget -r3 -a 0 -n 8 127.0.0.1
```

- Write requests
  - Coils (1 bit)

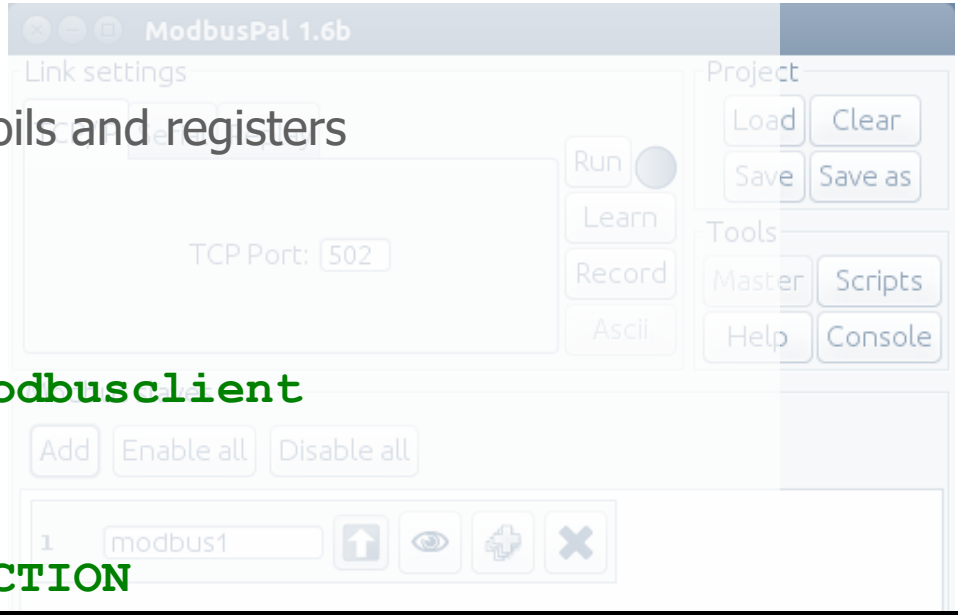
```
$ > ./mbtget -w5 #{VALUE} -a 0 127.0.0.1
```

  - Words (8 bits)

```
$ > ./mbtget -w6 #{VALUE} -a 0 127.0.0.1
```

# Lab session #2: ModbusPal + Metasploit

- A simple Modbus client
- Can perform read and write operations on coils and registers




- Launch msfconsole

```
$ > msfconsole
msf > use auxiliary/scanner/scada/modbusclient
msf auxiliary(modbusclient) > info
```

- Play!

```
msf auxiliary(modbusclient) > set ACTION
```

```
msf auxiliary(modbusclient) > set RHOST 127.0.0.1
RHOST => 127.0.0.1
msf auxiliary(modbusclient) > set DATA_ADDRESS 0
DATA_ADDRESS => 0
msf auxiliary(modbusclient) > run

[*] Sending READ REGISTER...
[+] Register value at address 0 : 123
[*] Auxiliary module execution completed
msf auxiliary(modbusclient) >
```

```
msf auxiliary(modbusclient) > set ACTION WRITE_REGISTER
ACTION => WRITE_REGISTER
msf auxiliary(modbusclient) > set DATA 321
DATA => 321
msf auxiliary(modbusclient) > run

[*] Sending WRITE REGISTER...
[+] Value 321 successfully written at registry address 0
[*] Auxiliary module execution completed
msf auxiliary(modbusclient) > set ACTION READ_REGISTER
ACTION => READ_REGISTER
msf auxiliary(modbusclient) > run

[*] Sending READ REGISTER...
[+] Register value at address 0 : 321
[*] Auxiliary module execution completed
msf auxiliary(modbusclient) >
```

# Lab session #3 : S7 using Snap7

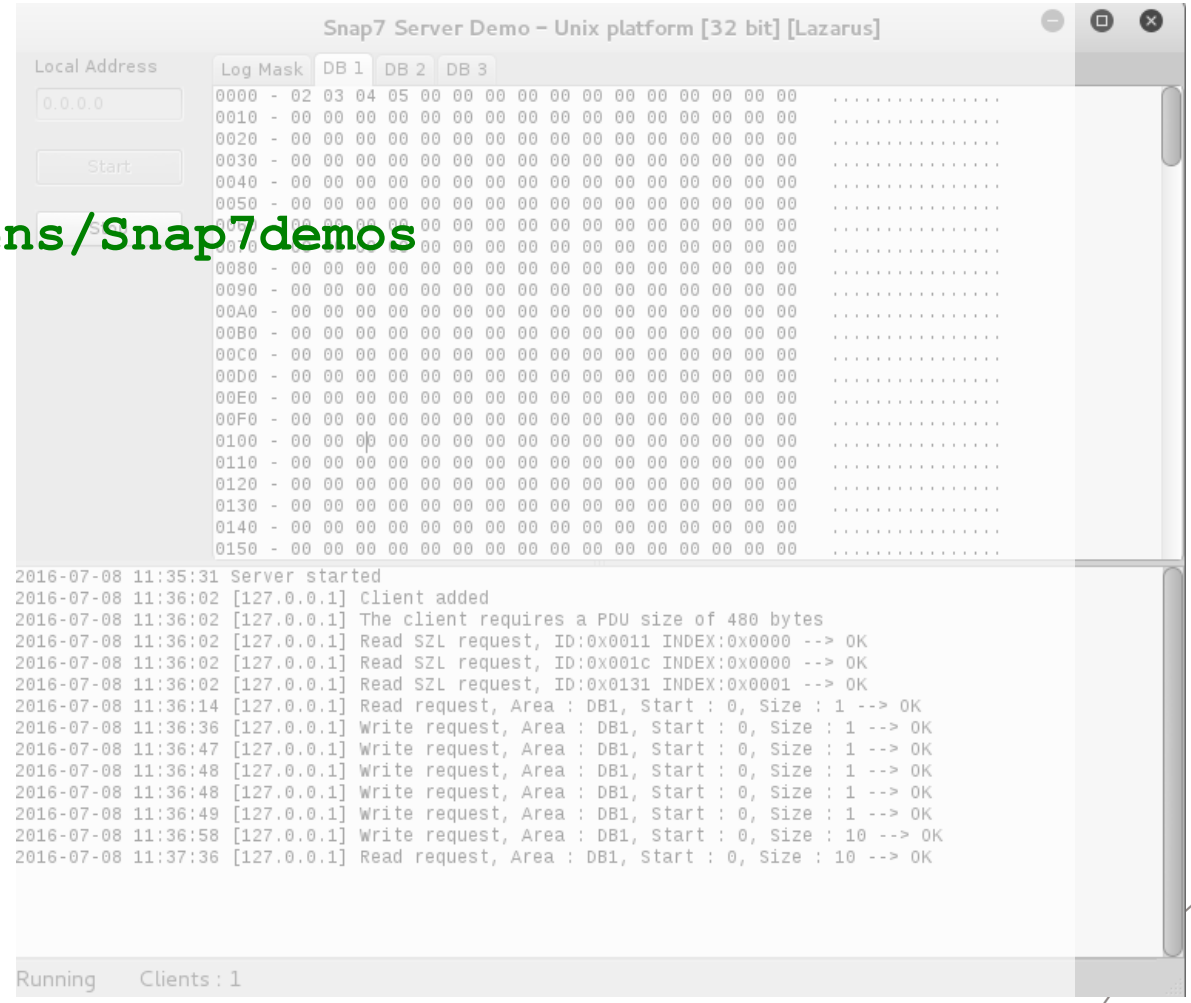
- Snap7 is an open-source library implementing Siemens S7 protocol
- It is pretty complete for "old" PLCs, but all functionalities do not work with more recent PLCs (S7-1200, S7-1500)

- Launch the demo server

```
$ > cd toolz/siemens/Snap7demos
$ > ./serverdemo
```

- Query it using the demo client

```
$ > ./clientdemo
```



# Lab session #3 : S7 using Snap7

- You can also use the scripts that rely on the Snap7 python wrappers

```
$ > cd toolz/siemens/
$ > python S7get.py
$ > python S7getDB.py
```

- Read request

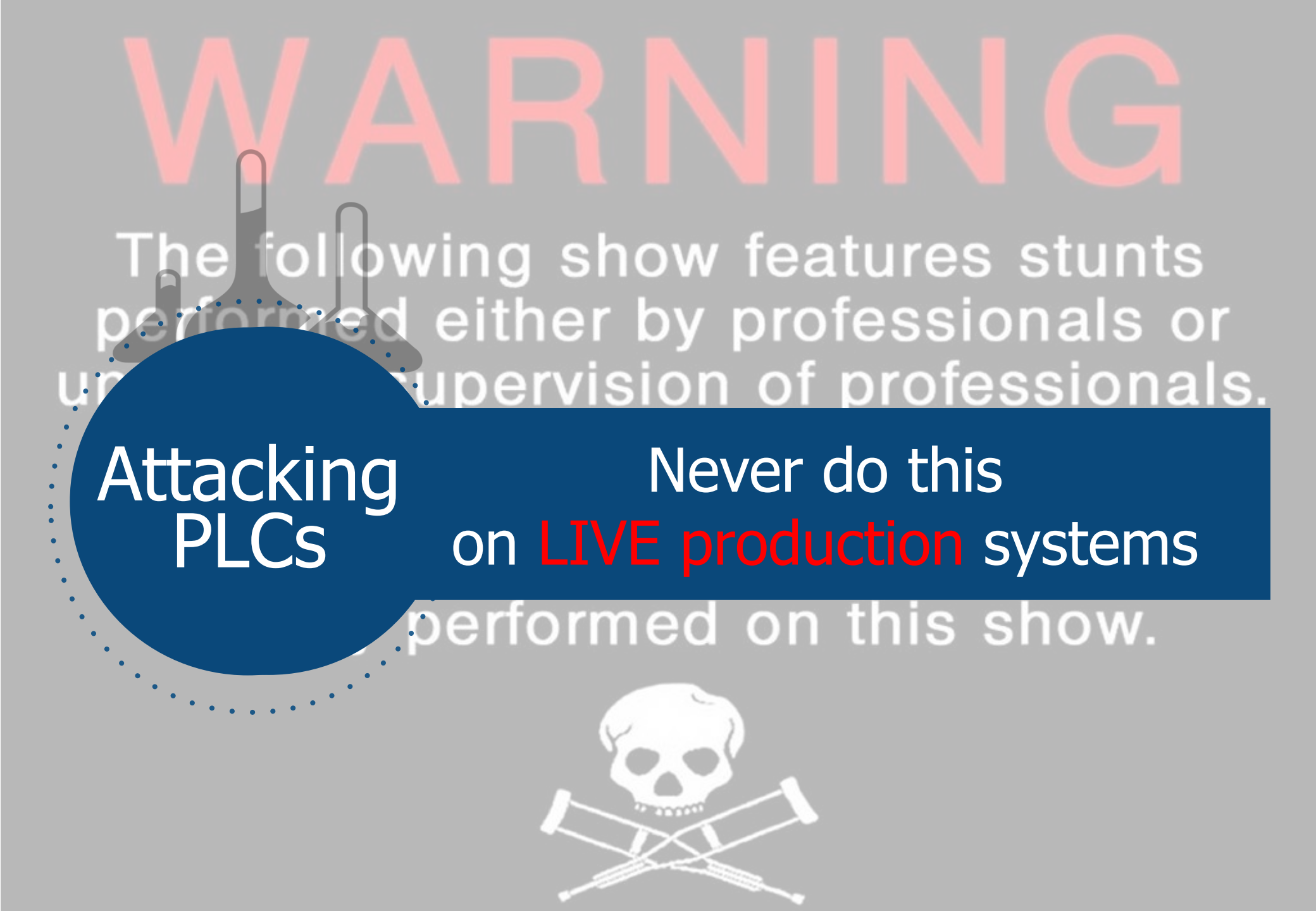
```
python s7get.py -a 0 -m r -n 8 XX.XX.XX.XX
```

```
root@kali:~/toolz/siemens# ./s7getDB.py -a 0 -db 1 -m r -n 8 192.168.0.50
===Outputs===
Output 0 : 1
Output 1 : 0
Output 2 : 0
Output 3 : 0
Output 4 : 0
Output 5 : 0
Output 6 : 0
Output 7 : 0
```

- Write request

```
python s7get.py -a 0 -m w -d 1 XX.XX.XX.XX
```

```
root@kali:~/toolz/siemens# ./s7getDB.py -a 0 -db 1 -m w -d 0 -n 1 192.168.0.50
[+] 0 has been written correctly from address 0
```

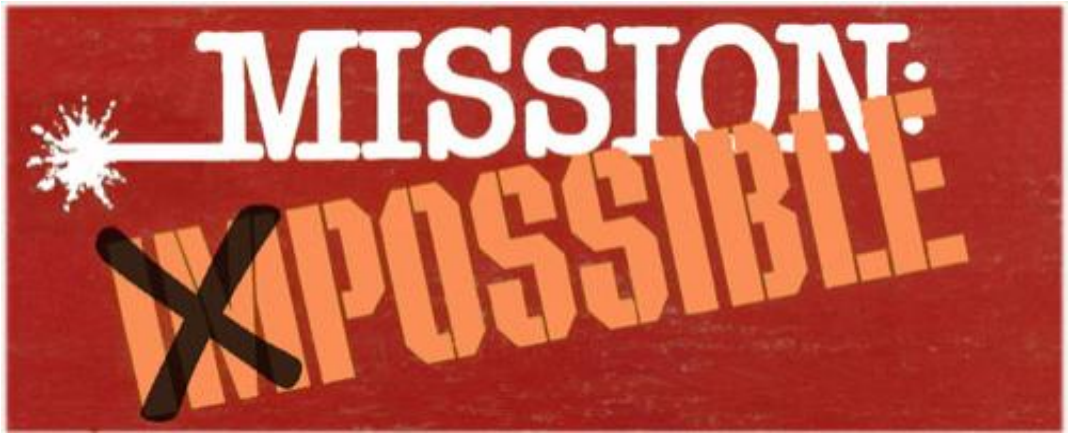

WARNING
The following show features stunts
performed either by professionals or
supervision of professionals.
performed on this show.
Attacking PLCs
Never do this
on LIVE production systems

# Capture the flag

Your mission, should you choose to accept it, is to stop the train and capture the flag with the robot arm.

Hacking ICS ?

It's ….




No crazy « hanging from the ceiling without sweating » stuff required !



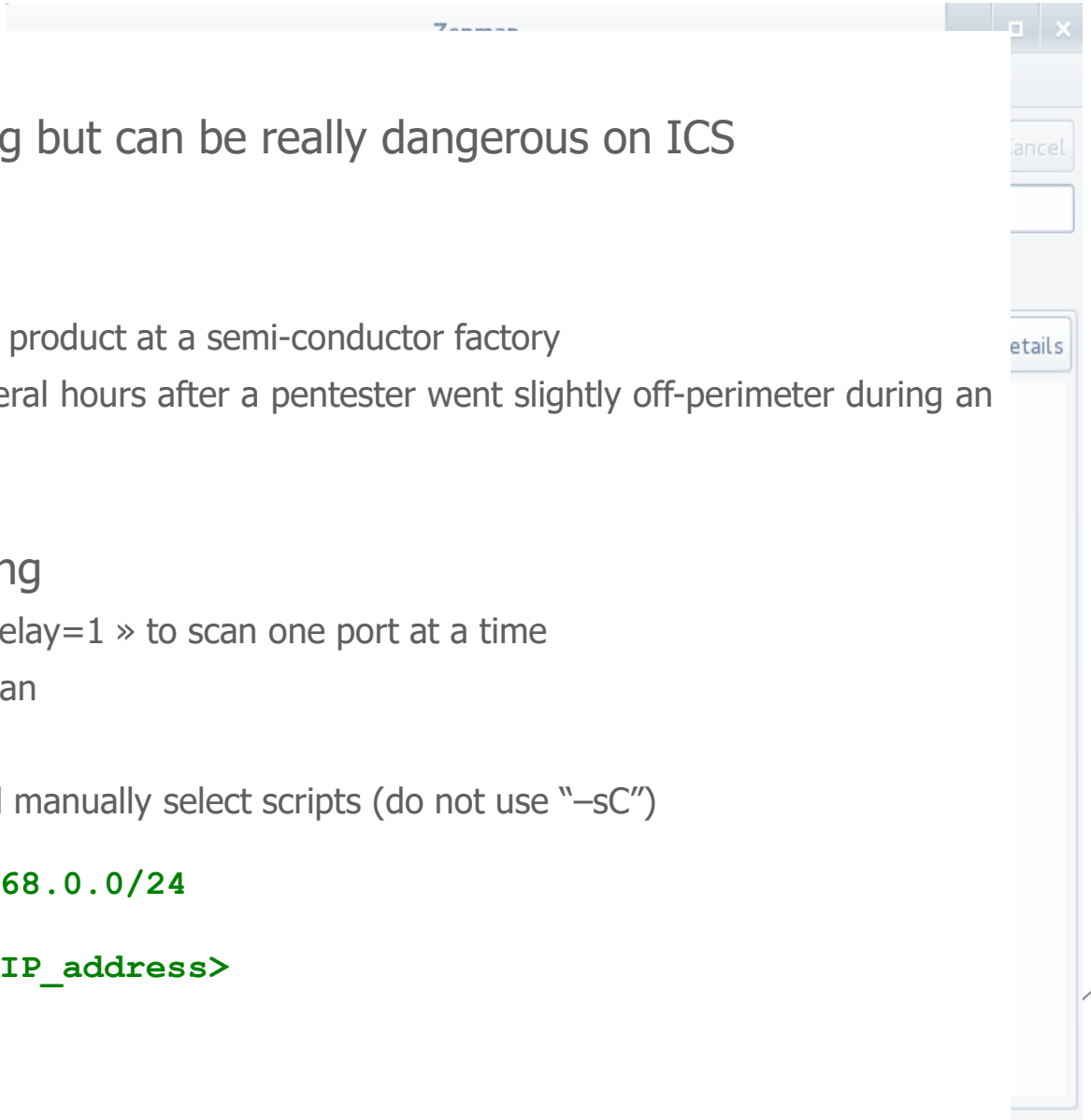
# Reconnaissance

- Objective : Identify all exposed services on a device or a range of devices
- Often the first step in a pentest

- We will use two tools
  - **Nmap**: The world's finest port scanner
  - **PLCSCAN**: A reconnaissance tool dedicated to PLCs

- Network information
  - Wifi SSID: "ICS_101" (pass : "yoloscada")
  - DHCP to obtain an address (192.168.0.100 and up)
  - Targets are between 192.168.0.0 and 192.168.0.75

# Reconnaissance (Nmap)



- The de-facto tool for port scanning but can be really dangerous on ICS
- Two stories from NIST SP800-82
  - A ping sweep broke for over 50 000$ in product at a semi-conductor factory
  - The blocking of gas distribution for several hours after a pentester went slightly off-perimeter during an assessment for a gas company
- Nmap useful setup for ICS scanning
  - Reduce scanning speed! Use « --scan-delay=1 » to scan one port at a time
  - Perform a TCP scan instead of a SYN scan
  - Do not perform UDP scan
  - Do not use fingerprinting functions, and manually select scripts (do not use "–sC")

```
$ nmap -sT --scan-delay=1 192.168.0.0/24

$ nmap -p- -sT --scan-delay=1 <IP_address>
```

# Reconnaissance (PLCSCAN)




- https://code.google.com/p/plcscan/ by SCADAStrangeLove (http://scadastrangelove.org/)
- Scans for ports **102** (Siemens) and **502** (Modbus) and tries to pull information about the PLC (modules, firmware version,…)
- Not exhaustive since not all PLCs use Modbus or are Siemens
- What if I told you there was another way… SNMP ?

```
$ python plcscan.py <IP_address>
```

# Attacking standard services

- Most PLCs have standard interfaces, such as **HTTP** and **FTP**
- Lets' say security was not the first thing in mind when introducing these features …




- On Schneider M340
  - FTP credentials are hardcoded

    sysdiag / factorycast@schneider
  - Allows you to retrieve the password file for the web UI



Copyright © 1999-2009, Schneider Automation SAS. All Rights Reserved.

# Lab session #4 : Attacking PLCs
# Unauthenticated administrative actions on Schneider PLCs



Gather information on PLC and project

```
msf > use auxiliary/admin/scada/schneider.rb
```

function GATHER_INFOS



Start / Stop PLC

```
msf > use auxiliary/admin/scada/modicon_command
```



Download PLC project

```
msf > use auxiliary/admin/scada/modicon_stux_transfer_ASO
```



Modify PLC project

```
msf > use auxiliary/admin/scada/modicon_stux_transfer_ASO
```



Force PLC digital outputs

```
msf > use auxiliary/admin/scada/schneider.rb
```

function M340_FORCE_OUTPUTS




Hotpatch the PLC

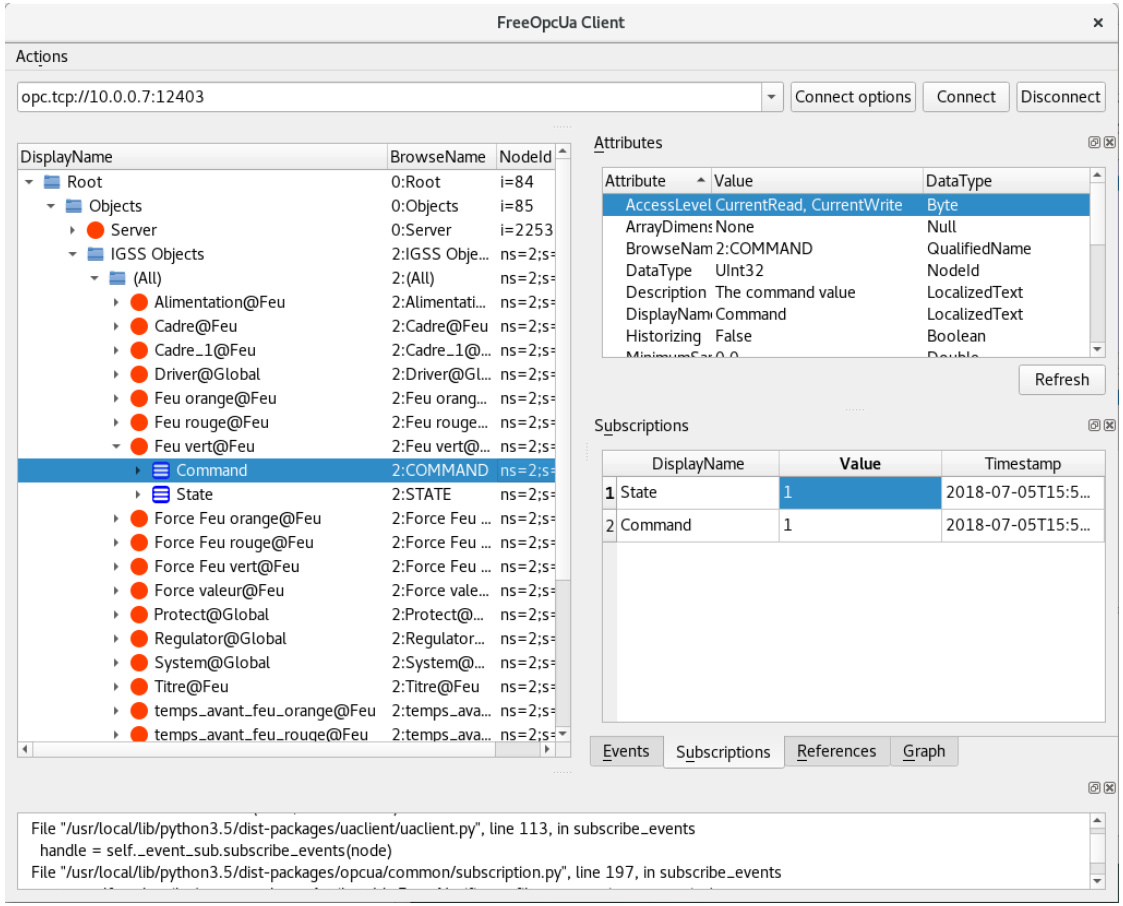
Hotpatch

# Lab session #5: OPC UA with FreeOpcUa (opcua-client)

- opcua-client is a OPC-UA client written in Python

```
$ > opcua-client
```

- Connect to a server
- Modify nodes value
- Subscribe to nodes
- Analyze the traffic with Wireshark



# Lab session #5: OPC UA with Automap_OPCUA

- Automap_OPCUA is based on « FreeOpcUa » in Python

```
$ > cd /toolz/opcua
$ > ./automap_opcua.py
```

- Verify the OPC UA server security level

```
$ > ./automap_opcua.py -t ip -p port -sc mask
```

- Browse the nodes hierarchy

```
$ > ./automap_opcua.py -t ip -p port -ti
```

- Monitor nodes value

```
$ > ./automap_opcua.py -t ip -p port -s 'node' -l -c 10
```

- Modify nodes value

```
$ > ./automap_opcua.py -t ip -p port -s 'node' -l -d
id=value
```



# Securing ICS

# ICS are vulnerable

## What's wrong with ICS security?

- Inexistent network segmentation
- Lack of « Patch management »
- Lack of security mechanism in equipment and protocols
- Lack of security supervision
- Lack of third party management
- Lack of awareness

## These vulnerabilities are gateways used to attack the information system

Appropriate organisational and technical security measures are necessary



# Securing ICS



## Network segmentation

- Limit your ICS exposure
  - Nothing on the Internet
  - Expose only necessary services on your internal network
- Implement filtering
  - Use DMZ / Data diodes to export data from ICS to corporate network
  - Use DPI (Deep Packet Analysis) technology

## Patch when you can

- Patching once a year during plant maintenance is better than doing nothing

## Apply corporate best practices

- Harden your system
  - Install antivirus
  - Disable unused services
- Manage user accounts
  - Change default passwords
  - Use nominative accounts
  - Harden password policies
- Secure administration

## Security supervision

- IPS have signatures for ICS
- Create your own signatures, it is not that difficult
  - Detect configuration change
  - Detect new equipment

# What about securing PLCs themselves?

**PLC hardening?**

- Not possible on older models as far as we know

**Use newer PLCs?**

- Only the latest high-end Schneider PLCs implement security
- Have them pentested

**Network and security monitoring?**

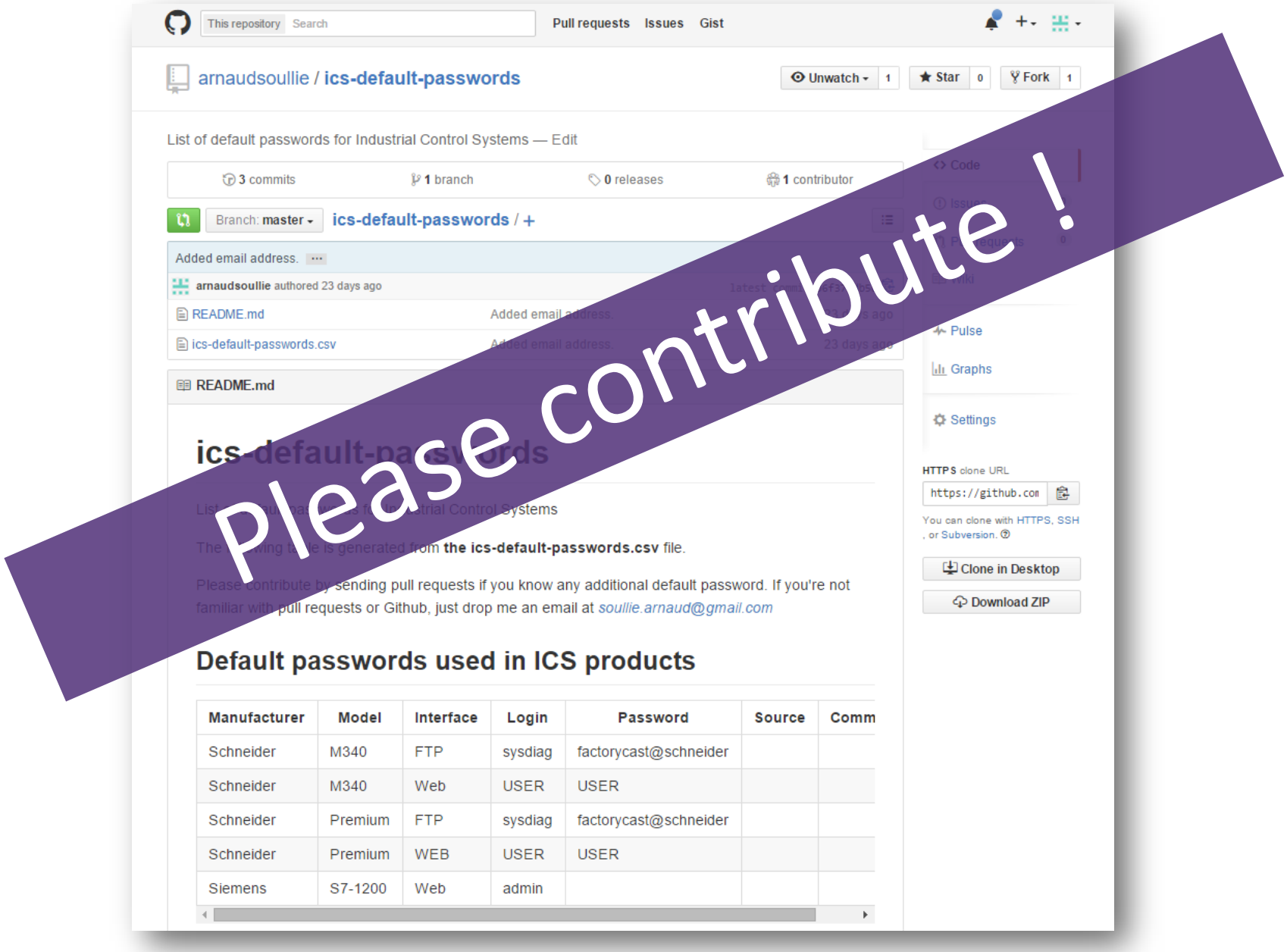
- There are signatures for Modbus function 90, use them



# Do you have any question?

That's all Folks!

Thanks for attending




https://github.com/arnaudsoullie/ics-default-passwords

WAVESTONE

**Arnaud SOULLIE**
Manager

**M** +33 (0)7 62 86 83 71
Arnaud.soullie@wavestone.com

**Alexandrine TORRENTS**
Consultant

**M** +33 (0)7 63 26 79 87
Alexandrine.torrents@wavestone.com

wavestone.com
@wavestone_

PARIS

LONDON

NEW YORK

HONG KONG

SINGAPORE *

DUBAI *

SAO PAULO *

LUXEMBOURG

MADRID *

MILANO *

BRUSSELS

GENEVA

CASABLANCA

ISTANBUL *

LYON

MARSEILLE

NANTES

* Partnerships

