

RSA®Conference2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN
ELEMENT

SESSION ID: LAB3-W02

Pentesting ICS 102



Arnaud SOULLIE

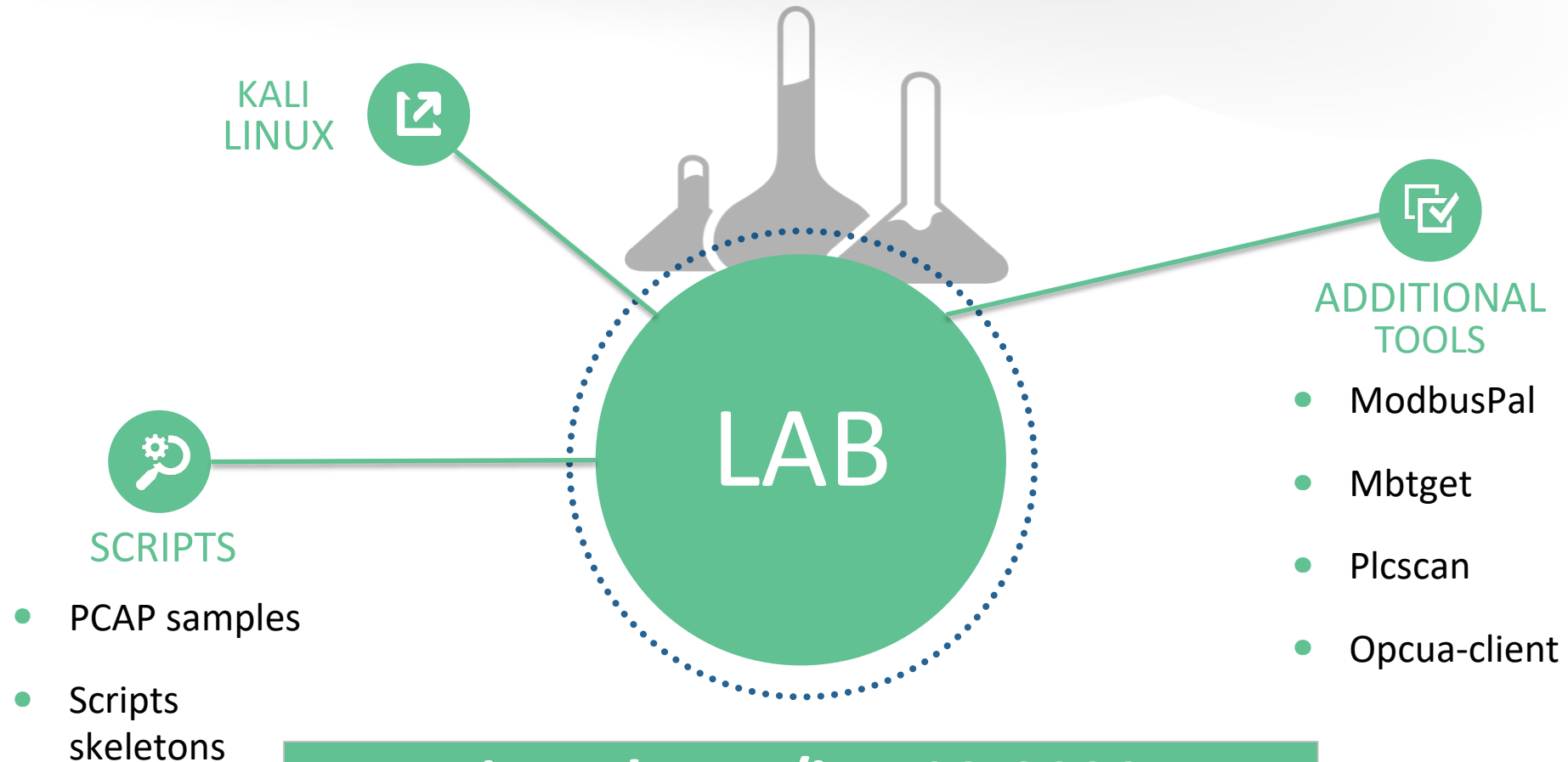
Manager
Wavestone
@arnaudsoullie

Alexandrine TORRENTS

Senior Consultant
Wavestone
@DrineTorrents

#RSAC

Lab Prerequisite



Agenda

- 01 Introduction to ICS
 - 02 What's wrong with ICS security?
 - 03 ICS protocols
 - 04 Capture the flag!
 - 05 Takeaways
-
- Hands-on !
- Hands-on !
- 30'
- 90'

RSA[®]Conference2020

Introduction to ICS



Where do we find Industrial 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.

A bit of vocabulary

ICS (Industrial Control System)

=

IACS (Industrial Automation and Control Systems)

≈

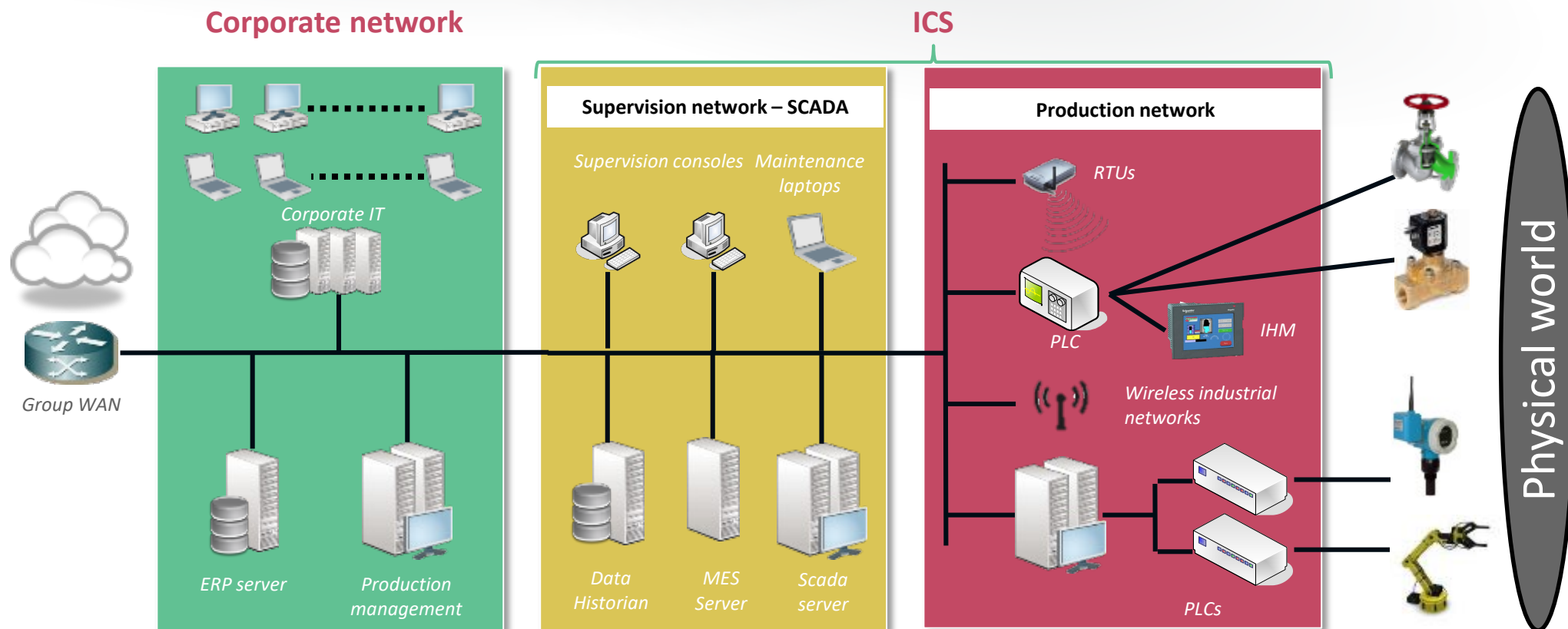
SCADA (Supervisory Control And Data Acquisition)

≈

DCS (Distributed Control System)

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

What is an Industrial Control System (ICS)?



Corporate IS handle data

≠

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

ICS Components

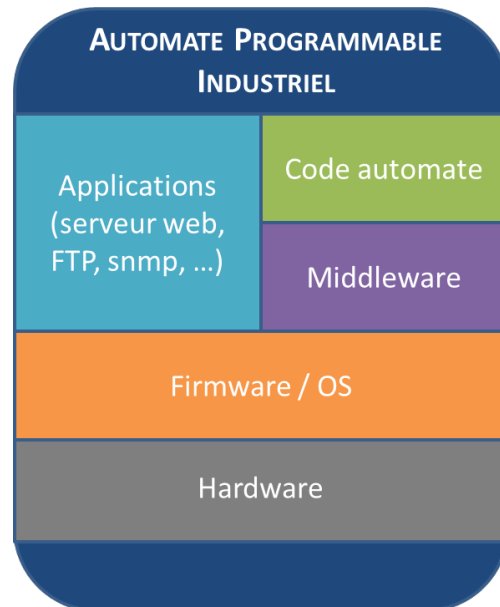
- **Sensors and actuators:** allow interaction with the physical world (pressure sensor, valves, motors, ...)
- **Local HMI:** Human-Machine Interface, permits the supervision and control of a subprocess
- **PLC (Programmable Logic Controller) :** manages the sensors and actuators
- **Supervision screen:** remote supervision of the industrial process
- **Data historian:** Records all the data from the production and Scada networks
- **MES:** Manufacturing execution system (production status, scheduling, etc.)
- **RTU:** Remote Terminal Unit (standalone PLC)
- **Other low level devices:** Intelligent electronic devices, wireless devices, variator frequency drives, remote I/O, etc

[illegible]

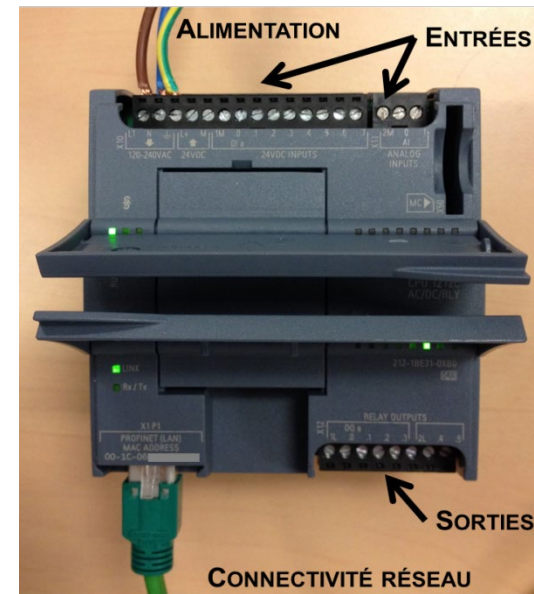
Focus on PLC

- 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?

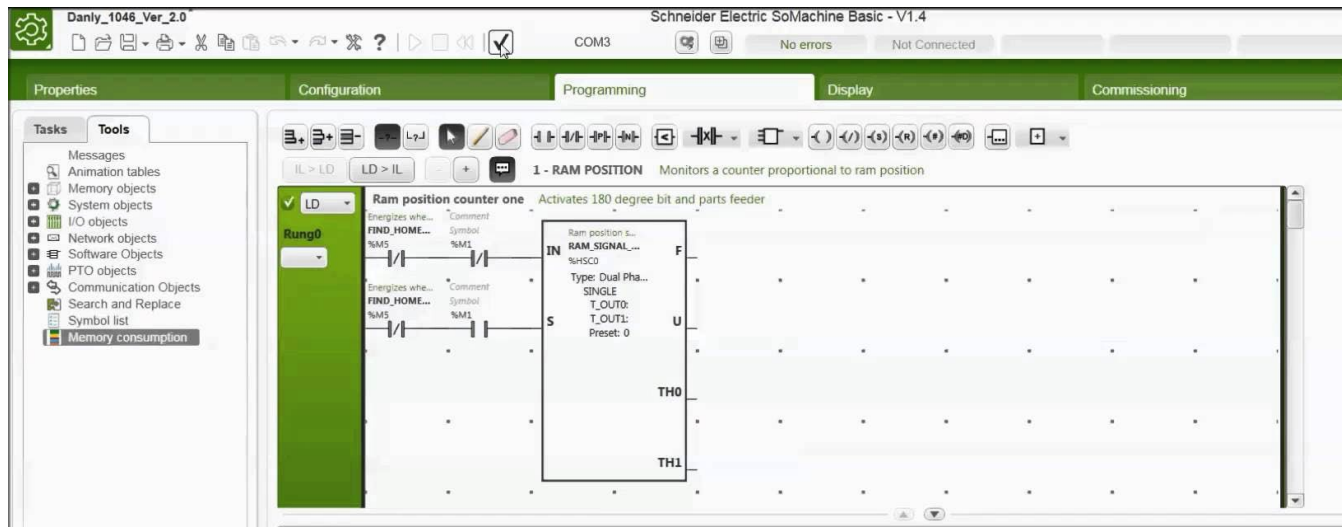


Siemens S7-1200



Focus on PLC programming

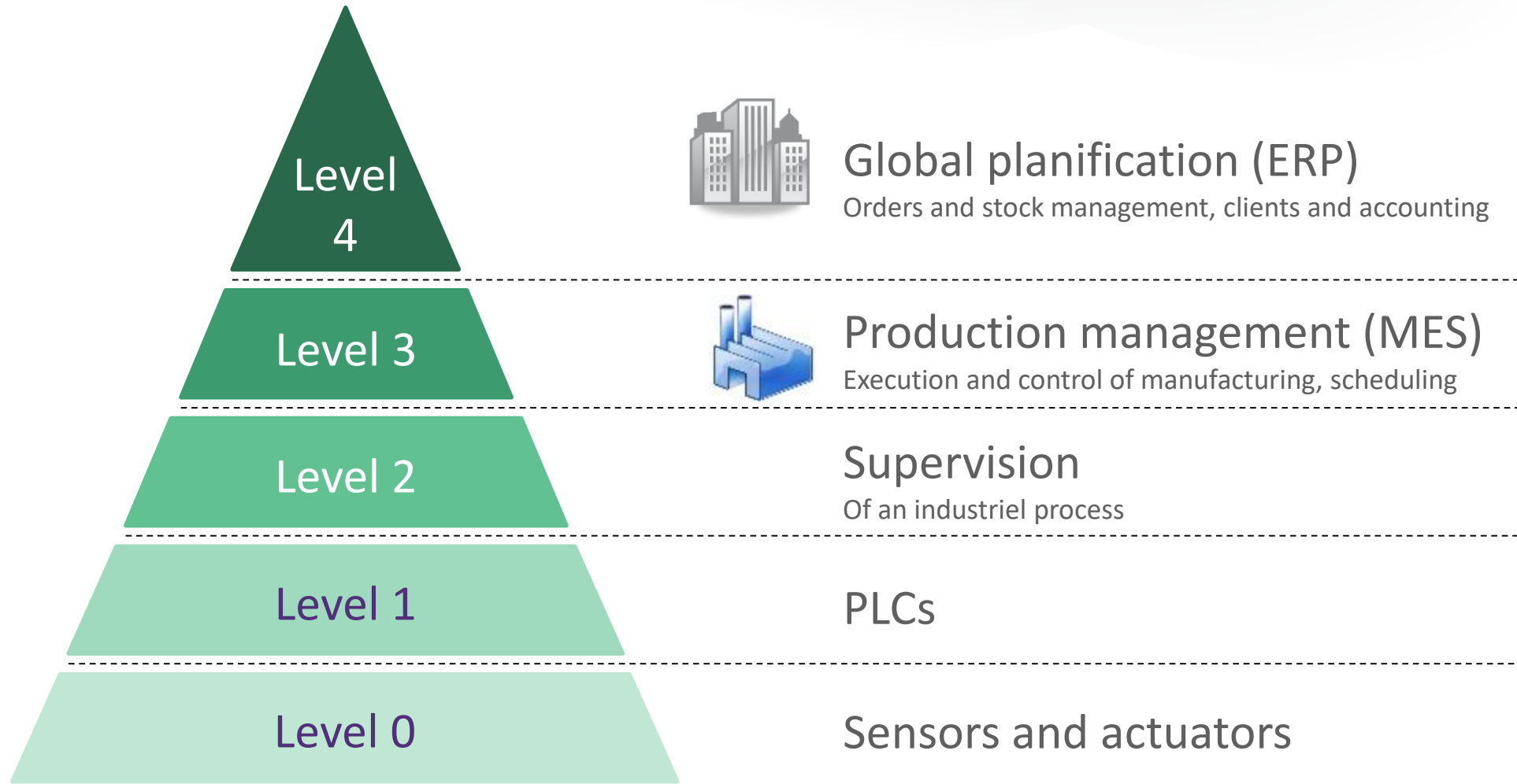
- SoMachineBasic is the software provided by Schneider Electric to program the entry-level PLCs.
- PLCs used in big plants are usually programmed using Unity Pro, for which there is no free demo version.
- Fortunately, the way this software work is very much the same



PLC programming

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

CIM (Computer Integrated Manufacturing) pyramid



Tired: IT vs OT

AKA: *Why OT security sucks compared to IT security*



Lifetime of components span **over decades**



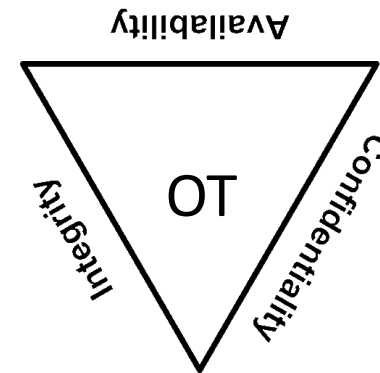
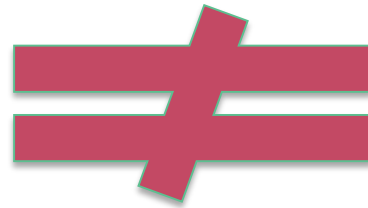
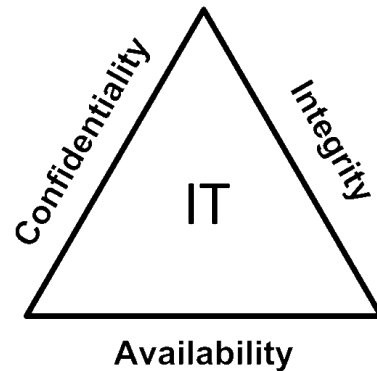
The essential criteria for ICS security is **availability**, not confidentiality



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**



Wired: OT vs IT

AKA: Leverage OT specificities to improve cybersecurity



Long lifetime means less change so it's easier to monitor for abnormal changes



Mostly no confidential data, so that's a thing less to worry about ;)



Strong culture of quality & change management



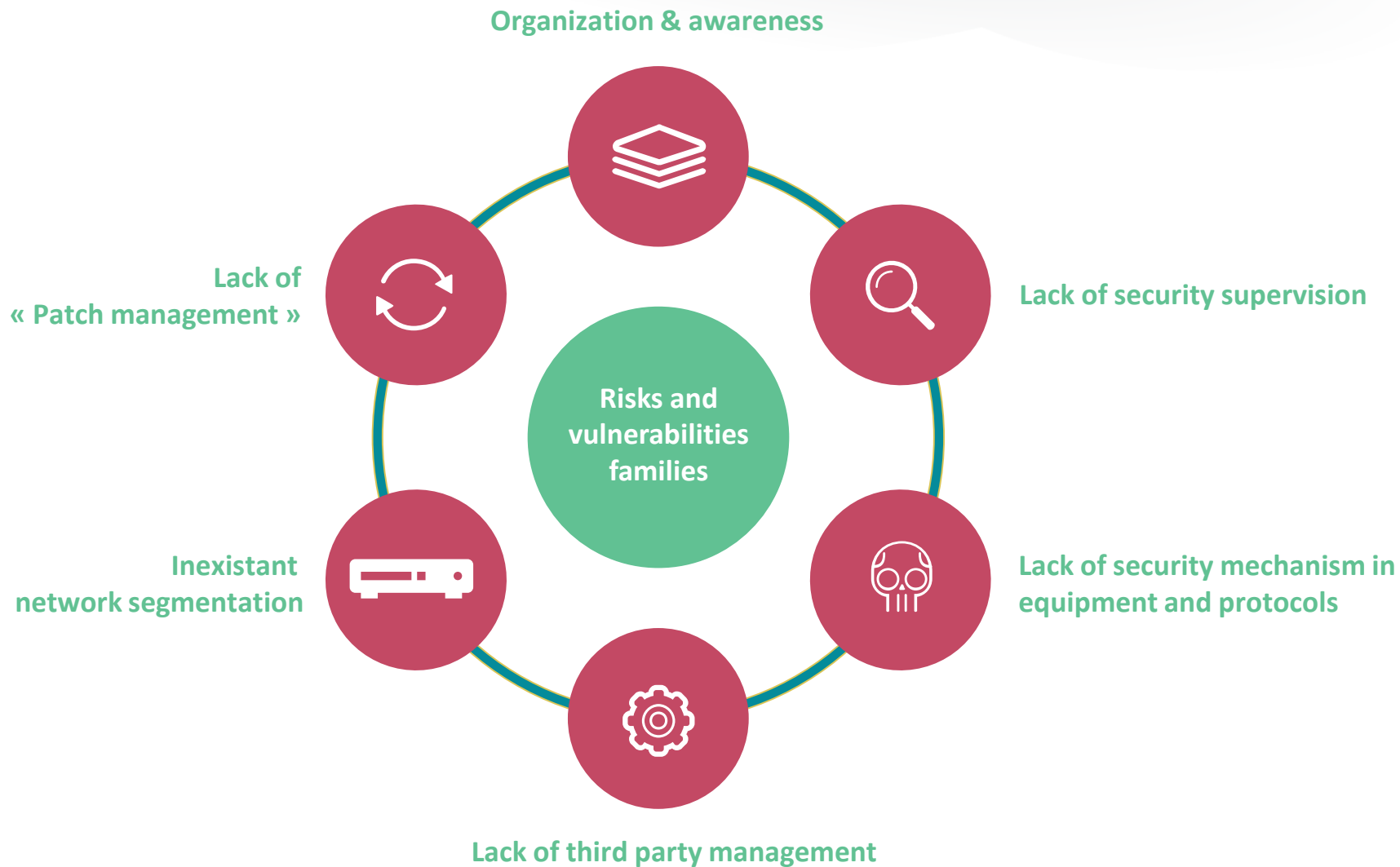
Safety is there to prevent all catastrophic events

ICS operations = Safety + Availability + Quality

RSA®Conference2020

What's wrong with ICS security?

What is wrong with current ICS security?



The slow evolution of ICS security

Most sites

Roles and responsibilities not formally defined ; lack of awareness

Lack of network filtering with the corporate network; unmanaged remote accesses

Incomplete cartography, security patches not applied and no obsolescence management

Protocols with no authentication and encryption ; no hardening

Multiple third parties with limited management

No supervision from a cybersecurity point of view



Mature sites

Organization & awareness

Creation of ICS cybersecurity sector with local relays

Network segmentation

Dedicated security equipment with firewall rules; remote accesses with strong authentication

Patch management

Full inventory, systems patched on a regular basis, plan to tackle obsolescence

Security in protocols

No change for protocols ; possibility to disable unused services


Third party management

Security requirements shared to third parties


Security supervision

Logs configured and centralized but no ICS specific detection scenarios yet

#Foreverdays



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



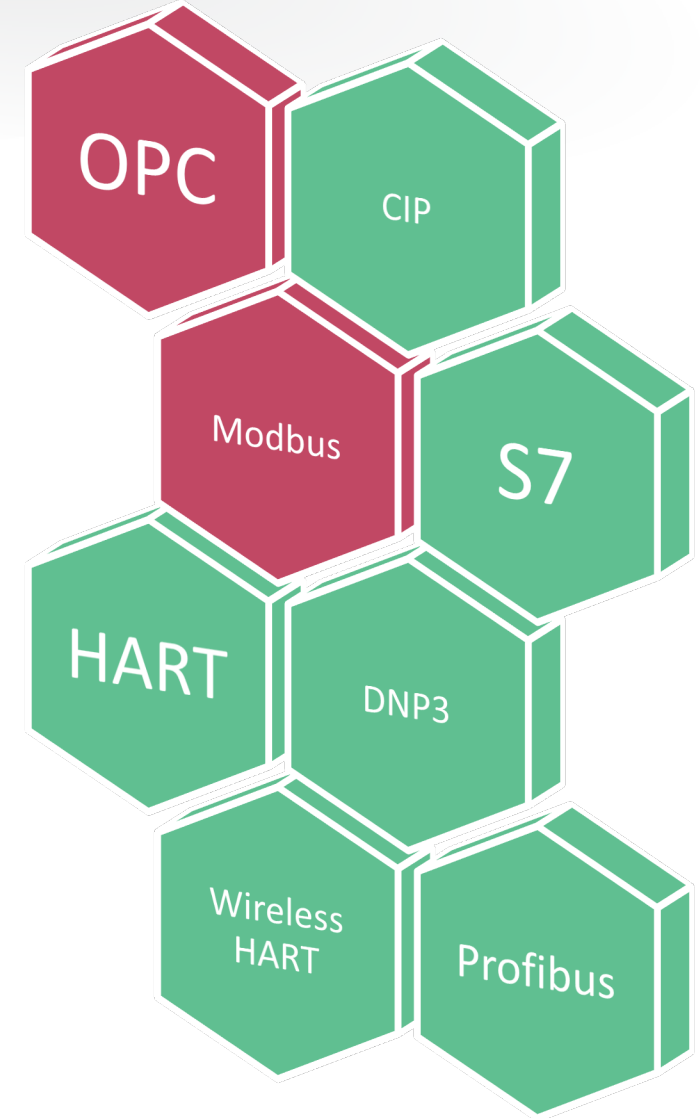
RSA®Conference2020

ICS Protocols

Security in protocols

At the beginning, specific protocols on specific **physical layer**
(RS232, RS485, 4-20 current loop)
Some protocols were **adapted to TCP/IP**, like Modbus,
and other were developed to allow interoperability.

ICS devices often use **specific protocols**, some of them are **proprietary**, and some of them are **common standards**
We will hereafter 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

- 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	Function 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

Comonly 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
			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	

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

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

4	0.001595	127.0.0.1	127.0.0.1	Modbus/TCP
5	0.001638	127.0.0.1	127.0.0.1	TCP
6	0.015000	127.0.0.1	127.0.0.1	Modbus/TCP
7	0.015047	127.0.0.1	127.0.0.1	TCP
8	0.015226	127.0.0.1	127.0.0.1	TCP
9	0.019268	127.0.0.1	127.0.0.1	TCP
10	0.019310	127.0.0.1	127.0.0.1	TCP
11	15.592238	127.0.0.1	127.0.0.1	TCP
12	15.592255	127.0.0.1	127.0.0.1	TCP

Transmission Control Protocol, Src Port: 33634 (33634), Dst Port: asa-appl-	
Modbus/TCP	
Transaction Identifier: 28737	
Protocol Identifier: 0	
Length: 6	
Unit Identifier: 1	
Modbus	
Function Code: Read Holding Registers (3)	
Reference Number: 0	
Word Count: 16	

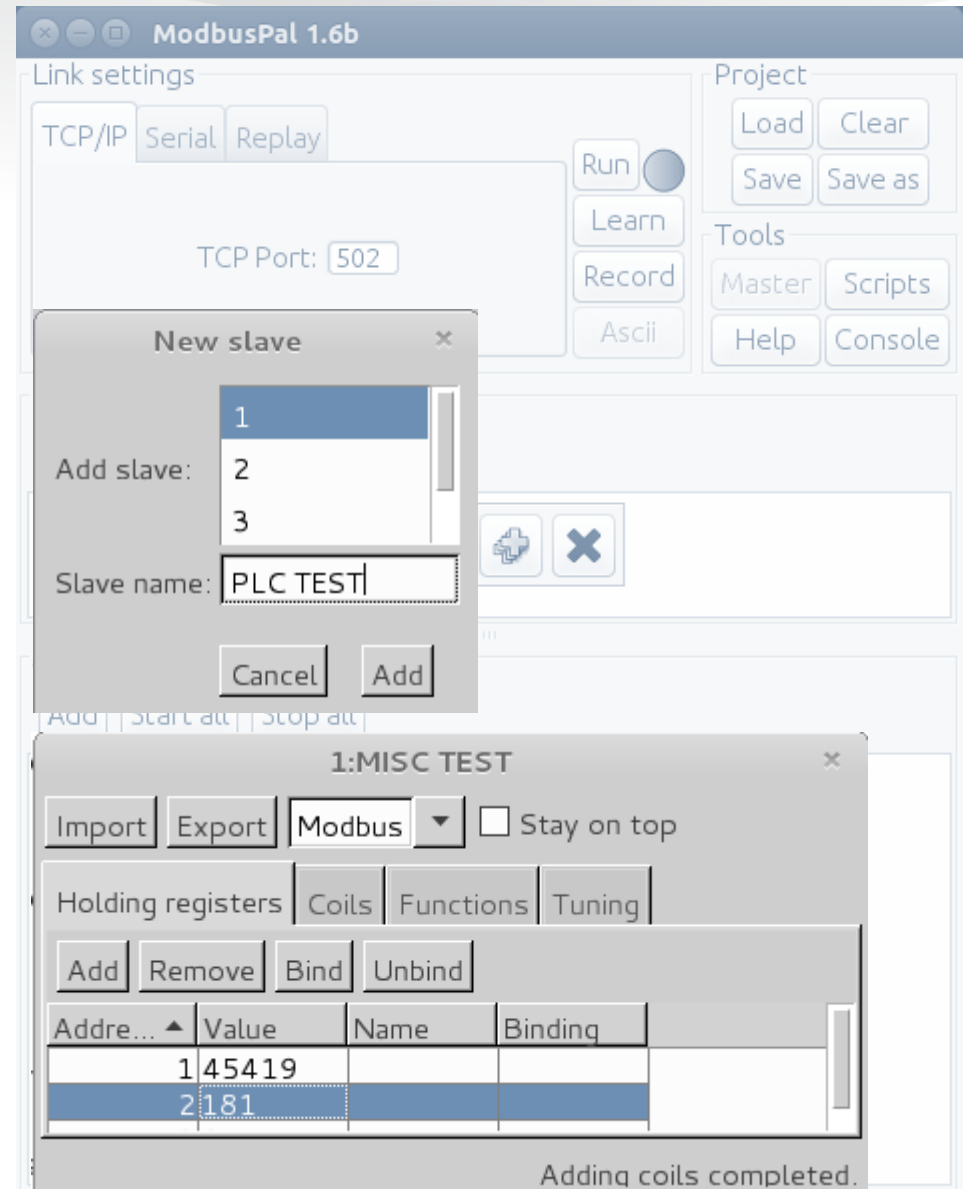
- Launch Wireshark
- Open « modbus1.pcap »
- Try to understand what's going on
 - Reading request
 - Writing request
 - PLC's answer
- What's the value of register #123 at the end?

Lab session #2a: ModbusPal

- ModbusPal is a Modbus simulator

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

- Add a Modbus slave
- Set some register values
- Query it with MBTGET Perl script
- Analyze traffic with Wireshark



Lab session #2a: 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) :

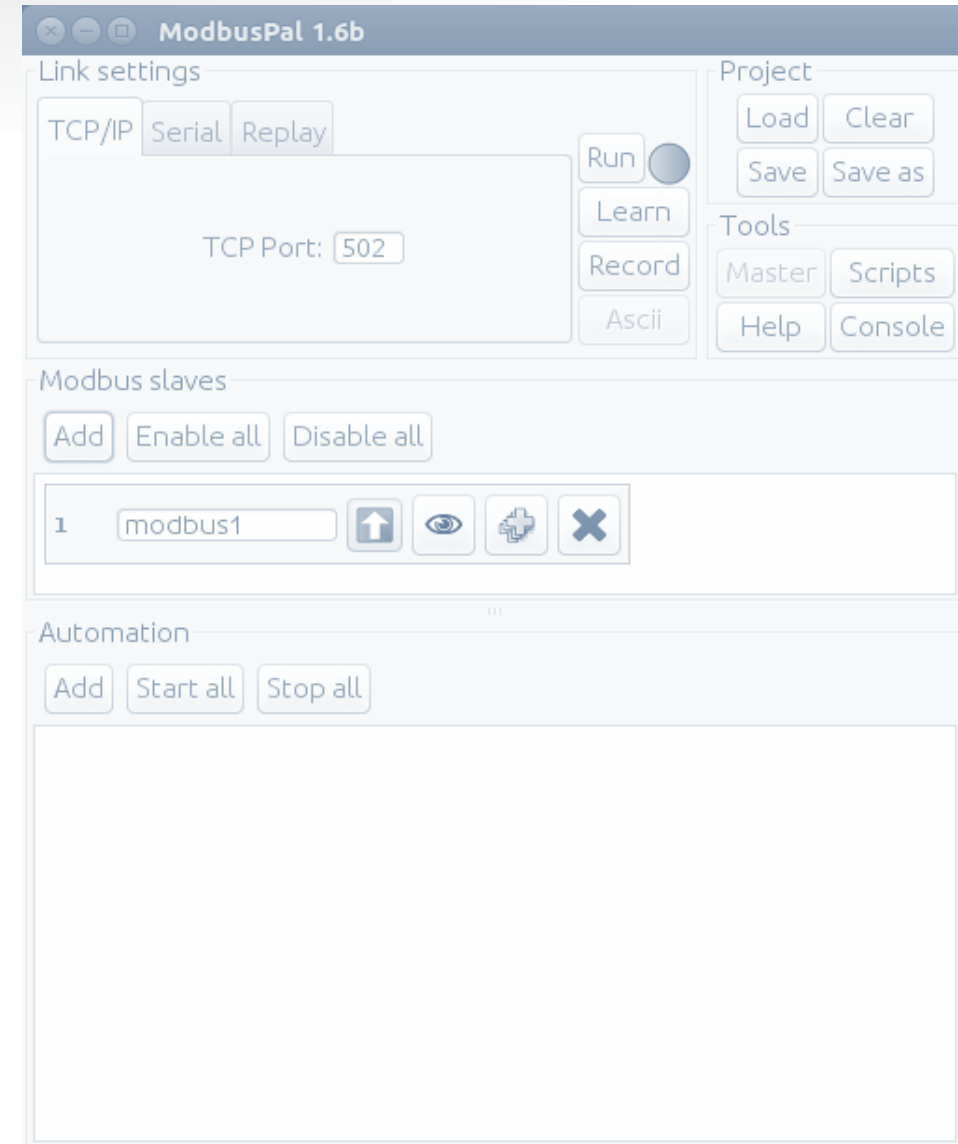
```
$ > ./mbtget -r3 -a 0 -n 8 127.0.0.1
```

- 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
```



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
-

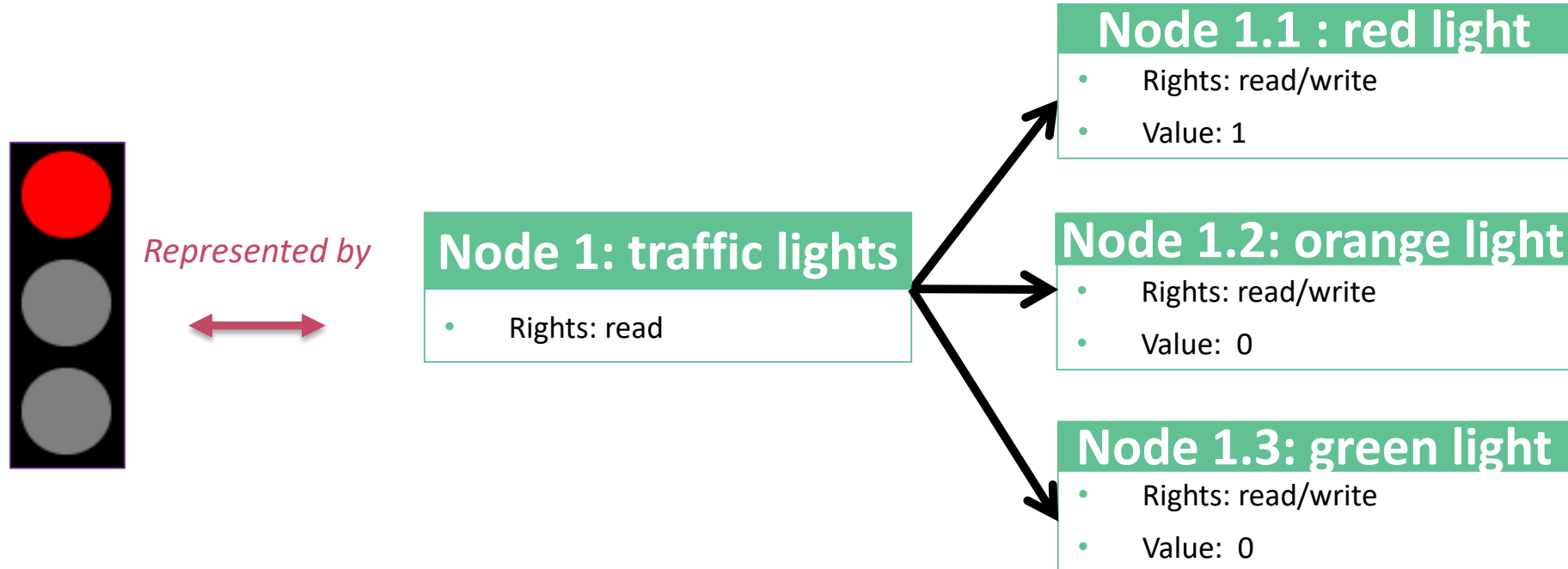
How it works

- 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).

OPC-UA



OPC-UA

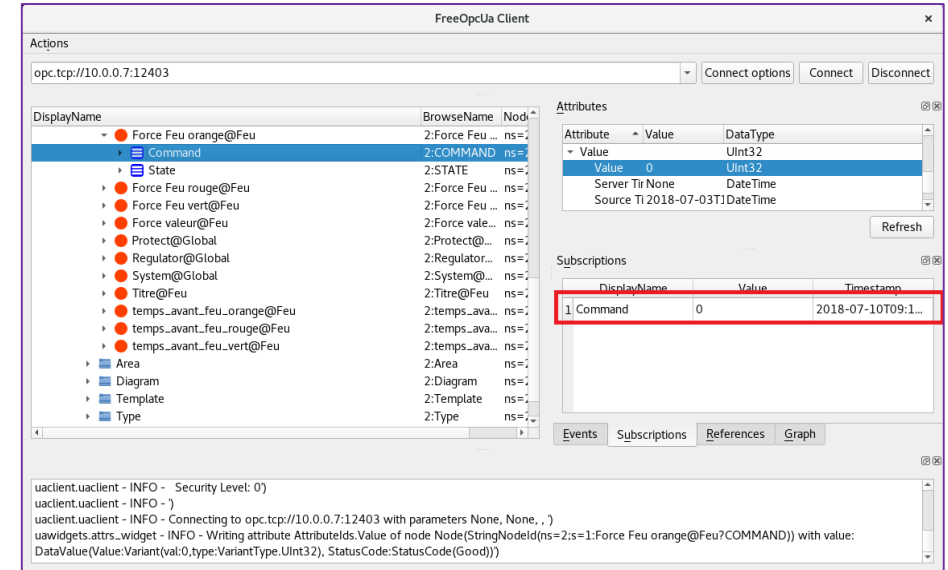
A promising protocol

- Security features integrated by design to the protocol
- Possible integration to an AD
- Fine-grained rights and privileges

But

- Current implementations are weak
- Only a few compatible PLCs
- 17 zero-day vulnerabilities in the OPC foundation products
- Ease the comprehension of the industrial process for an attacker

```
root@kali:~/Documents/toolz/modbus/mbtget/scripts#
./mbtget -r1 -a 0 -n 8 127.0.0.1
values:
 1 (ad 00000):      0
 2 (ad 00001):      0
 3 (ad 00002):      0
 4 (ad 00003):      0
 5 (ad 00004):      0
 6 (ad 00005):      0
```



Scenarii/Security mode	« None »	« Sign »	« Encrypt and sign »
Connection to the server	Yes	Yes	No
Identity theft	Yes	No	No
Trick the supervision	Yes	No	No
Information gathering	Yes	Yes	No

Lab Session #1b: Analyzing an OPC-UA communication with Wireshark

- Analyze an opc-ua communication with Wireshark
- Wireshark owns by default an opc-ua dissector

No.	Time	Source	Destination	Protocol
149	17.937435	192.168.0.15	10.0.2.15	OpcUa
150	17.940843	10.0.2.15	192.168.0.15	OpcUa
152	17.945418	192.168.0.15	10.0.2.15	OpcUa
154	23.039340	10.0.2.15	192.168.0.15	OpcUa
156	23.045154	192.168.0.15	10.0.2.15	OpcUa
158	23.053165	10.0.2.15	192.168.0.15	OpcUa
160	23.067247	192.168.0.15	10.0.2.15	OpcUa
162	23.344418	192.168.0.15	10.0.2.15	OpcUa
164	23.345797	10.0.2.15	192.168.0.15	OpcUa
166	26.128772	10.0.2.15	192.168.0.15	OpcUa
168	26.132215	192.168.0.15	10.0.2.15	OpcUa
169	26.134039	10.0.2.15	192.168.0.15	OpcUa
171	26.137928	192.168.0.15	10.0.2.15	OpcUa
173	26.854666	192.168.0.15	10.0.2.15	OpcUa

▼ OpcUa Binary Protocol
Message Type: MSG
Chunk Type: F
Message Size: 154
SecureChannelId: 6
Security Token Id: 1
Security Sequence Number: 42
Security RequestId: 42
▼ OpcUa Service : Encodeable Object
► TypeId : ExpandedNodeId
▼ WriteRequest
► RequestHeader: RequestHeader
▼ NodesToWrite: Array of WriteValue
ArraySize: 1
▼ [0]: WriteValue
► NodeId: NodeId

- Launch Wireshark
- Open « opcua.pcap »
- Try to understand what's going on
 - Browse request
 - Read request
 - Write request
 - Create subscription request
 - Create monitored item request
 - Publish request
- Which node has been changed and what was the value?

Lab session #2b: 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

FreeOpcUa Client

Actions

opc.tcp://10.0.0.7:12403

Connect options Connect Disconnect

DisplayName	BrowseName	NodeId
Root	0:Root	i=84
Objects	0:Objects	i=85
Server	0:Server	i=2253
IGSS Objects	2:IGSS Obj...	ns=2;s=...
(All)	2:(All)	ns=2;s=...
Alimentation@Feu	2:Alimentati...	ns=2;s=...
Cadre@Feu	2:Cadre@Feu	ns=2;s=...
Cadre_1@Feu	2:Cadre_1@...	ns=2;s=...
Driver@Global	2:Driver@Gl...	ns=2;s=...
Feu orange@Feu	2:Feu orang...	ns=2;s=...
Feu rouge@Feu	2:Feu rouge...	ns=2;s=...
Feu vert@Feu	2:Feu vert@...	ns=2;s=...
Command	2:COMMAND	ns=2;s=...
State	2:STATE	ns=2;s=...
Force Feu orange@Feu	2:Force Feu ...	ns=2;s=...
Force Feu rouge@Feu	2:Force Feu ...	ns=2;s=...
Force Feu vert@Feu	2:Force Feu ...	ns=2;s=...
Force valeur@Feu	2:Force vale...	ns=2;s=...
Protect@Global	2:Protect@...	ns=2;s=...
Regulator@Global	2:Regulator...	ns=2;s=...
System@Global	2:System@...	ns=2;s=...
Titre@Feu	2:Titre@Feu	ns=2;s=...
temps_avant_feu_orange@Feu	2:temps_ava...	ns=2;s=...
temps_avant_feu_rouge@Feu	2:temps_ava...	ns=2;s=...

Attributes

Attribute	Value	DataType
AccessLevel	CurrentRead, CurrentWrite	Byte
ArrayDimens	None	Null
BrowseNam	2:COMMAND	QualifiedName
DataType	UInt32	NodeId
Description	The command value	LocalizedText
DisplayName	Command	LocalizedText
Historizing	False	Boolean
MinimumExp...		Double

Refresh

Subscriptions

DisplayName	Value	Timestamp
1 State	1	2018-07-05T15:5...
2 Command	1	2018-07-05T15:5...

Events Subscriptions References Graph

File "/usr/local/lib/python3.5/dist-packages/uclient/uclient.py", line 113, in subscribe_events
 handle = self._event_sub.subscribe_events(node)
 File "/usr/local/lib/python3.5/dist-packages/opcu/common/subscription.py", line 197, in subscribe_events

RSAConference2020

Capture the flag!

WARNING

The following show features stunts performed either by professionals or under supervision of professionals.

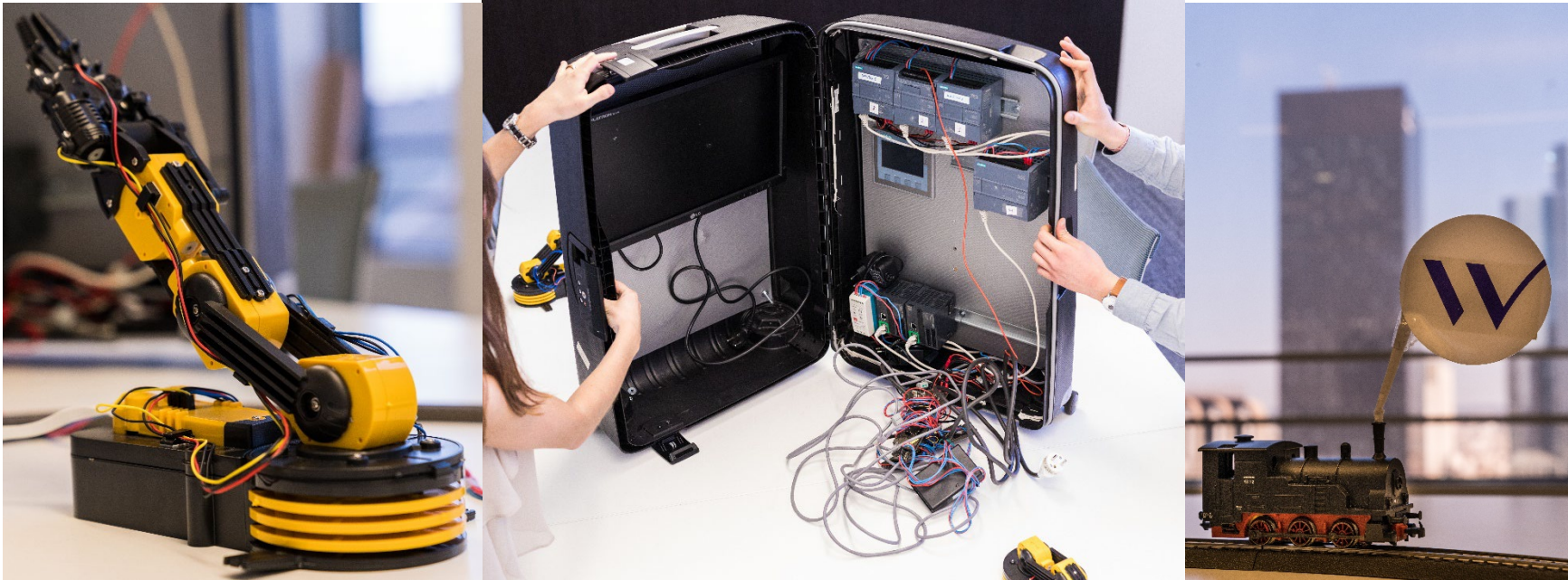
Attacking
PLCs

Never do this
on **LIVE production** systems



Capture the flag

Stop the train and capture the flag with the robot arm



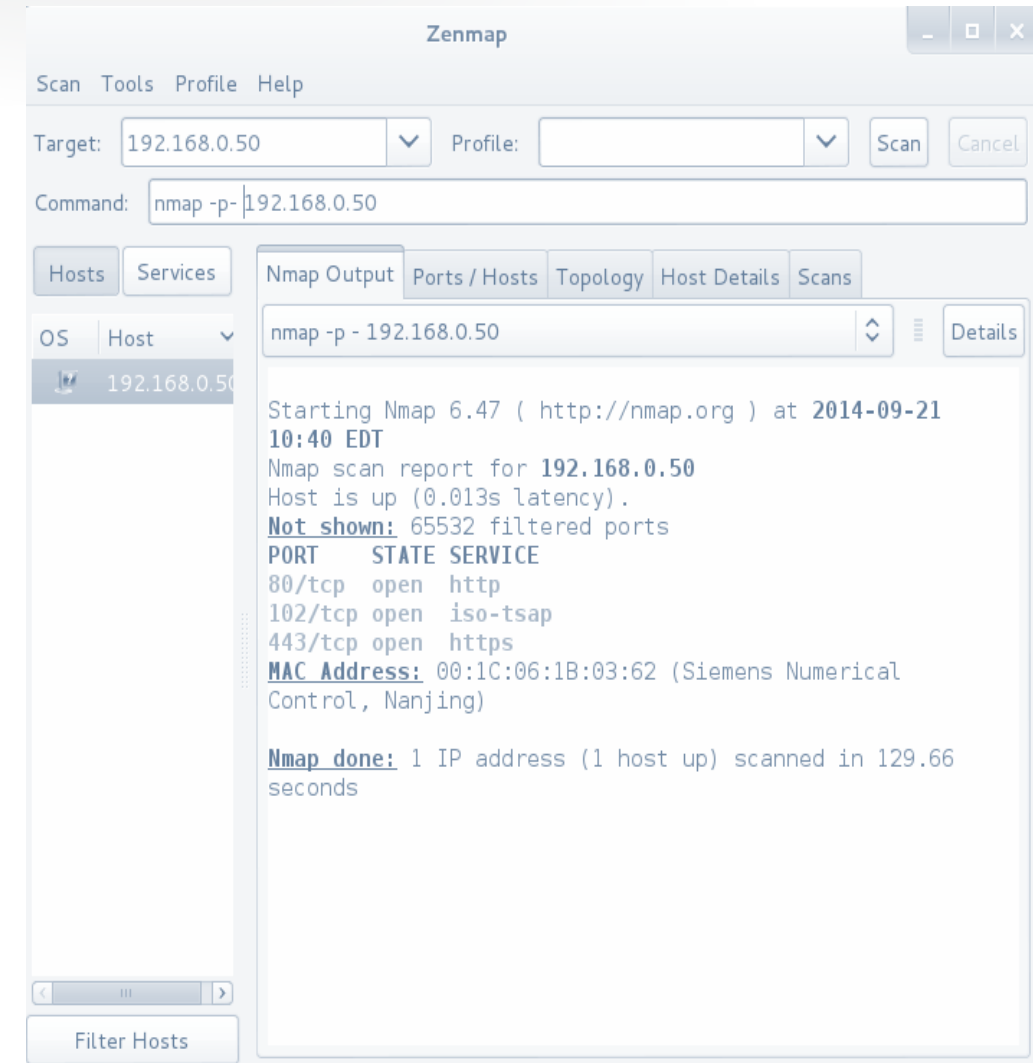
Reconnaissance

- Objective : Identify all exposed services on a device or a range of devices
- Is often the first step in a pentest
- We will use **Nmap**, the world's finest port scanner

Network information : Wifi SSID: "ICS_101" (pass : "yoloscada")
DHCP to obtain an address

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>
```



RSA®Conference2020

Takeaways

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 organizational and technical security measures are necessary



Securing ICS: where to start?

1. Know your ICS & your industrial processes

1. Start by going on-site! Do a site tour, meet the people, understand the context. Each industry has its own very specific constraints
2. Identify & map the ICS resources (servers, PLCs, other low-level devices)

2. Limit your exposure

1. Separate IT & OT networks (logically at least)
2. Limit exchanges with the IT to what's absolutely necessary
3. Ensure no direct access to low-level devices from IT / Internet

3. Patch wisely

1. Is it really worth the effort patching PLC vulnerabilities when you have foreverdays?
2. Being able to quickly patch a vulnerability exploited in the wild is probably more important than installing all patches every month
3. Everything that can be reached by the corporate IT or an external network must be patched regularly

4. Business continuity

1. Beware of ransomwares! Have offline backups, and try to perform a restore at least once a year
2. Discuss business continuity with people on-site, and ensure they took OT into consideration