Brucon 2016 Workshop –

Start To Pentest ICS

Tijl Deneut

Hendrik Derre

Introduction

- **Hendrik Derre**
  - Research associate at KU Leuven
  - Master's degree in engineering technologies
  - Currently active in industrial security research project
  - @derre_h / www.linkedin.com/in/hendrikderre

- **Tijl Deneut**
  - Researcher and lecturer at Howest University College
  - Ethical Hacker
  - Currently active in industrial security research project
  - @tijldeneut / linkedin.com/in/tijldeneut

Introduction

What to expect?

- **ICS Crash course** (30 min)
  - Industrial Control systems
  - Programmable Logic controller
  - Industrial communication
- **ICS pen testing** (30 min)
  - Introducing our environment
  - ICS vulnerabilities
  - Start to pentest ICS
- **ICS Hacking FUN** (Full 2 days)

Industrial Control Systems

Industrial Control Systems

"An ICS is an broad class of command and control networks and systems that are used to support all types of industrial processes. "

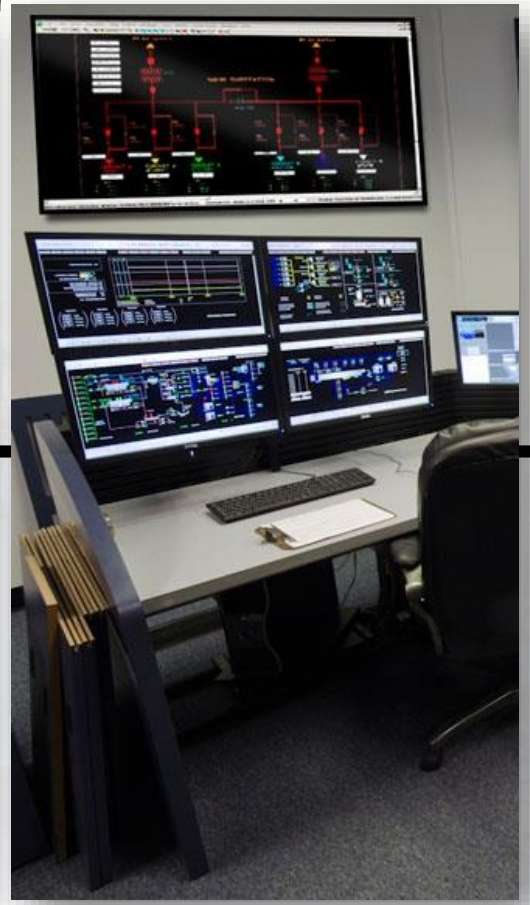
They include a **variety of system types** including:

- supervisory control and data acquisition (**SCADA**) systems,
- distributed control systems (**DCS**),
- process control systems (**PCS**),
- safety control systems (**SIS**),
- smaller control systems configurations such as programmable logic controllers (**PLC's**).

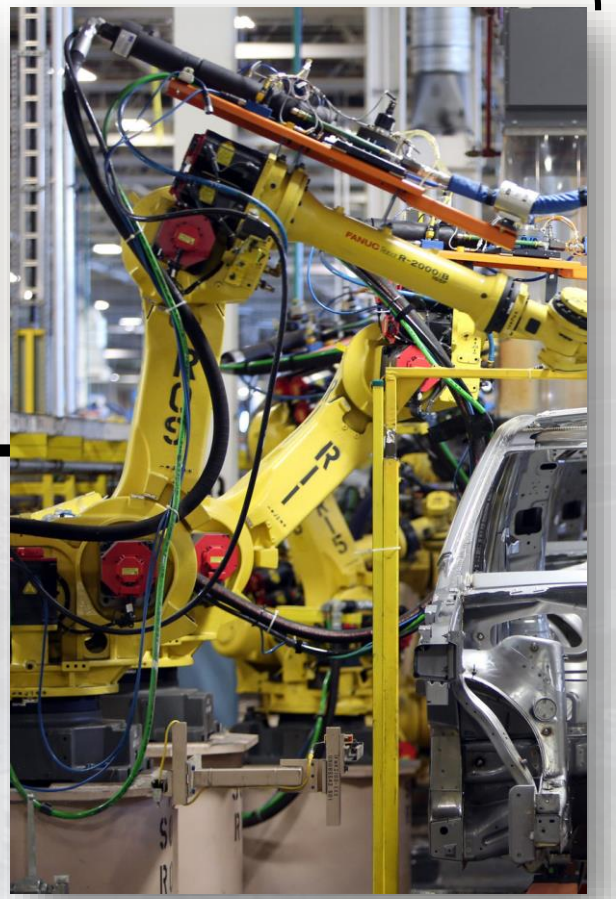
Industrial Control Systems

Office

Industrial Control System

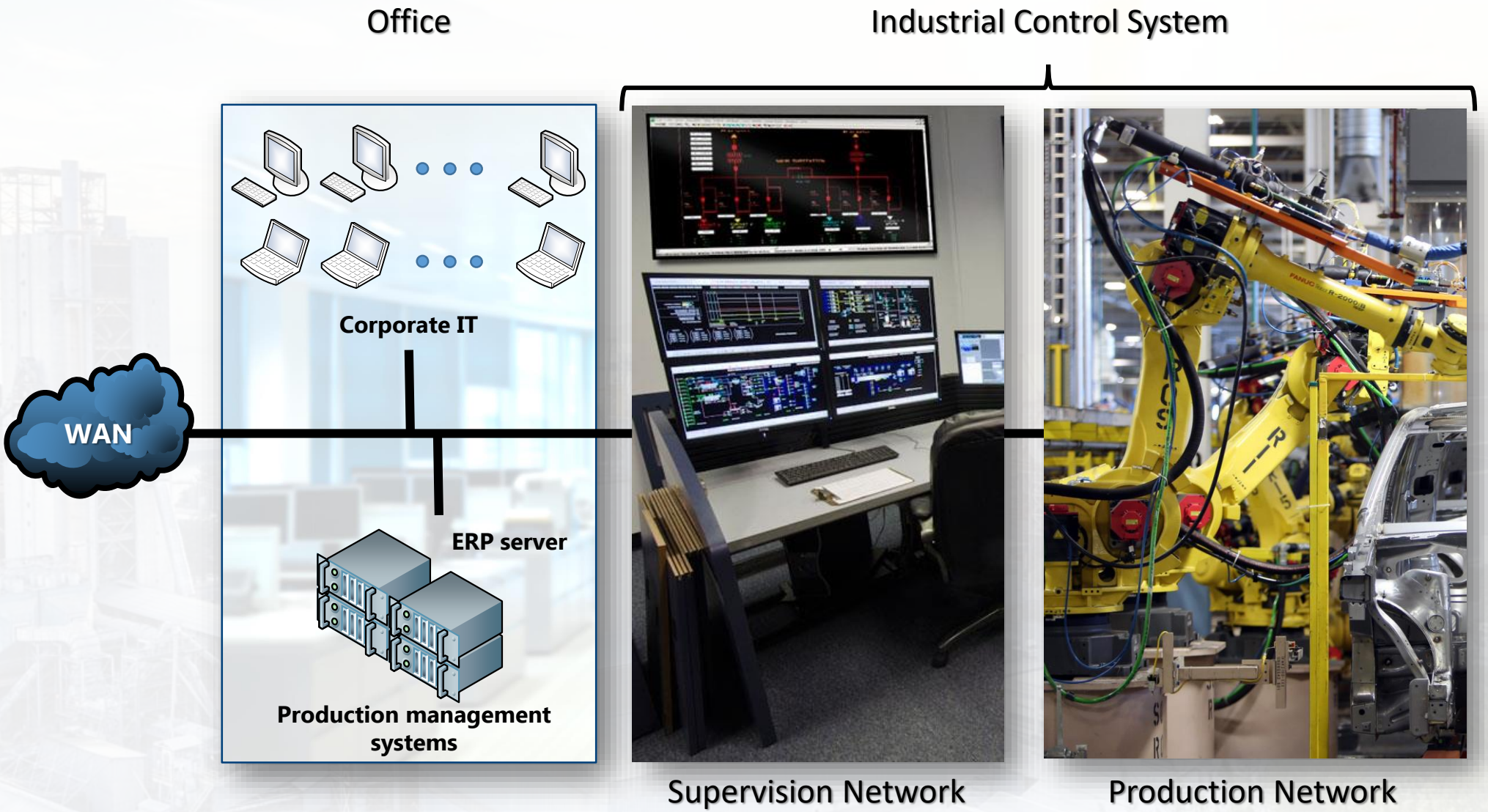


Supervision Network

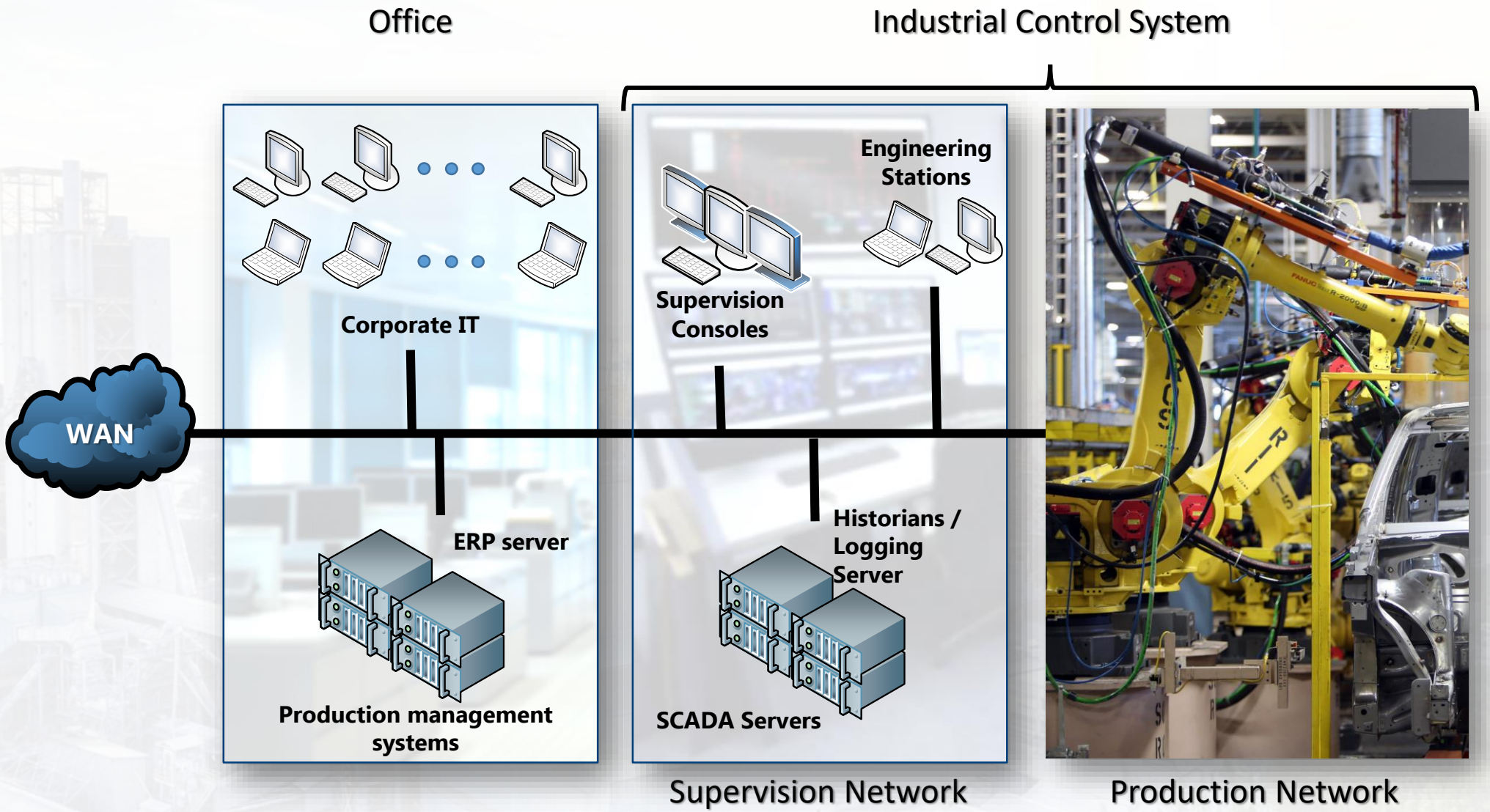


Production Network

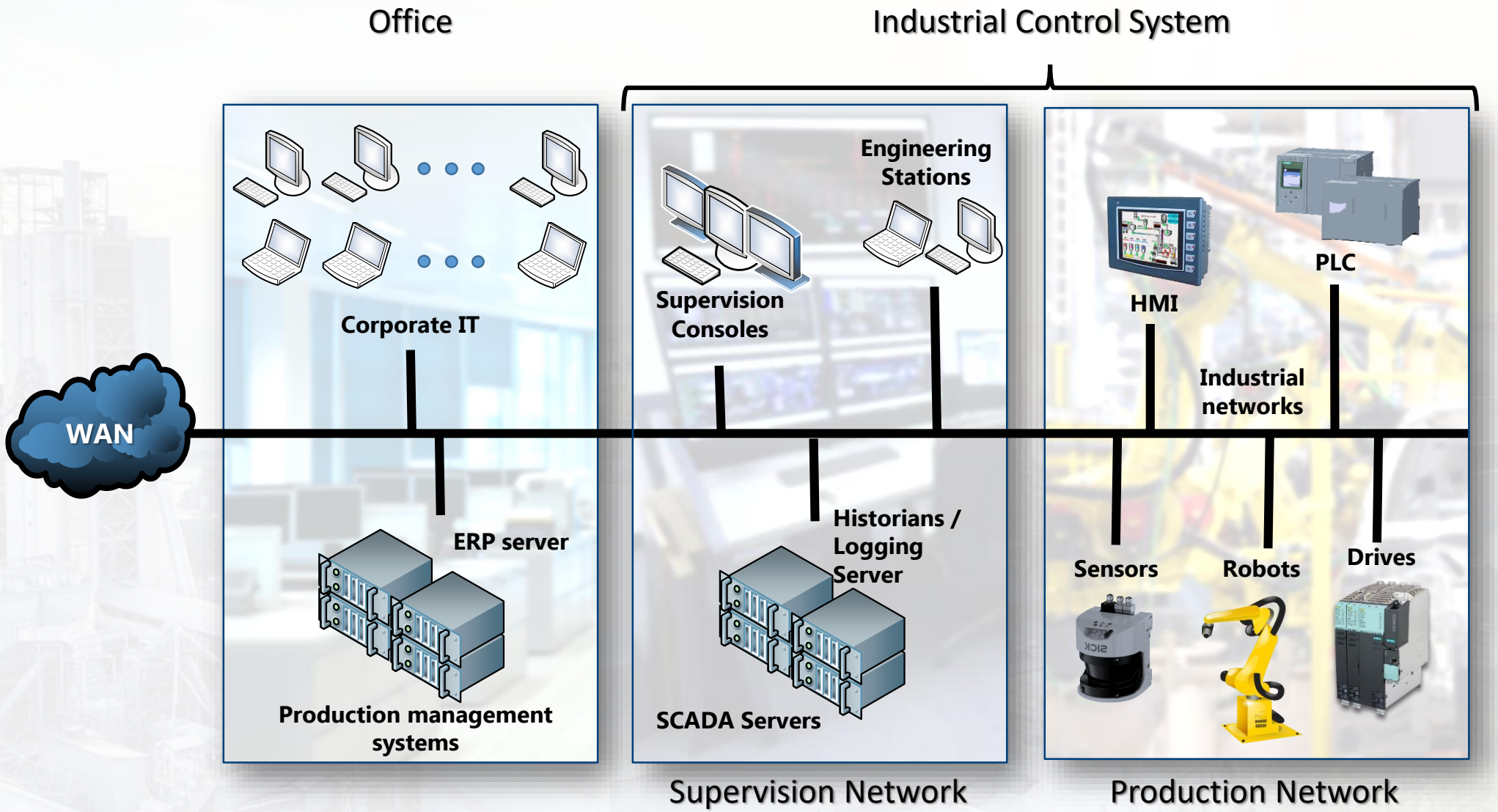
Industrial Control Systems



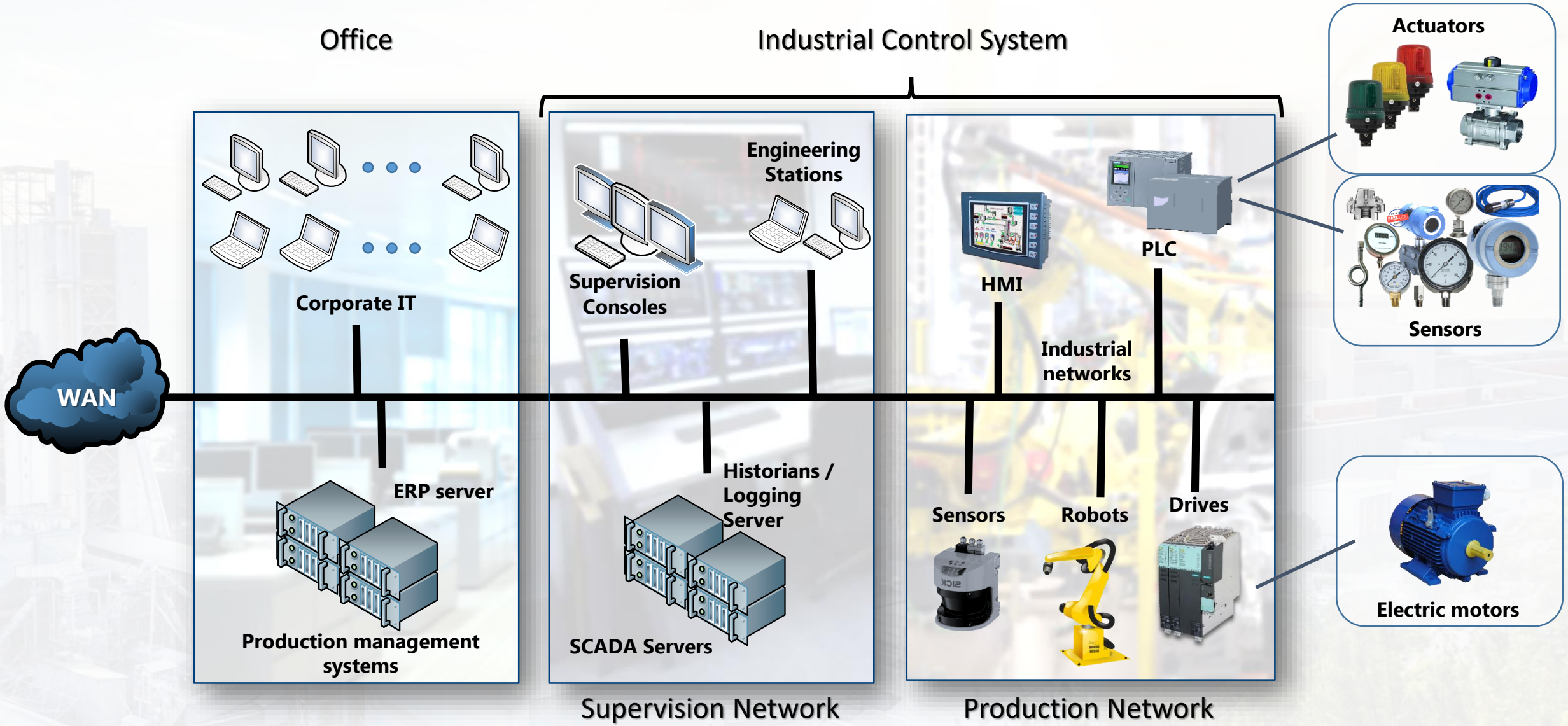
Industrial Control Systems



Industrial Control Systems



Industrial Control Systems



Industrial Control Systems

Where can we find Industrial Control systems?

Industrial Control Systems

Industrial Control Systems

Programmable Logic Controller

Industrial Control Systems

PLC: Programmable Logic Controller

A Solid-state control system that has a user-programmable memory for storing instructions for the purpose of implementing specific functions such as I/O control, logic, timing, counting, three mode (PID) control, communication, arithmetic, and data and file processing.

Industrial Control Systems

Feature		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C
Physical size (mm)		90 x 100 x 75	90 x 100 x 75	110 x 100 x 75	130 x 100 x 75
User memory	Work	30 Kbytes	50 Kbytes	75 Kbytes	100 Kbytes
	Load	1 Mbyte	1 Mbyte	4 Mbytes	4 Mbytes
	Retentive	10 Kbytes	10 Kbytes	10 Kbytes	10 Kbytes
Local on-board I/O	Digital	6 inputs/4 outputs	8 inputs/6 outputs	14 inputs/10 outputs	14 inputs/10 outputs
	Analog	2 inputs	2 inputs	2 inputs	2 inputs / 2 outputs
Process image size	Inputs (I)	1024 bytes	1024 bytes	1024 bytes	1024 bytes
	Outputs (Q)	1024 bytes	1024 bytes	1024 bytes	1024 bytes
Bit memory (M)		4096 bytes	4096 bytes	8192 bytes	8192 bytes
Signal module (SM) expansion		None	2	8	8
Signal board (SB), Battery board (BB), or communication board (CB)		1	1	1	1
Communication module (CM) (left-side expansion)		3	3	3	3
High-speed counters	Total	3 built-in I/O, 5 with SB	4 built-in I/O, 6 with SB	6	6
	Single phase	3 at 100 kHz SB: 2 at 30 kHz	3 at 100 kHz 1 at 30 kHz SB: 2 at 30 kHz	3 at 100 kHz 3 at 30 kHz	3 at 100 kHz 3 at 30 kHz
	Quadrature phase	3 at 80 kHz SB: 2 at 20 kHz	3 at 80 kHz 1 at 20 kHz SB: 2 at 20 kHz	3 at 80 kHz 3 at 20 kHz	3 at 80 kHz 3 at 20 kHz
Pulse outputs[1]		4	4	4	4
Memory card		SIMATIC Memory card (optional)			
Real time clock retention time		20 days, typ. / 12 day min. at 40 degrees C (maintenance-free Super Capicator)			
PROFINET		1 Ethernet communication port			2 Ethernet communication ports
Real math execution speed		2.3 µs/instruction			
Boolean execution speed		0.08 µs/instruction			

Industrial Control Systems

Feature		CPU 1211C	CPU 1212C	CPU 1214C
Physical size (mm)		90 x 100 x 75	90 x 100 x 75	110 x 100 x 75
User memory	Work	30 Kbytes	50 Kbytes	75 Kbytes
	Load	1 Mbyte	1 Mbyte	4 Mbytes
	Retentive	10 Kbytes	10 Kbytes	10 Kbytes
size	Outputs (Q)	1024 bytes	1024 bytes	1024 bytes	1024 bytes
Bit memory (M)		4096 bytes	4096 bytes	8192 bytes	8192 bytes
Signal module (SM) expansion		None	2	8	8
Signal board (SB), Battery board (BB), or communication board (CB)		1	1	1	1
Communication module (CM) (left-side expansion)		3	3	3	3
High-speed counters	Total	3 built-in I/O, 5 with SB	4 built-in I/O, 6 with SB	6	6
	Single phase	3 at 100 kHz SB: 2 at 30 kHz	3 at 100 kHz 1 at 30 kHz SB: 2 at 30 kHz	3 at 100 kHz 3 at 30 kHz	3 at 100 kHz 3 at 30 kHz
	Quadrature phase	3 at 80 kHz SB: 2 at 20 kHz	3 at 80 kHz 1 at 20 kHz SB: 2 at 20 kHz	3 at 80 kHz 3 at 20 kHz	3 at 80 kHz 3 at 20 kHz
Real math execution speed		2.3 µs/instruction			
Boolean execution speed		0.08 µs/instruction			
PROFINET		1 Ethernet communication port			2 Ethernet communication ports

Industrial Control Systems

Feature		CPU 1211C	CPU 1212C	CPU 1214C
Physical size (mm)		90 x 100 x 75	90 x 100 x 75	110 x 100 x 75
User memory	Work	30 Kbytes	50 Kbytes	75 Kbytes
	Load	1 Mbyte	1 Mbyte	4 Mbytes
	Retentive	10 Kbytes	10 Kbytes	10 Kbytes
size	Outputs (Q)	1024 bytes	1024 bytes	1024 bytes	1024 bytes
Bit memory (M)		4096 bytes	4096 bytes	8192 bytes	8192 bytes
Signal module (SM) expansion		None	2	8	8
Signal board (SB), Battery board (BB), or communication board (CB)		1	1	1	1
Communication module (CM) (left-side expansion)		3	3	3	3
High-speed counters	Total	3 built-in I/O, 5 with SB	4 built-in I/O, 6 with SB	6	6
	Single phase	3 at 100 kHz SB: 2 at 30 kHz	3 at 100 kHz 1 at 30 kHz SB: 2 at 30 kHz	3 at 100 kHz 3 at 30 kHz	3 at 100 kHz 3 at 30 kHz
	Quadrature phase	3 at 80 kHz SB: 2 at 20 kHz	3 at 80 kHz 1 at 20 kHz SB: 2 at 20 kHz	3 at 80 kHz 3 at 20 kHz	3 at 80 kHz 3 at 20 kHz
Real math execution speed		2.3 µs/instruction			
Boolean execution speed		0.08 µs/instruction			
PROFINET		1 Ethernet communication port			2 Ethernet communication ports
Real math execution speed		2.3 µs/instruction			
Boolean execution speed		0.08 µs/instruction			

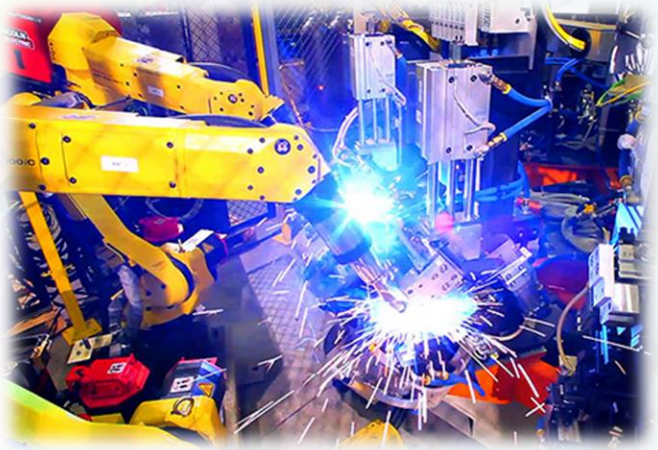
Industrial Control Systems

Why do we even use PLC's?

Industrial Control Systems

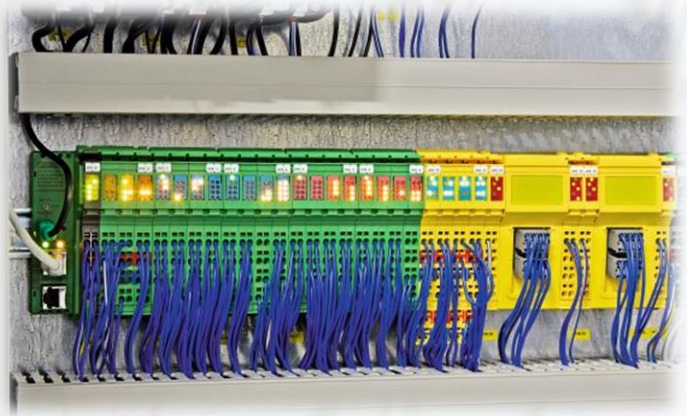
Key PLC features:

- Reliable & Robust
- Extreme long product lifecycle
- Flexible
- Standardised I/0 signals
- Easy programming and debugging



Industrial Control Systems

All Shapes and Sizes

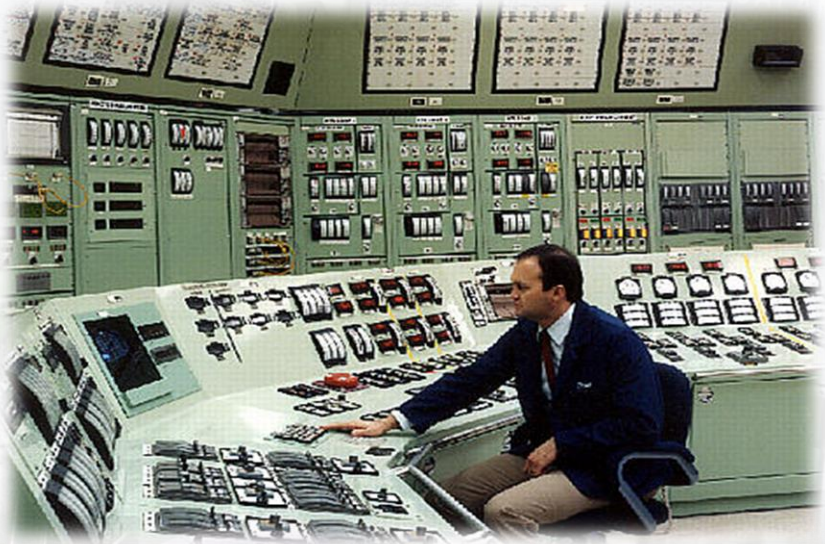


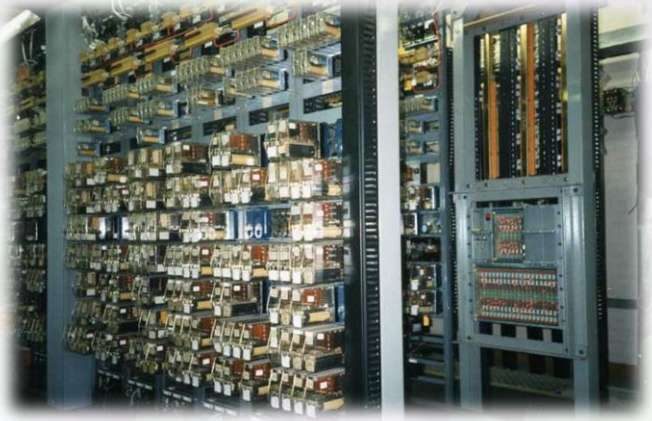
Industrial Communication

Industrial Communication

How it started:

- Serial Communications
- Proprietary Protocols
- Limited Interconnections
- No external Communications



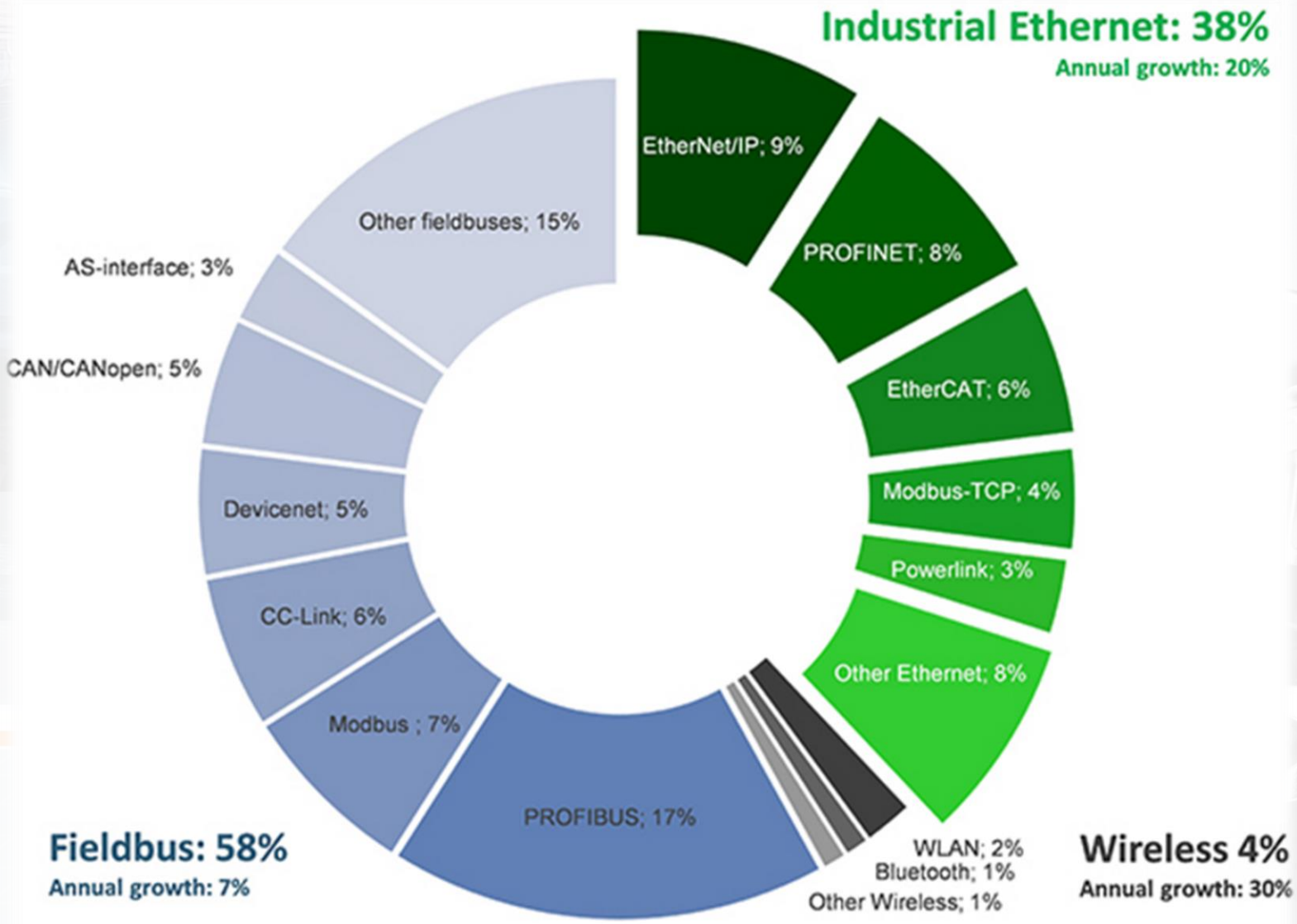


Industrial Communication

Present:

- Evolving towards Ethernet based communications
- Open Protocols
- Interconnected with other business systems
- Internet connections for remote access

Industrial Communication



Source: HMS (2016)

Industrial Communication

Ethernet Based Protocols

- **Universal ICS Protocols**
  - Modbus TCP: TCP/502
  - OPC UA: TCP/4840
  - OP UA XML: TCP/80, TCP/443

- **Process automation specific protocols**
  - EtherCat: UDP/34980
  - Ethernet/IP: TCP/44818, UDP/2222,44818
  - FL-net: UDP/55000 to 55003
  - Fieldbus HSE: TCP/1089-1091, UDP/1089-1091
  - HART-IP: TCP/5094, UDP/5094
  - PROFINET: TCP/34962-34964, UDP/ 34962-34964
  - S7-Comm: TCP/102

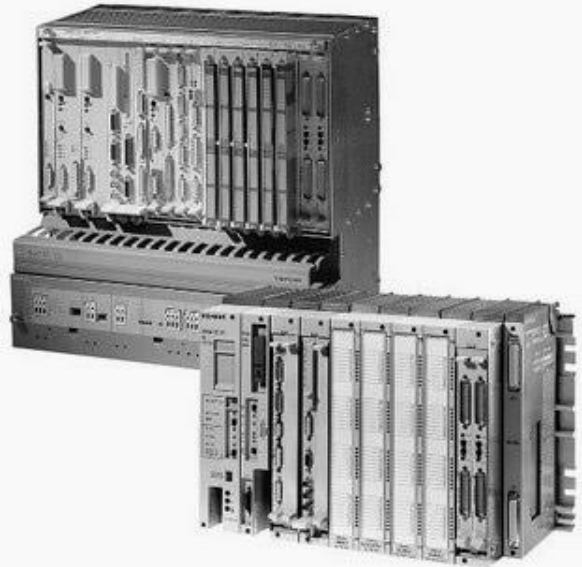
- **Building automation Specific protocols**
  - BACnet/IP: UDP/47808
  - LonTalk: UDP/1628, UDP/1629
  - FOX (Tridium/niagara): TCP/1911

- **Energy Sector specific protocols**
  - DNP3: TCP/20000, UDP/ 20000
  - DLMS/COSEM: TCP/4059, UDP/4059
  - ICCP: TCP/102
  - IEC 104: TCP/102
  - IEE C37.118: TCP/4712, UDP/4713
  - MMS: TCP/102

Industrial Communication

Modbus

- Designed in 1979 by Modicon (now Schneider Electric)
- Widely adopted protocol used in multiple industries
- Mostly used on field level (sometimes between PLC & HMI)
- Open Standard, freely distributed by the modbus organization

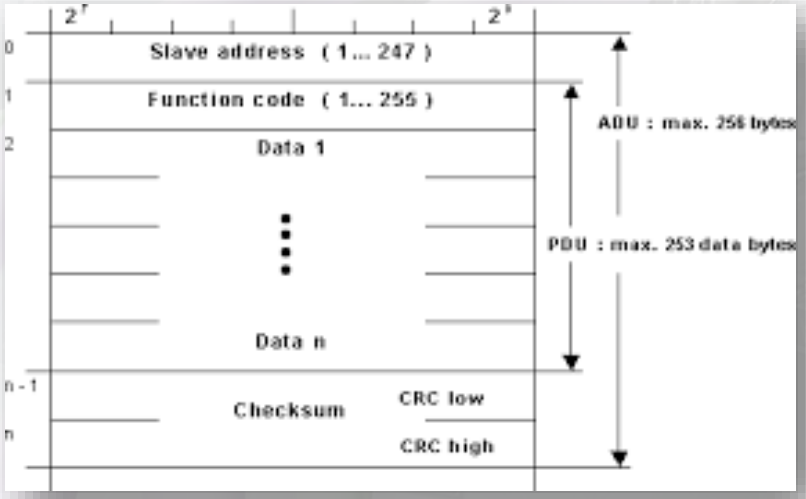


Industrial Communication

Modbus

- Master / slave (Standard 1 master and max 247 slaves in a network)
- Request / reply (Only master can initialize communication)
- Each Modbus device is assigned an unique (in the network) address.
- Function codes used to described the action desired from device

Function Code	Action	Table Name
01 (01 hex)	Read	Discrete Output Coils
05 (05 hex)	Write single	Discrete Output Coil
15 (0F hex)	Write multiple	Discrete Output Coils
02 (02 hex)	Read	Discrete Input Contacts
04 (04 hex)	Read	Analog Input Registers
03 (03 hex)	Read	Analog Output Holding Registers
06 (06 hex)	Write single	Analog Output Holding Register
16 (10 hex)	Write multiple	Analog Output Holding Registers



Industrial Communication

Modbus

- Lack of Authentication
- Lack of Encryption
- Lack of broadcast suppression

```
   153 0.206422    141.81.0.10     141.81.0.66     Modbus/TCP     66    Query: Trans:  1426; Unit: 255, Func:   2: Read Discrete Inputs
   168 0.248615    141.81.0.46     141.81.0.10     Modbus/TCP     78 Response: Trans: 28215; Unit: 255, Func:  15: Write Multiple Coils
   169 0.249547    141.81.0.26     141.81.0.10     Modbus/TCP    340 Response: Trans: 18527; Unit: 255, Func:   4: Read Input Registers
```

```
> Frame 153: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
> Ethernet II, Src: HewlettP_e0:02:5e (78:e7:d1:e0:02:5e), Dst: ElauAg_02:36:35 (00:04:17:02:36:35)
> Internet Protocol Version 4, Src: 141.81.0.10, Dst: 141.81.0.66
> Transmission Control Protocol, Src Port: 54138 (54138), Dst Port: 502 (502), Seq: 13, Ack: 12, Len: 12
v Modbus/TCP
    Transaction Identifier: 1426
    Protocol Identifier: 0
    Length: 6
    Unit Identifier: 255
v Modbus
    Function Code: Read Discrete Inputs (2)
    Reference Number: 0
    Bit Count: 11
```

Security?

Industrial Communication

SIEMENS *S7-Comm*

- S7 Protocol, is the backbone of the Siemens communications,
- Ethernet implementation relies on ISO TCP (RFC1006)
- S7 Protocol is Function oriented or Command oriented

SIEMENS *S7-Comm*

- Each command consists of
  - A header
  - A set of parameters
  - A parameters data (optional)
  - A data block (optional)

SIEMENS *S7-Comm*

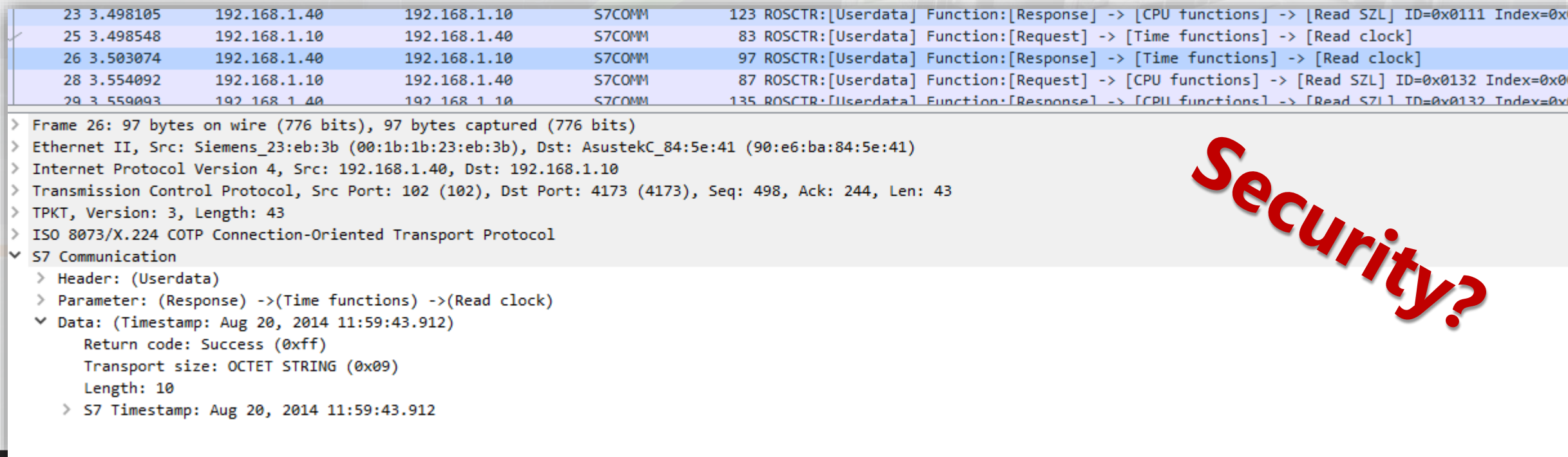
- Each command consists of
  - A header
  - A set of parameters
  - A parameters data (optional)
  - A data block (optional)

	CPU					
	300	400	WinAC	Snap7S	1200	1500
DB Read/Write	O	O	O	O	O	O(3)
EB Read/Write	O	O	O	O	O	O
AB Read/Write	O	O	O	O	O	O
MK Read/Write	O	O	O	O	O	O
TM Read/Write	O	O	O	O	-	-
CT Read/Write	O	O	O	O	-	-
Read SZL	O	O	O	O	O	O
Multi Read/Write	O	O	O	O	O	O
Directory	O	O	O	O	-	-
Date and Time	O	O	O	O	-	-
Control Run/Stop	O	O	O	O	-	-
Security	O	O	O	O	-	-
Block Upload/Down/Delete	O	O	O	-	-	-

Industrial Communication

SIEMENS *S7-Comm*

- Lack of Authentication
- Lack of Encryption

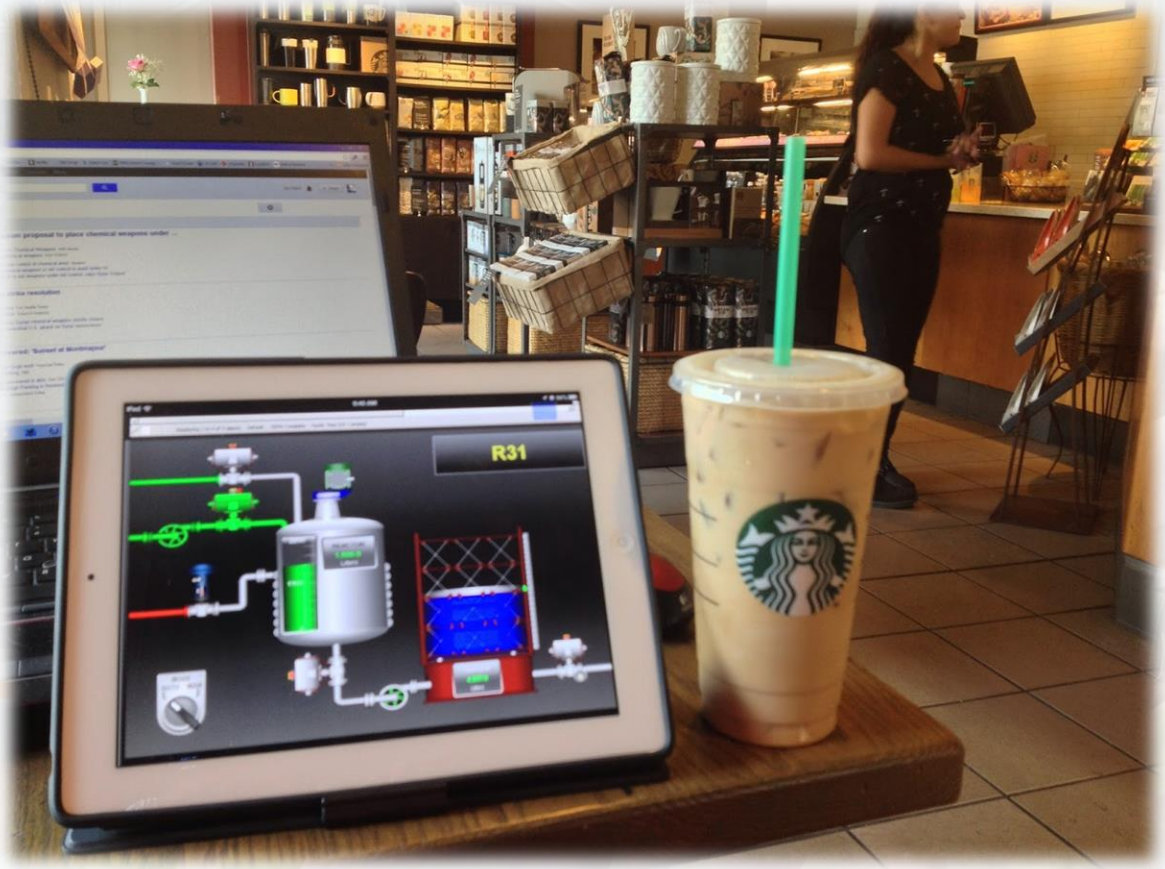


Security?

Industrial Communication

Remote access over internet to ICS networks

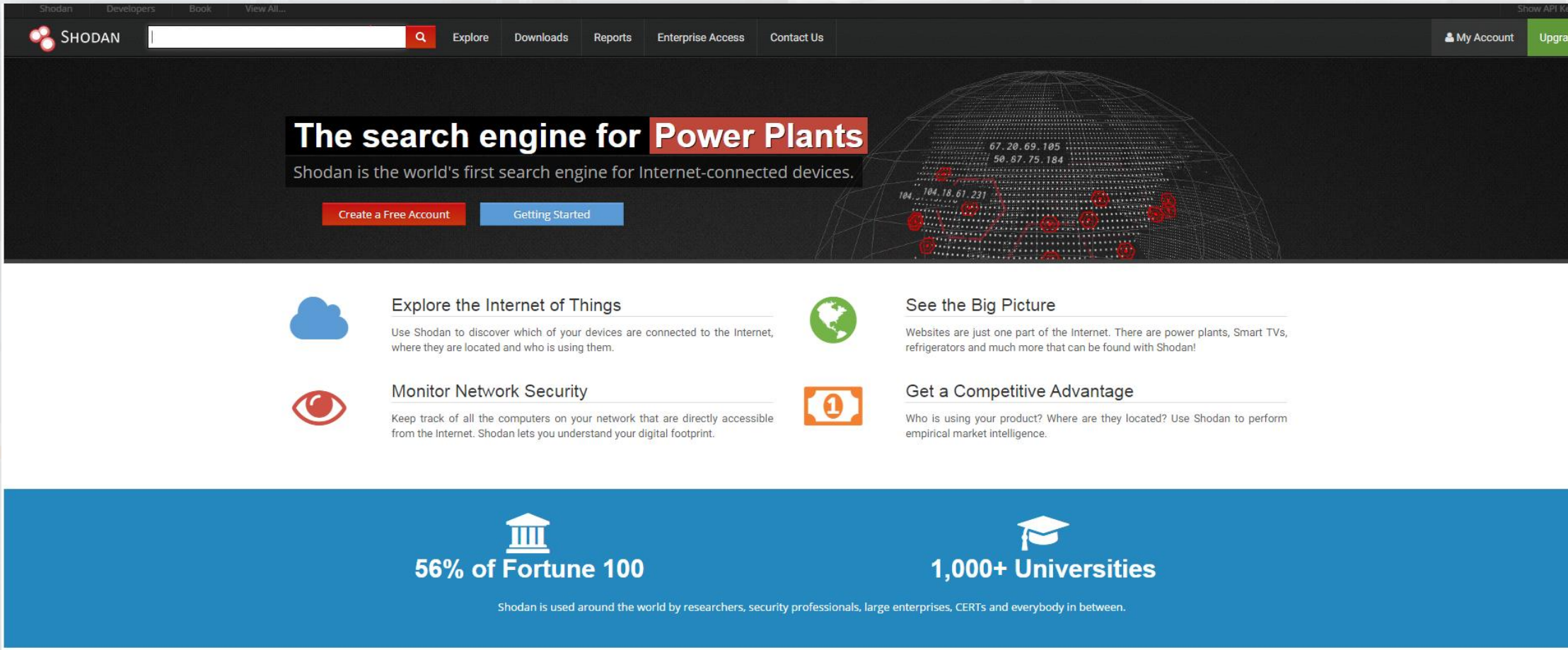
what can go wrong?



Industrial Communication

Remote access over internet to ICS networks

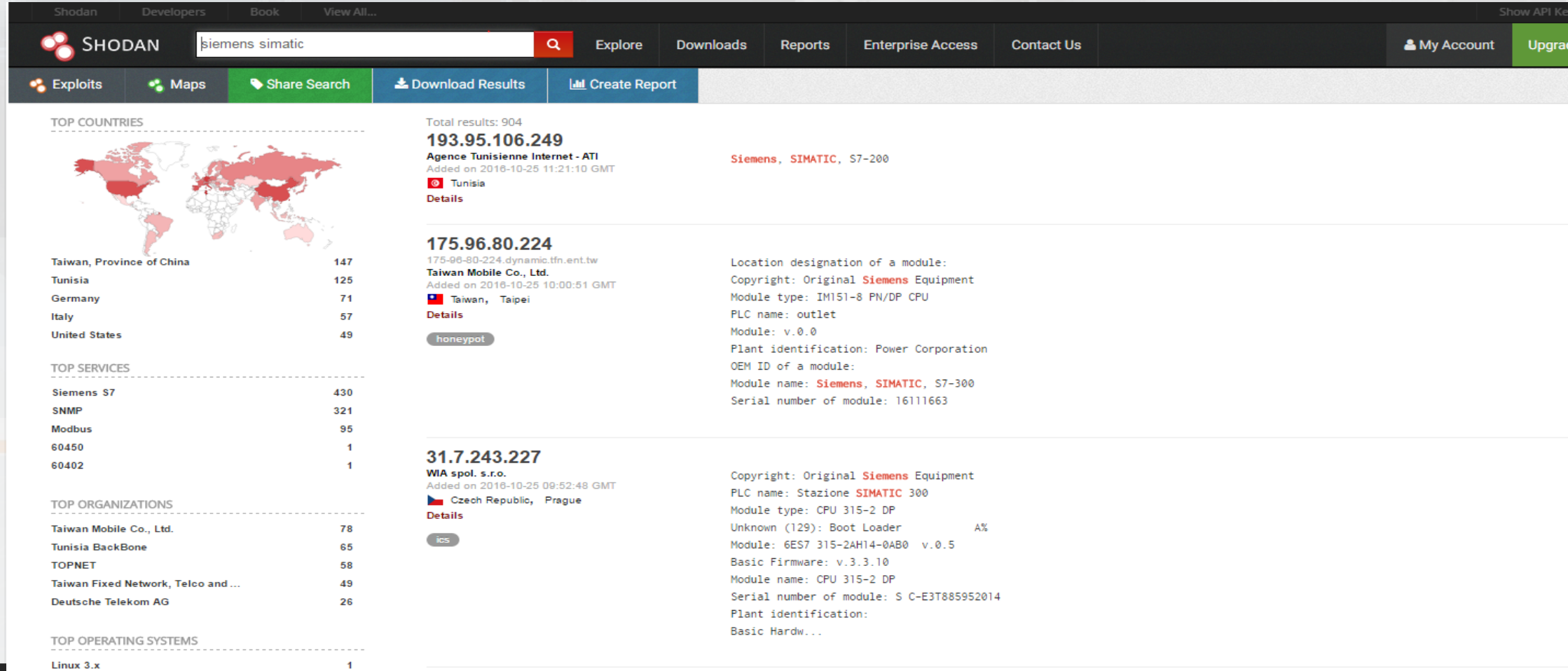
what can go wrong?



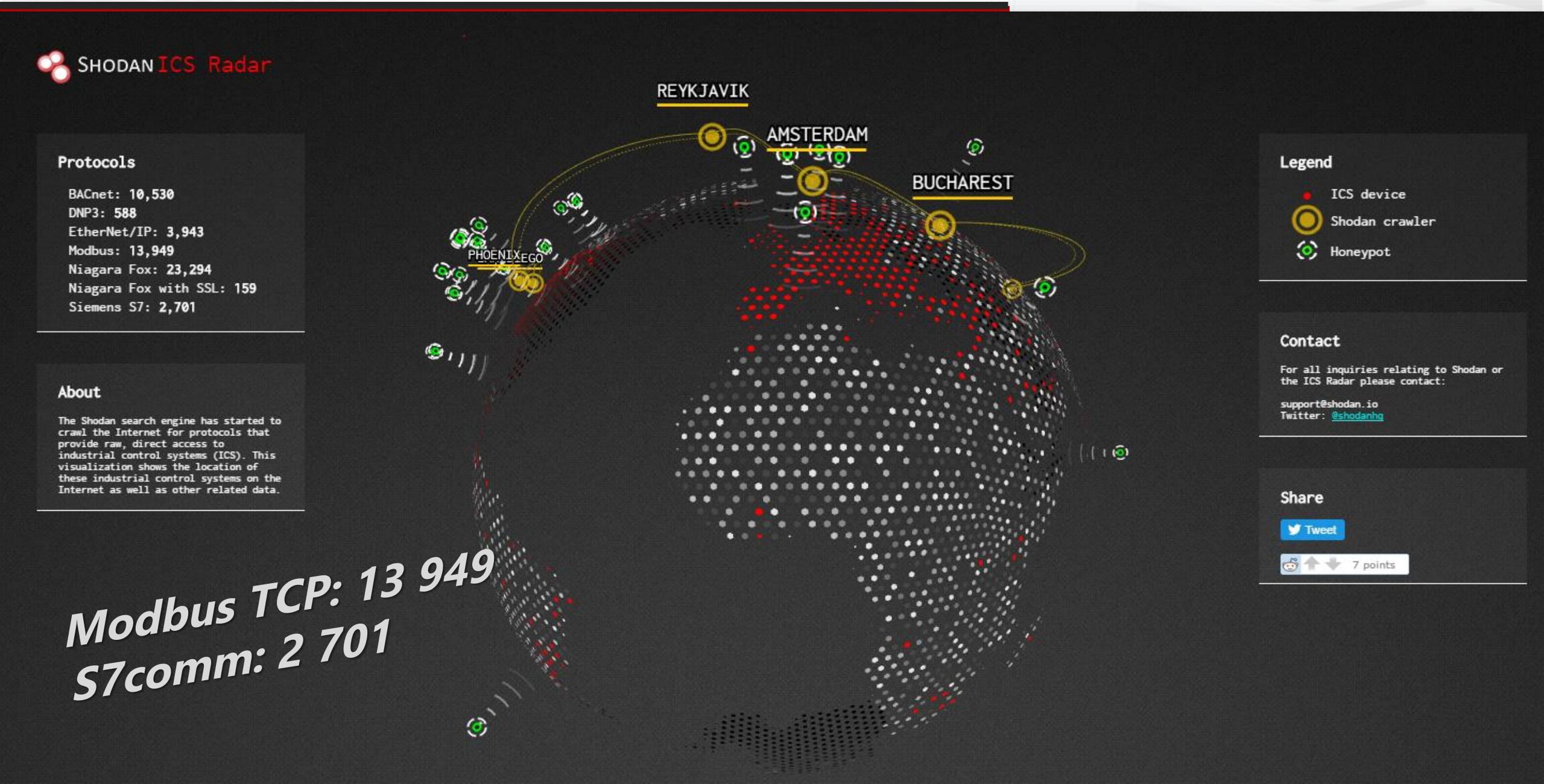
Industrial Communication

Remote access over internet to ICS networks

what can go wrong?



Industrial Communication

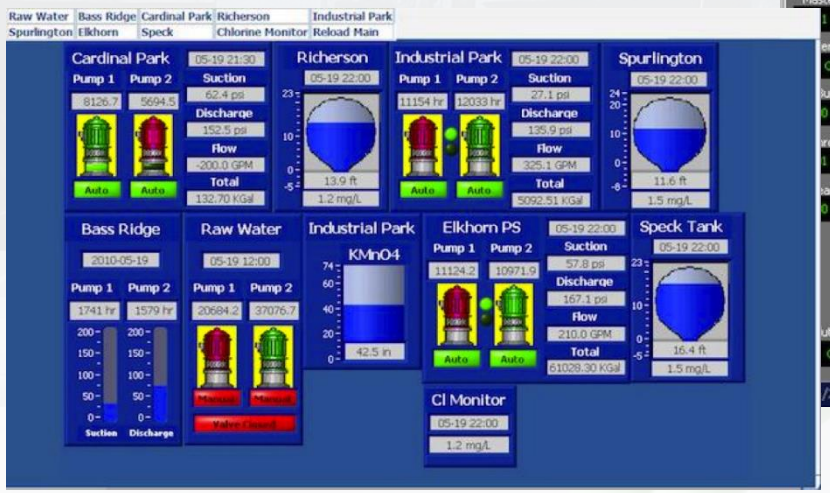


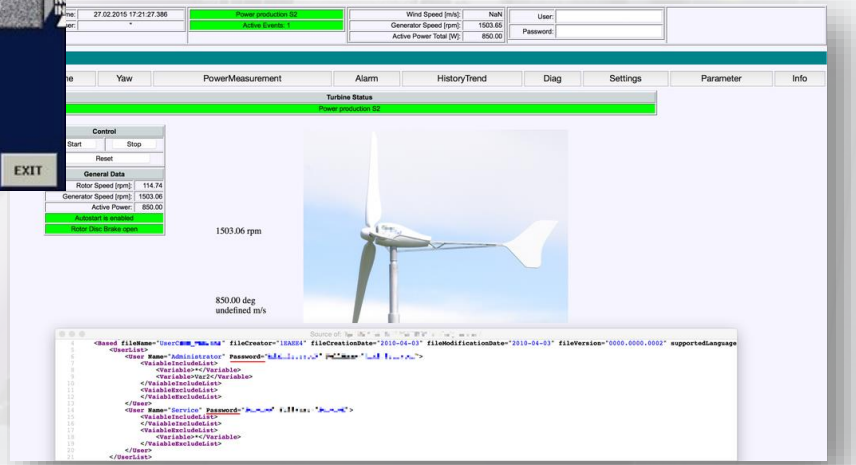
Modbus TCP: 13 949
S7comm: 2 701

Industrial Communication

Remote access over internet to ICS networks

what can go wrong?





Protocols are hard...

But we have webservers ☺

Why use passwords when nobody knows your IP...

Industrial Research

- ICS Vulnerabilities

General Vulnerability Overview

In **BELGIUM** (focus for my research), there are three most-used PLC vendors:

General Vulnerability Overview

Early vulnerabilities

- Beckhoff: http://<ip>/config has a website with a Session ID vulnerability
  - CVE-2014-4051, disclosed in 2015, fixed **3.1.4018.13 (August 2015)**
  - Authentication bypass, exploit: https://www.exploit-db.com/exploits/38514
  - PLUS **a lot** of issues with the used **Windows CE (6),** CERDisplay anyone?

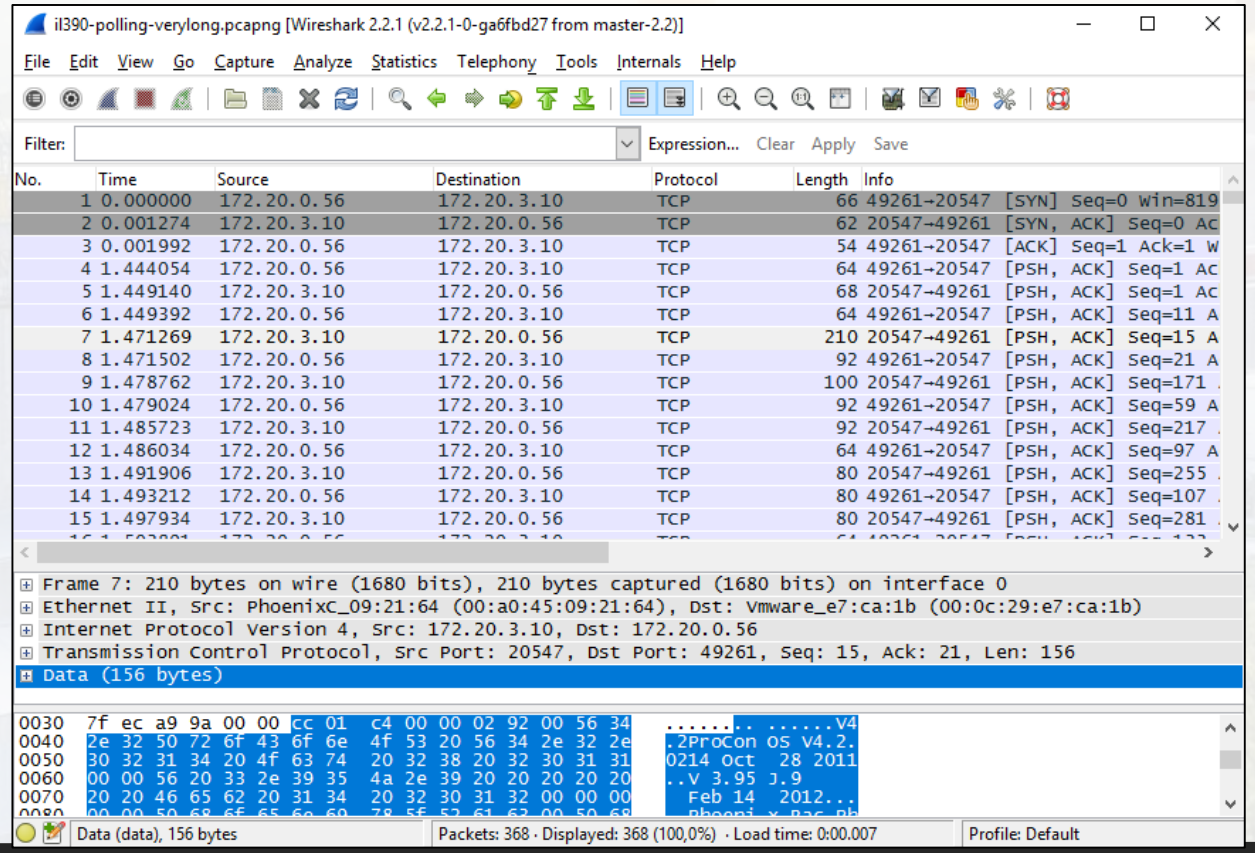
- Siemens:
  - CVE-2013-5709, Scalance Switches have a website with a Session ID vulnerability, disclosed by Eireann Leverett (blackswanburst) fixed with firmware **v5.0.0**
  - CVE-2016-6204, XSS vulnerability in newest Scalance Router (Cinema Remote Connect), disclosed this summer by me ☺ fixed with **v1.2**

But things are WORSE

As seen earlier; industrial devices rely on **old** and **insecure** protocols.

So some personal research to investigate these protocols …

- Phoenix Contact: completely proprietary, not even Wireshark has any idea what we are dealing with



Phoenix Contact, the verdict

By sending A LOT of data and investigating the resulting packets, I got to deduce the necessary packets

Resulting in **a script to control PLCs**

Just replaying only the necessary packets, while changing some parameters

Reading out the PLC state + controlling it

RESULT

```
ef initMonitor(s):
    send_and_recv(s,1000,'0100000000002f000000000000000cfff4164652e52656d6f74696e672e53657276696365732e4950726f436f6e4f
    send_and_recv(s,1000,'0100000000002e000000000000000000004164652e52656d6f74696e672e53657276696365732e4950726f436f6e4f
    send_and_recv(s,1000,'0100000000002900000000000000000004164652e52656d6f74696e672e53657276696365732e4944617461416363
    send_and_recv(s,1000,'0100000000002a000000000000000d4ff4164652e52656d6f74696e672e53657276696365732e4944657669636549
    send_and_recv(s,1000,'0100000000002900000000000000000004164652e52656d6f74696e672e53657276696365732e4944657669636549
    send_and_recv(s,1000,'0100000000002500000000000000000d9ff4164652e52656d6f74696e672e53657276696365732e49466f7263655365
    send_and_recv(s,1000,'0100000000002400000000000000000004164652e52656d6f74696e672e53657276696365732e49466f7263655365
    send_and_recv(s,1000,'0100000000003000000000000000000ceff4164652e52656d6f74696e672e53657276696365732e4953696d706c6546
    send_and_recv(s,1000,'0100000000003000000000000000000004164652e52656d6f74696e672e53657276696365732e4953696d706c6546
    send_and_recv(s,1000,'0100000000002a000000000000000d4ff4164652e52656d6f74696e672e53657276696365732e4944657669636549
    send_and_recv(s,1000,'0100000000002900000000000000000004164652e52656d6f74696e672e53657276696365732e4944657669636549
    send_and_recv(s,1000,'0100000000002a000000000000000d4ff4164652e52656d6f74696e672e53657276696365732e4944617461416363
    send_and_recv(s,1000,'0100000000002900000000000000000004164652e52656d6f74696e672e53657276696365732e4944617461416363
    send_and_recv(s,1000,'0100000000002a000000000000000d4ff4164652e52656d6f74696e672e53657276696365732e4944617461416363
    send_and_recv(s,1000,'0100000000002900000000000000000d5ff4164652e52656d6f74696e672e53657276696365732e49427265616b706f
    send_and_recv(s,1000,'0100000000002800000000000000000d6ff4164652e52656d6f74696e672e53657276696365732e4943616c6c737461
    send_and_recv(s,1000,'0100000000002500000000000000000004164652e52656d6f74696e672e53657276696365732e4944656275675365
    send_and_recv(s,1000,'0100000000002f000000000000000cfff4164652e52656d6f74696e672e53657276696365732e4950726f436f6e4f
    send_and_recv(s,1000,'0100000000002e000000000000000000004164652e52656d6f74696e672e53657276696365732e4950726f436f6e4f
    send_and_recv(s,1000,'0100000000003000000000000000000ceff4164652e52656d6f74696e672e53657276696365732e4953696d706c6546
    send_and_recv(s,1000,'0100000000003000000000000000000004164652e52656d6f74696e672e53657276696365732e4953696d706c6546
    send_and_recv(s,1000,'0100020000000e0003000300000000005000000124013401300114012000')
    return
```

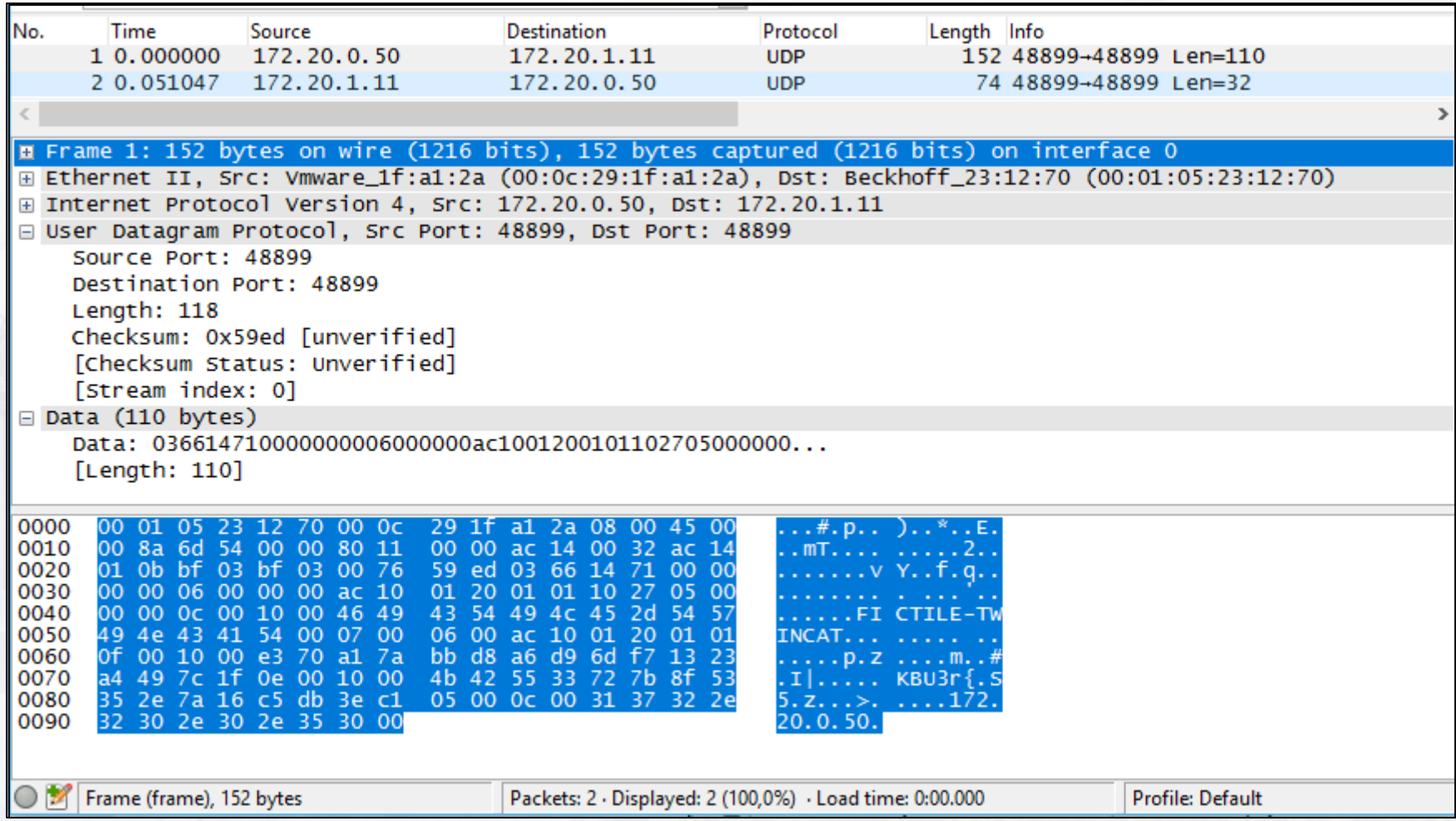
Beckhoff, the verdict

Beckhoff uses a software implementation called “Twincat”, the underlying protocol is called AMS and uses the concept of “routes” to allow PCs to program the PLC. A PLC without routes does not listen!

AMS is understood by Wireshark!

Adding routes is done using a proprietary, unknown UDP protocol! It requires a password, but can be sniffed.

However:
a bug in Beckhoff CX9010, CP66xx and any other WinCE6 PLC:
Password is ignored!



Siemens, the verdict

Siemens uses S7Comm, which has already been researched closely, e.g. in snap7, scan7 or plcscan

However, **every ICS Vendor** uses some kind of **broadcast** protocol to easily find their devices on any network.

-> For Beckhoff, it would be the aforementioned UDP protocol
-> For Siemens, this is called Profinet Discovery Protocol, known and understood by Wireshark ☺

So I wrote an all-in-one script …

-1- And we noticed that not only PLCs respond but also Siemens Routers, Switches, HMI's
and also PhoenixContact PLC's, IO Islands etc …
-2- There is a Profinet DCP GET, but also Profinet DCP SET.
Which allows, a.o., IP data to be set, ALSO on Switches and Routers !!

Pentesting ICS

- Start to Pentest

Pentesting ICS

- **Not** as easy as it sounds
  - ICS hardware (HMI, PLC, OPC …) has **very, very** limited hardware
  - These devices cannot handle an excessive amount of traffic

- For example (http://csrc.nist.gov/publications/nistpubs/800-82/SP800-82-final.pdf):
  - In a factory for integrated circuits, a **ping sweep** caused a 3M robotic arm to swing out of control
  - Natural gas distribution utility, a pen tester went **a little bit** out of scope and caused a gas blockage to the customers for four (4!) hours

Can I still scan?

Yes, even Nmap can still be used and here is why:

- Reduce the scanning speed! Use "*--scan-delay=1*" to scan only one port at a time

- Do not perform ping, SYN or UDP scans, but stick to TCP scans
  - *nmap -sn -PR -n*

- No fingerprinting options, these will send a lot of extra packets to **force** the listener to respond with certain crafted packets. These often contain "fingerprints" that nmap compares with a built-in database to determine services and operating systems.

There is always another way

It is always to use the tools that are designed to work on a network ☺

→ On other words: use your head

Example of such a protocol:

SNMP!

- Simple Network Management Protocol is a technology designed to gather information on “your” network. (SNMP GET)
- It can sometimes even be used to configure certain devices (SNMP SET)

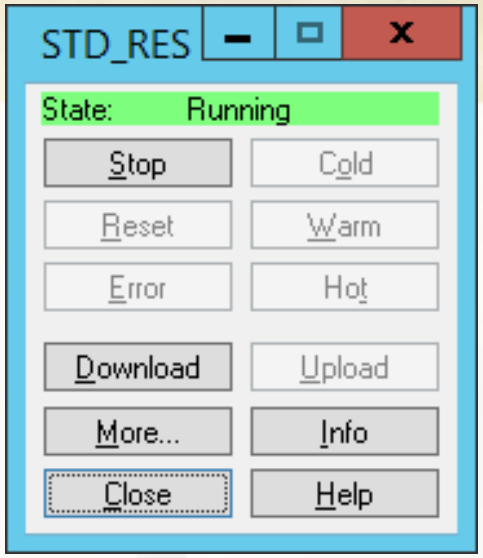
- Since devices that allow this traffic are supposed to be able to handle the load it is a more or less “safe” way of scanning a network
  - Sometimes with surprising results ☺

SNMP HowTo

- 3 versions of which SNMPv1 is too old and SNMPv3 is too new to be seen regularly.
  - SNMPv3 supports credentials, but is hardly used nor supported in older hardware

- So SNMP version 2*c* is the main focus.
  - Good and Free Windows Tool to get and set data for a given device:
    - MIB Browser (http://ireasoning.com/mibbrowser.shtml)
  - But off course extensive scanners are out there as well:
    - snmpwalk, for Windows or Linux (*apt-get install snmp*)
    - And Home Made Scripts ☺

And ICS itself?



- As seen earlier today: **No Focus On Security**
- Most common ICS protocols like ProfiNet, Modbus and S7Comm do not even support encryption and sometimes not even authentication.
- Easily confirmed: for any PLC or HMI, just download the **official** programming software (e.g. *trial*). Install it and start connecting to and controlling the PLC!

ICS security tools?

- Yes, a lot of specialized tools:
- PLCScan: http://www.digitalbond.com/tools/plcscan/
  by http://scadastrangelove.org/
  - Only scans ports 102 (Siemens S7Comm) & 502 (Modbus)
- Mbtget for ModbusTCP: https://github.com/sourceperl/mbtget
  - Stop/Run PLC's can be done with modicon_command
  - Logic down- and upload can be done with modicon_stux_transfer
- Profinet, IEC, S7-1200 scripts:
  https://github.com/atimorin/scada-tools

- Some are built into metasploit …

- And we have some of our own as well ☺

```
msf auxiliary(modbusdetect) > show info

       Name: Modbus Version Scanner
Pr
Ba
Description:
  This module detects the Modbus service, tested on a SAIA PCD1.M2
  system. Modbus is a clear text protocol used in common SCADA
  systems, developed originally as a serial-line (RS232) async
  protocol, and later transformed to IP, which is called ModbusTCP.

References:
  http://www.saia-pcd.com/en/products/plc/pcd-overview/Pages/pcd1-m2.aspx
  http://en.wikipedia.org/wiki/Modbus:TCP
```

```
msf auxiliary(modbusdetect) > use auxiliary/scanner/scada/
use auxiliary/scanner/scada/digi_addp_reboot
use auxiliary/scanner/scada/digi_addp_version
use auxiliary/scanner/scada/digi_realport_serialport_scan
use auxiliary/scanner/scada/digi_realport_version
use auxiliary/scanner/scada/indusoft_ntwebserver_fileaccess
use auxiliary/scanner/scada/koyo_login
use auxiliary/scanner/scada/modbus_findunitid
use auxiliary/scanner/scada/modbusclient
use auxiliary/scanner/scada/modbusdetect
use auxiliary/scanner/scada/sielco_winlog_fileaccess
```

All right, time to hack …euh… pentest …euh… audit!

Methodology

- Scan
  - find all IP addresses and open ports on a network
- Enumerate
  - find OS and Service versions, create connections
- Vulnerability assessment
  - use specific vulnerability scanners
- Exploit
  - gain unauthorized access to a system
- Gather
  - reuse found data/credentials on earlier services (and repeat)

Find all IP addresses and ports

Nmap (Linux/Windows) is our preferred tool here

→ Zenmap is the GUI, but you lose control on what **exactly** is happening

- E.g. *nmap -n -sP 10.0.0.0/24*
  - Use the resulting MAC addresses (if possible) to determine the device types
- E.g. *nmap -n -sT 10.0.0.4,6,10-20*
  - The port number gives you an idea what service it is
- Important: use your head
  - what would be your reaction if you see an open port **80** ?
  - what would you try if you see an open port **21** ?

Thinking out of the box (scanning)

Devices such as Routers, Switches, Printers, Access Points, Card Readers, Airconditioners, PDU's … might **also** have an IP address!

- So don't limit your scan to a single network…
- Which networks?
  - Use **tracert -d** (Windows) or **traceroute -n** (Linux)

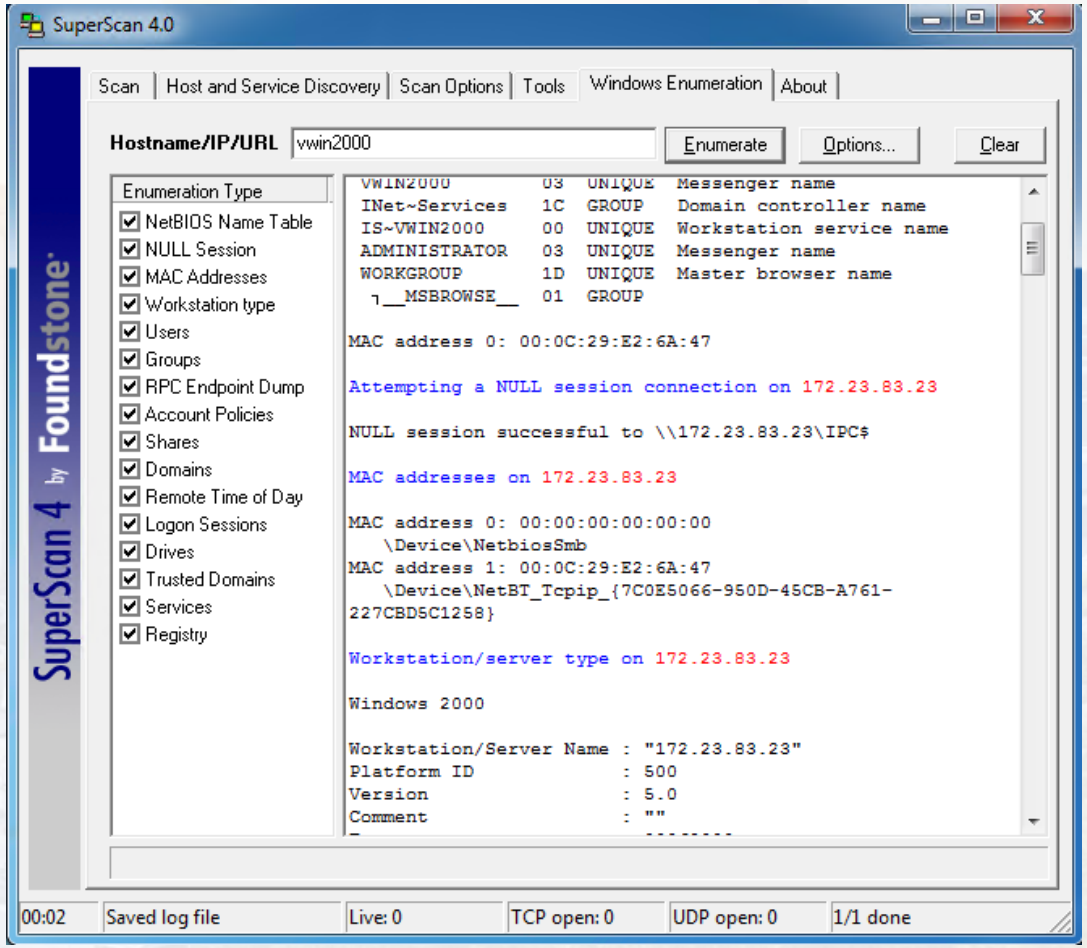
- Make sure to also including scanning the **default** network range for most of the above devices:
  - 192.168.1.0/24 and 192.168.0.0/24
  - They may be forgotten about upon installation ☺
    →E.g. Managed Switch still has IP 192.168.1.1 and default credentials, but has a lot of VLAN information

Enumerate

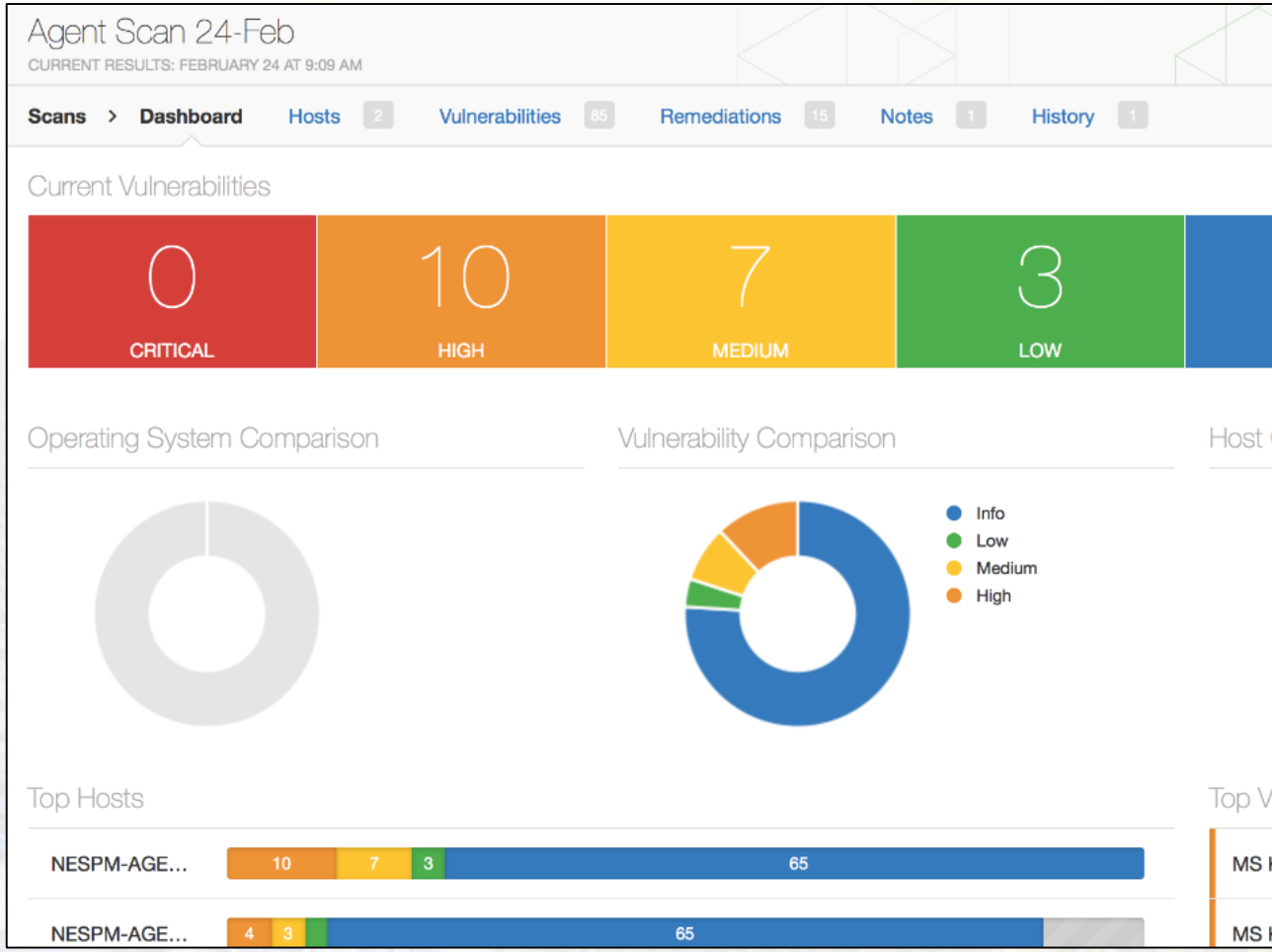
- Again, Nmap can make connections and perform basic version detection
- E.g. *nmap -n -sV -p102,3389,443,445 10.0.0.4,6,10-20*
  - This may even already disclose vulnerabilities,
    e.g. if you find SMB version 5.1.2600 (WinXPSP2)
- Can even run scripts (C:\Program Files (x86)\Nmap\scripts)
  E.g. *nmap --script=tftp-enum,vnc-info 10.0.0.1-20*
  - There are more scripts online
  - Siemens Scada: https://github.com/drainware/nmap-scada
  - ICS: https://github.com/digitalbond/Redpoint

Thinking out of the box (enumeration)

- The most common services can be enumerated
- SNMP (see earlier)
- DNS (zone transfer, anyone?)
- NetBIOS →
- Active Directory / LDAP
  - (E.g. with ridenum)
- SMTP (*nmap --script=smtp**)



Vulnerability Assessment



Finding vulnerabilities can be automated

- Tenable Nessus Vulnerability Scanner
- Has a free version (16IP limitation)
- Some (ICS) companies run Nessus on a weekly base

- Delivers beautiful reports
- Can be configured with scan delays
- Detailed information on vulnerabilities and potential exploitability

Thinking out of the box (vulnerabilities)

- There are also **protocol vulnerabilities**:
  - DNS
  - HTTP
  - TELNET
  - FTP
  - SMTP
  - …
- … all have one thing in common: they're **cleartext** protocols!
- Which means: they can be MiTM'ed and sniffed easily
  - Windows-tool: Cain & Abel, Linux-tool: ettercap

Exploiting

metasploit®

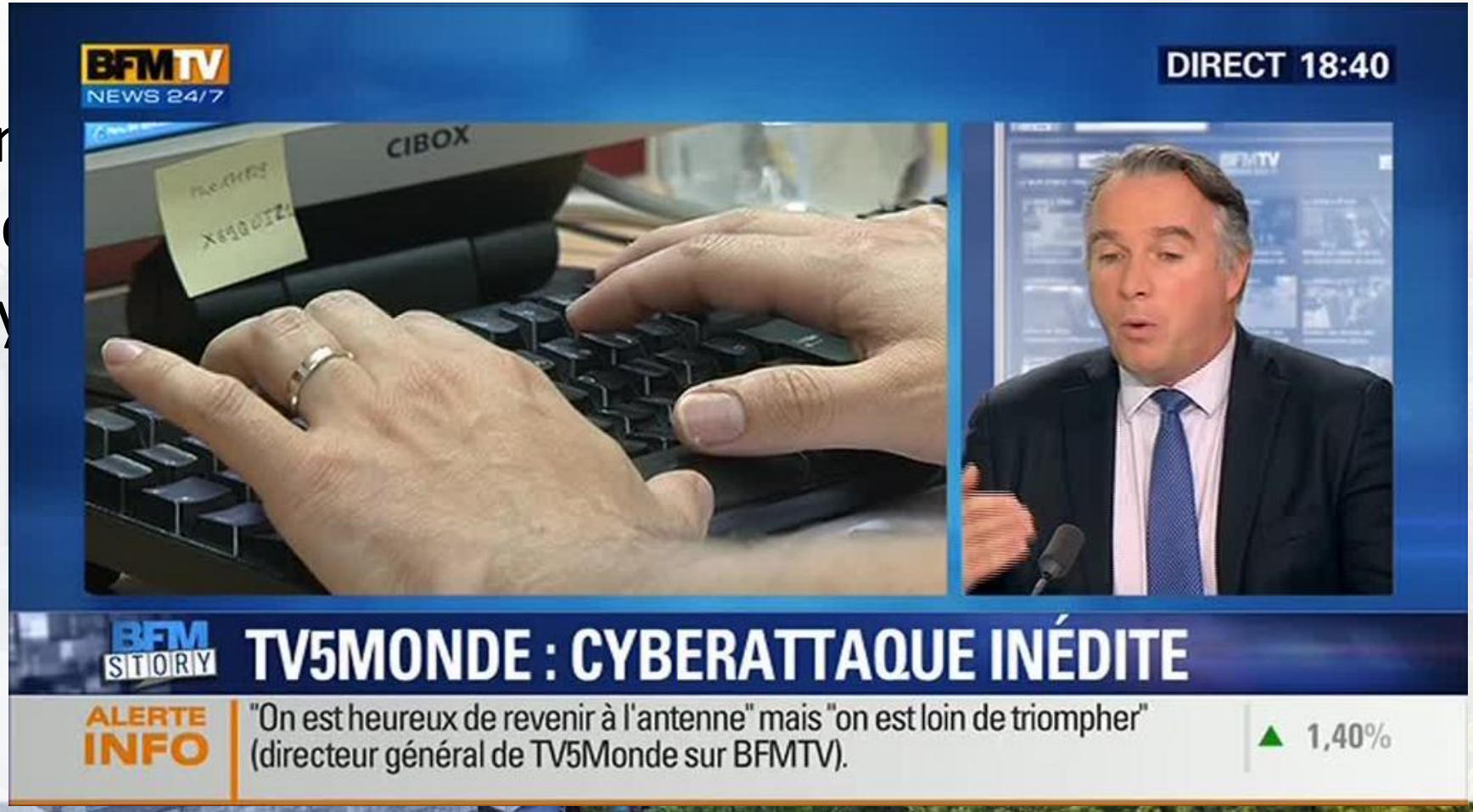
- Now is the time to send special crafted packets to certain devices or portals
  - SQL Injection on websites?
  - Buffer Overflow on Windows SMB?
  - Profinet Set on PLC?

- There is a good tool for that: the ***Metasploit Framework***
  - Has a community edition (free), and exists for both Windows (buggy) and Linux (better)
  - Has a lot of built in exploits (currently around 1540), scanners and modules
- There are also online exploit databases like www.exploit-db.com

Thinking out of the box (exploitation)

- Just logging into a device without permission is also considered exploitation

- Not only devices car
  (and they often are
  - Dropping USB Key
  - Phishing emails
  - Shoulder surfing
  - Or - - - - - - - - - - >



Gather

- The main goal for most

```
root@kalitijl:~/Desktop# ls /usr/share/metasploit-framework/modules/post/windows
/gather/
arp_scanner.rb                  enum_ms_product_keys.rb
bitcoin_jacker.rb               enum_muicache.rb
bitlocker_fvek.rb               enum_patches.rb
cachedump.rb                    enum_powershell_env.rb
```

```
root@kalitijl:~/Desktop# ls /usr/share/metasploit-framework/modules/post/windows
/gather/credentials/
bulletproof_ftp.rb          gpp.rb                      skype.rb
coreftp.rb                  idm.rb                      smartermail.rb
credential_collector.rb     imail.rb                    smartftp.rb
domain_hashdump.rb          imvu.rb                     spark_im.rb
dyndns.rb                   mcafee_vse_hashdump.rb      sso.rb
enum_cred_store.rb          meebo.rb                    steam.rb
enum_laps.rb                mremote.rb                  tortoisesvn.rb
enum_picasa_pwds.rb         mssql_local_hashdump.rb     total_commander.rb
epo_sql.rb                  nimbuzz.rb                  trillian.rb
filezilla_server.rb         outlook.rb                  vnc.rb
flashfxp.rb                 razer_synapse.rb            windows_autologin.rb
ftpnavigator.rb             razorsql.rb                 winscp.rb
ftpx.rb                     rdc_manager_creds.rb        wsftp_client.rb
```

```
enum_hostfile.rb            win_privs.rb
enum_ie.rb                  wmic_command.rb
enum_logged_on_users.rb     word_unc_injector.rb
```

Thinking out of the box (gather)

- Just looking through files, browser history and stored passwords is one thing the aforementioned modules can assist with
- However nothing beats a manual search through some text files (*password.txt* on the desktop?), installing keyloggers or going for software like KeePass.
- And then there is also the PC's **Memory**
  - Data stored inside PC memory is generally not encrypted
  - It is a default setting for Windows to store all Kerberos and Windows local passwords in memory for future authentication (and backwards compatibility)
  - Mimikatz is a software that can extract these secrets
- Don't underestimate the importance of password hashes

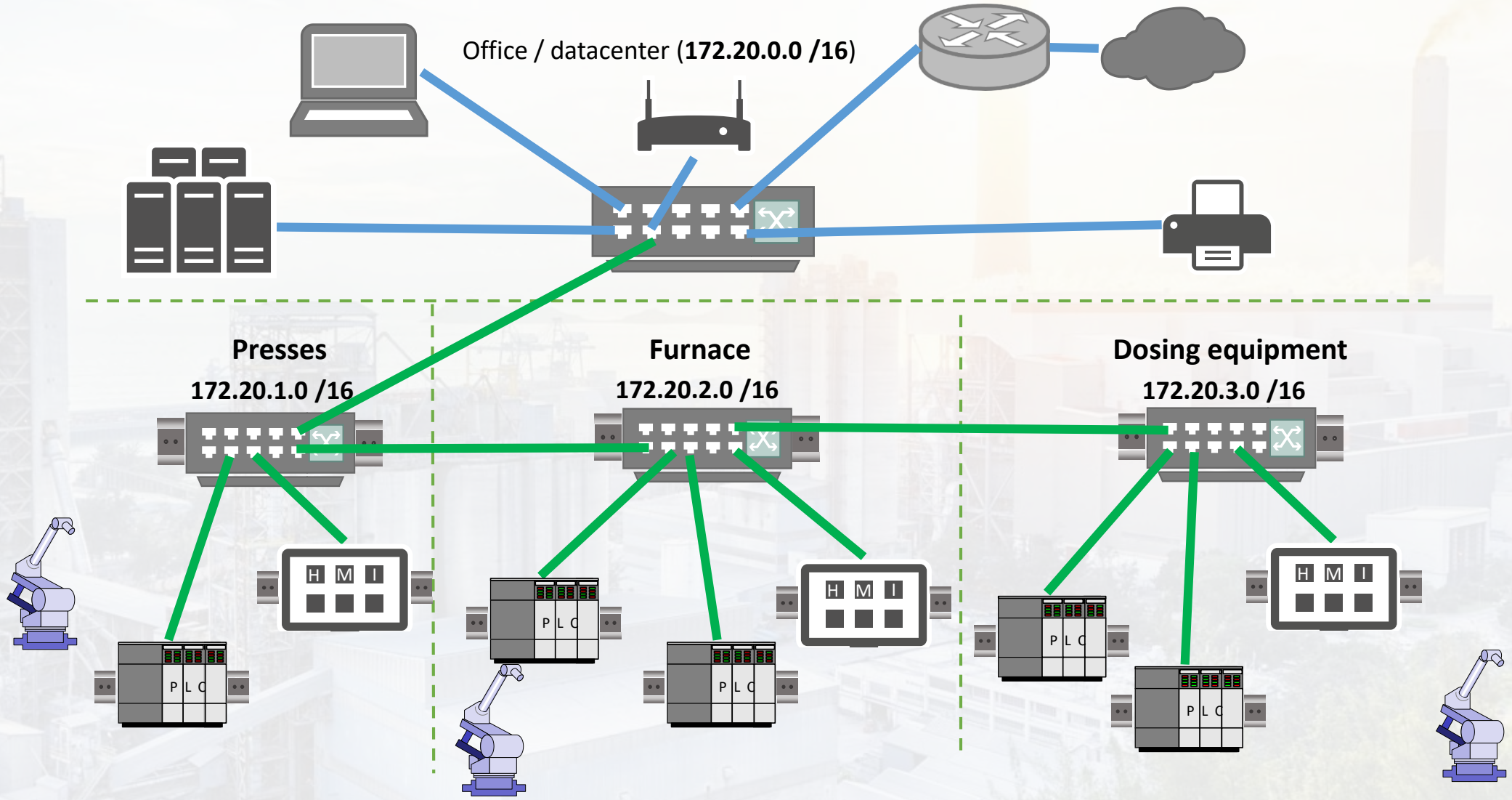
The Workshop

- Perform your own scanning on our *special* demo network.
  - Nothing is off limits, we have everything backed up
  - Learn to use Nmap, in Windows or Linux, maybe extended with extra (ICS) scripts
  - Feel free to try enumeration (recommended), vulnerability scanning and maybe even exploitation

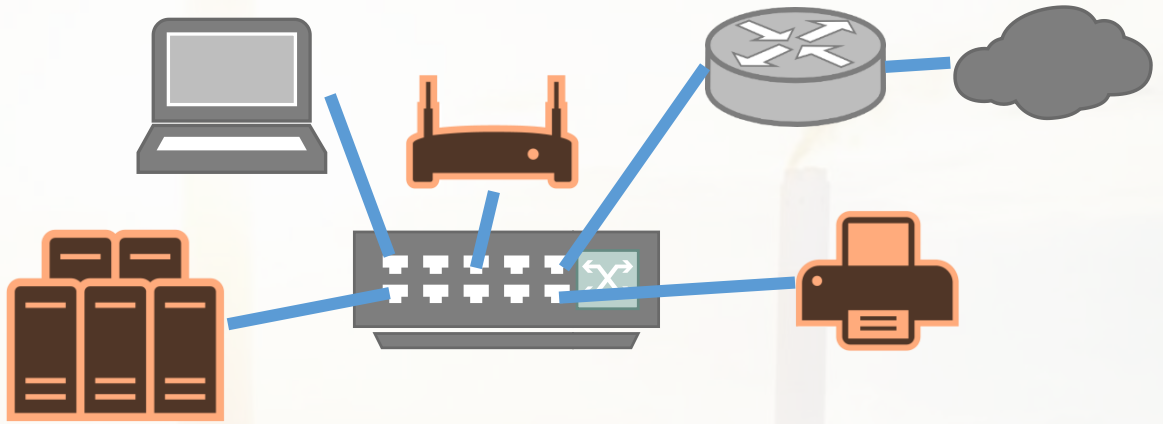
- Inside this network we have also set up several Siemens S7-1200 PLC's connected to a functional setup, on every table.
  - There are several exercises, all explained on the local posters

Introducing our Factory

Fictile: the network

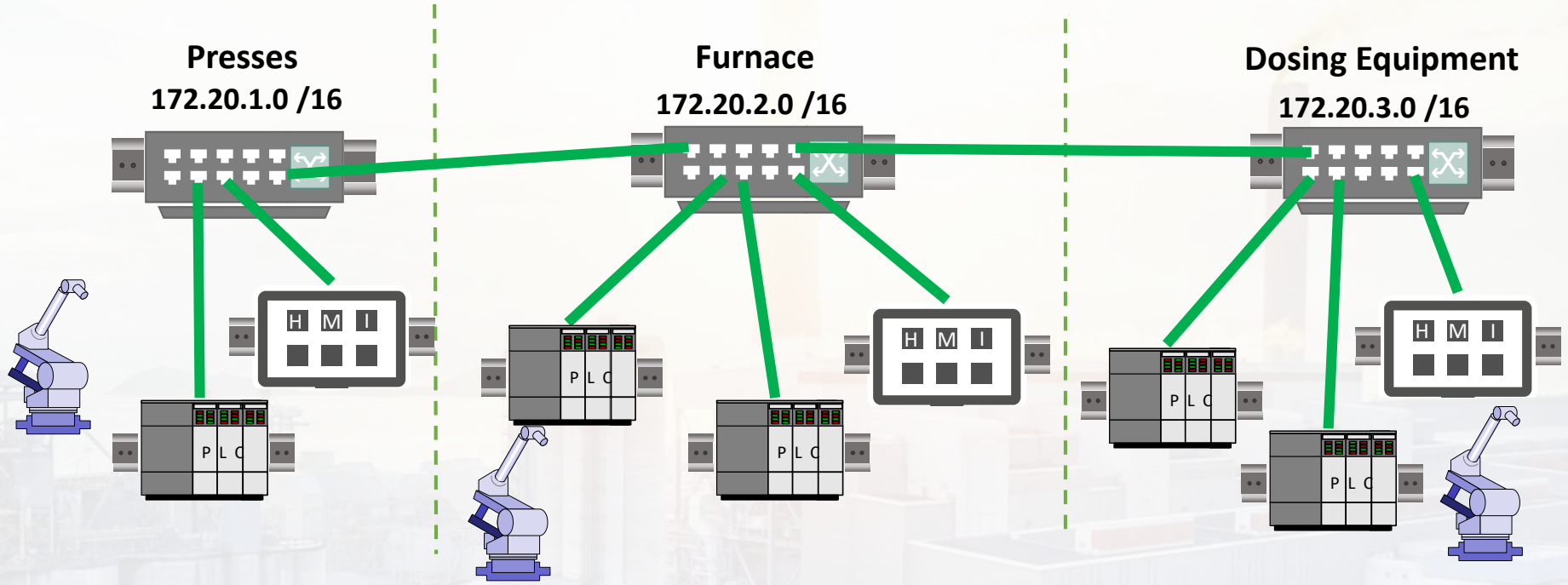


What can be done?



- Everything off course ☺
- The AD Environment is completely vulnerable
  - Don’t go directly for the Domain Controller, but it is the ultimate goal
  - Find the weak spot!
- Can you print any text on the network printer? (One command!)
- There is also a vulnerability on the WiFi AP (on the LAN side)

And ICS Hacking?



Also everything

- For PhoenixContact; there are misconfigurations and vulnerabilities
  - No special scripts needed for some of these
- Beckhoff has several vulnerabilities in both the HMI as the PLC
  - Special scripts may be needed but are available
- Siemens has the most common configuration