

# Fun with Modbus

## Ox5a

---

Nothing new. Still relevant ?

Arnaud SOULLIÉ, Wavestone

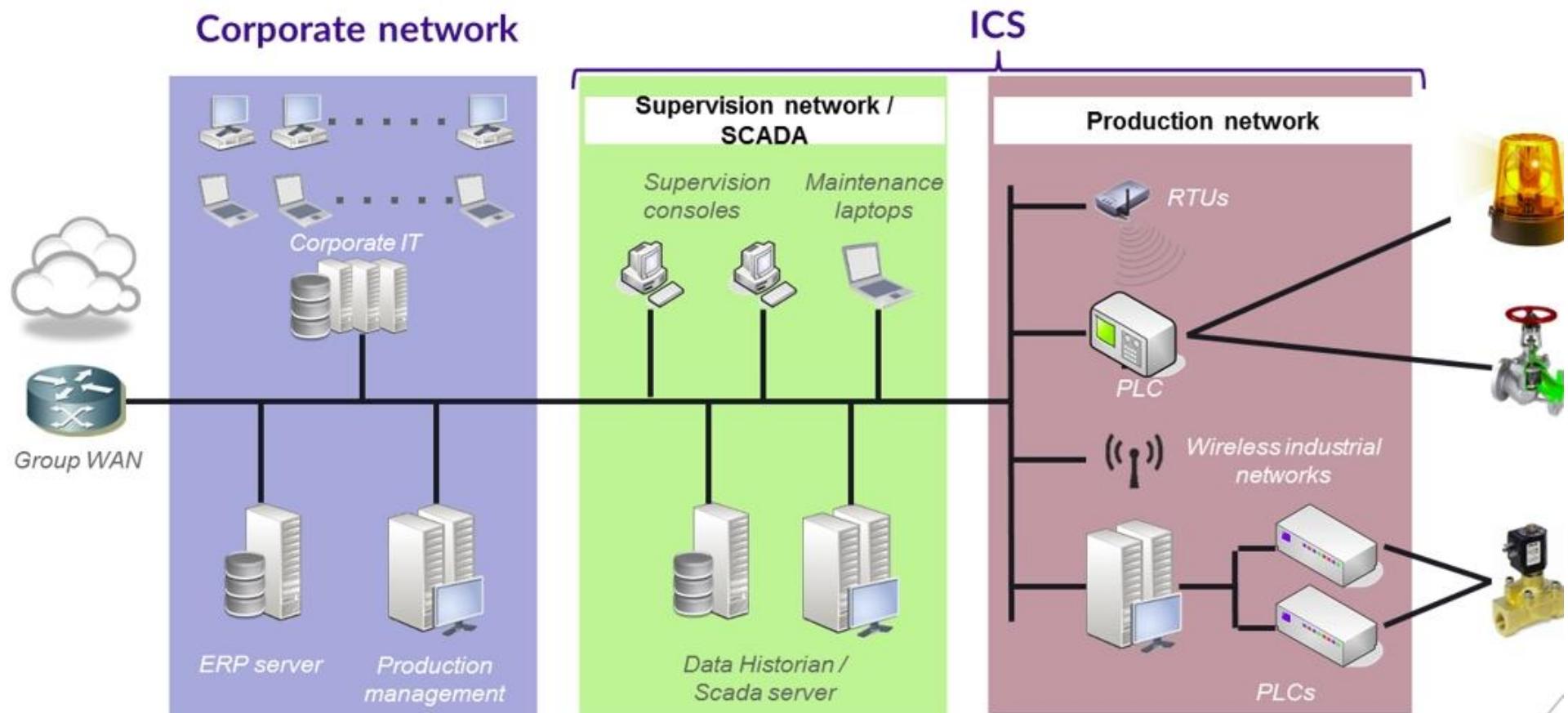
# About me

Arnaud SOULLIÉ

- Pentester, and some R&D
- Manager at Wavestone  
*(but I don't manage anyone)*
- Love ICS. Love ICS.
- Give ICS pentesting workshops at cons (*BlackHat Europe, BruCon, DEFCON, BSides Las Vegas, ...*)
- Also like Active Directory. And Wine. And trail running. And hiking. And lots of stuff you probably do not care about.

# If you don't know anything about ICS...

No worries ! In a nutshell:



# Modbus

Invented in 1979 by Schneider Electric (*Modicon* at the time)

One the most widely used protocol in ICS

Designed to work over RS-432 and RS-485 serial lines

Binary (RTU) or ASCII data representation

ADU: <i>Application Data Unit</i>			
Address	PDU: <i>Protocol Data Unit</i>		
	Function code	Data	Error Check

*Modbus frame format*

⇒ **no encryption, no authentication**

# Modbus/TCP

Basically, Modbus on top of TCP

Error detection is handled by underlying protocols

ADU: Application Data Unit					
MBAP: Modbus Application Protocol			PDU: Protocol Data Unit		
Transaction identifier	Protocol identifier	Length	Unit identifier	Function code	Data

Modbus TCP frame format

Common function codes:

- Read Discrete Inputs (2)
- Read Coils (1)
- Write Single Coil (5)
- Write Multiple Coils (15)
- Read Input Registers (4)
- Read Multiple Holding Registers (3)
- Write Single Holding Register (6)
- Write Multiple Holding Registers (16)

⇒ still no security

# More than coils & registers

Some Modbus functions are also designated for diagnostic purposes, eg :

Read Device Information (43)

**43/14 Read Device Identification**

Slave ID:

Code:

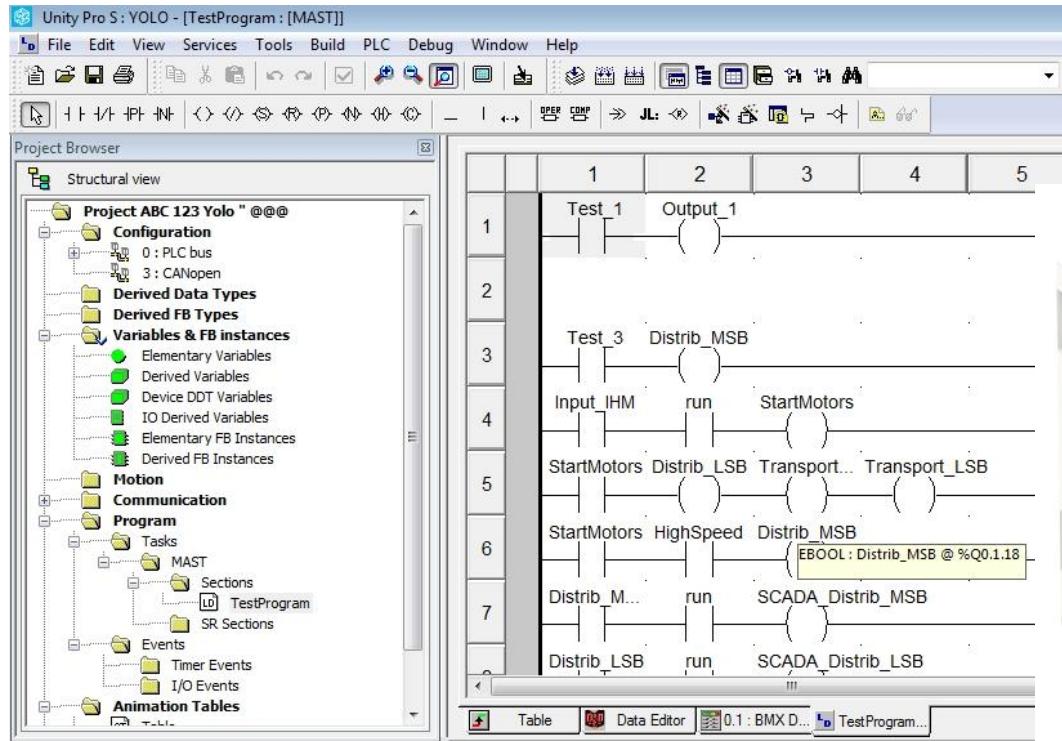
First Object ID:

Received Data:

ID	Name	Value
00	Vendor Name	Schneider Electric
01	Product Code	BMX P34 20302
02	Major/Minor Revision	v2.5

# Schneider PLCs

Quantum, Premium, M340, TM2XX, M580



What happens when you program a PLC ?  
Wireshark to the rescue !

# Wireshark packet dump

Here's a packet dump when uploading a new program to a Schneider PLC :

23 Query: Trans: 19690; Unit: 1, Func: 90: Unity (Schneider)  
25 Response: Trans: 19690; Unit: 1, Func: 90: Unity (Schneider)  
26 Query: Trans: 19691; Unit: 1, Func: 90: Unity (Schneider)  
28 Response: Trans: 19691; Unit: 1, Func: 90: Unity (Schneider)  
29 Query: Trans: 19692; Unit: 1, Func: 90: Unity (Schneider)  
31 Response: Trans: 19692; Unit: 1, Func: 90: Unity (Schneider)

► Transmission Control Protocol, Src Port: 49186, Dst Port: 502,  
► Modbus/TCP  
▼ Modbus  
    .101 1010 = Function Code: Unity (Schneider) (90)  
    Data: 000100

0000 00 04 00 01 00 06 08 00 27 85 c5 cd 00 00 08 00 .  
0010 45 00 00 33 05 aa 40 00 80 06 73 19 c0 a8 00 7b E..3..@  
0020 c0 a8 00 36 c0 22 01 f6 c8 c9 89 87 a6 20 00 01 ...6..  
0030 50 18 04 1f 20 cb 00 00 4c ea 00 00 00 05 01 5a P...  
0040 00 01 00

Runs on top of ModbusTCP

Uses the undocumented 90 function code

Already in Wireshark dissector ?

## Previous work

### Demo: Uploading Rogue Ladder Logic to a Modicon PLC

May 2, 2012 by [Dale G Peterson](#) — [1 Comment](#)

Everything started with @reverseics work.

# So, what evil things could we do?

Gather some information about the PLC and the project



START/STOP the PLC



Alter the ladder logic

Force outputs



Hotpatch the PLC

# What about the newer models ?

TM221: New entry-level model released in 2014

Still using the same protocol, but with a few differences so existing exploits do not work out of the box.

**START/STOP** has been ported for this PLC by  
Alexandrine TORRENTS

Forcing output values should soon work as well.



No.	Length	Info
23	67	Query: Trans: 19690; Unit: 1, Func: 90: Unity (Schneider)
25	79	Response: Trans: 19690; Unit: 1, Func: 90: Unity (Schneider)
26	66	Query: Trans: 19691; Unit: 1, Func: 90: Unity (Schneider)
28	66	Response: Trans: 19691; Unit: 1, Func: 90: Unity (Schneider)
29	102	Query: Trans: 19692; Unit: 1, Func: 90: Unity (Schneider)
31	67	Response: Trans: 19692; Unit: 1, Func: 90: Unity (Schneider)
32	66	Query: Trans: 19693; Unit: 1, Func: 90: Unity (Schneider)
34	66	Response: Trans: 19693; Unit: 1, Func: 90: Unity (Schneider)
35	66	Query: Trans: 19694; Unit: 1, Func: 90: Unity (Schneider)
37	108	Response: Trans: 19694; Unit: 1, Func: 90: Unity (Schneider)

# What about the newer models ?

M580 PLC adds IPSEC authentication between engineering workstation and PLC.



# What do we do now ?

PLC Hardening ?

⇒ Not possible as far as I know

Use newer PLCs ?

⇒ Only the latest high-end Schneider PLCs implement security

⇒ have them pentested

NSM

⇒ There are signatures for Modbus fct 90, use them

# Summary

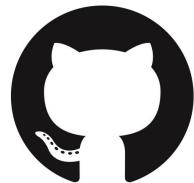
Most Schneider PLCs, including those released as late as 2014, offer no mechanism to prevent unauthorized access.

Attackers can leverage this lack of security to influence the physical process handled by the PLC.

Forcing the electrical outputs might allow to influence the process without having to reprogram the PLC and without any impact on the HMIs.



# Any question ?



<https://github.com/arnaudsoullie/funwithmodbus0x5a>

arnaudsoullie / **funwithmodbus0x5a**

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

Material from my ICS Village talk at DEFCON 25 Edit

ics schneider modbus modbus-tcp Manage topics

2 commits 1 branch 0 releases 1 contributor GPL-3.0

Branch: master New pull request Create new file Upload files Find file Clone or download

arnaudsoullie	Initial commit before the talk.	Latest commit 4ba6a0f 10 minutes ago
LICENSE	Initial commit	29 minutes ago
README.md	Initial commit before the talk.	10 minutes ago
modicon_command_CTv2.rb	Initial commit before the talk.	10 minutes ago
modicon_stux_transfer_ASO.rb	Initial commit before the talk.	10 minutes ago
schneider.rb	Initial commit before the talk.	10 minutes ago

[arnaud.soullie@wavestone.com](mailto:arnaud.soullie@wavestone.com)  
@arnaudsoullie