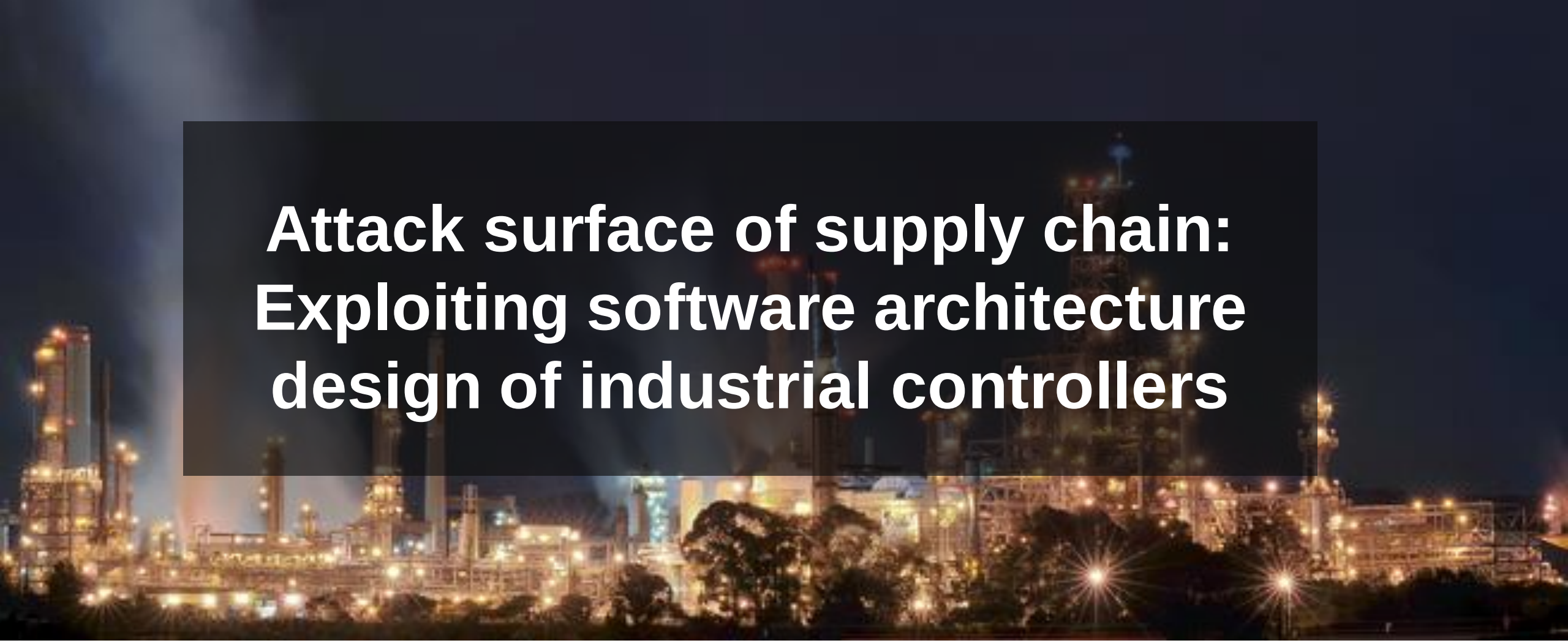


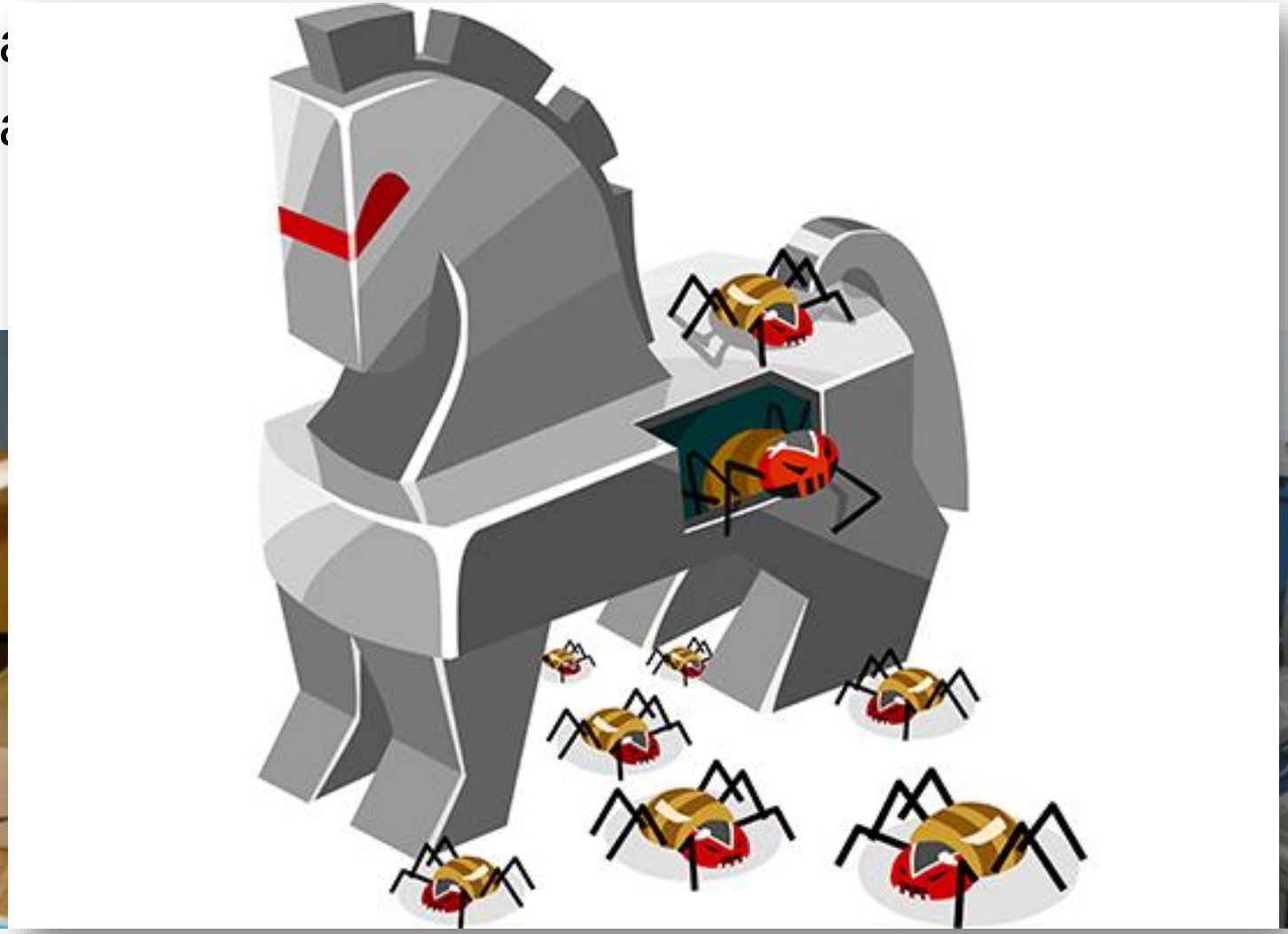
Attack surface of supply chain: Exploiting software architecture design of industrial controllers

Marina Krotofil

Swiss Cyber Storm
Bern, Switzerland
12.10.2021

Supply chain security

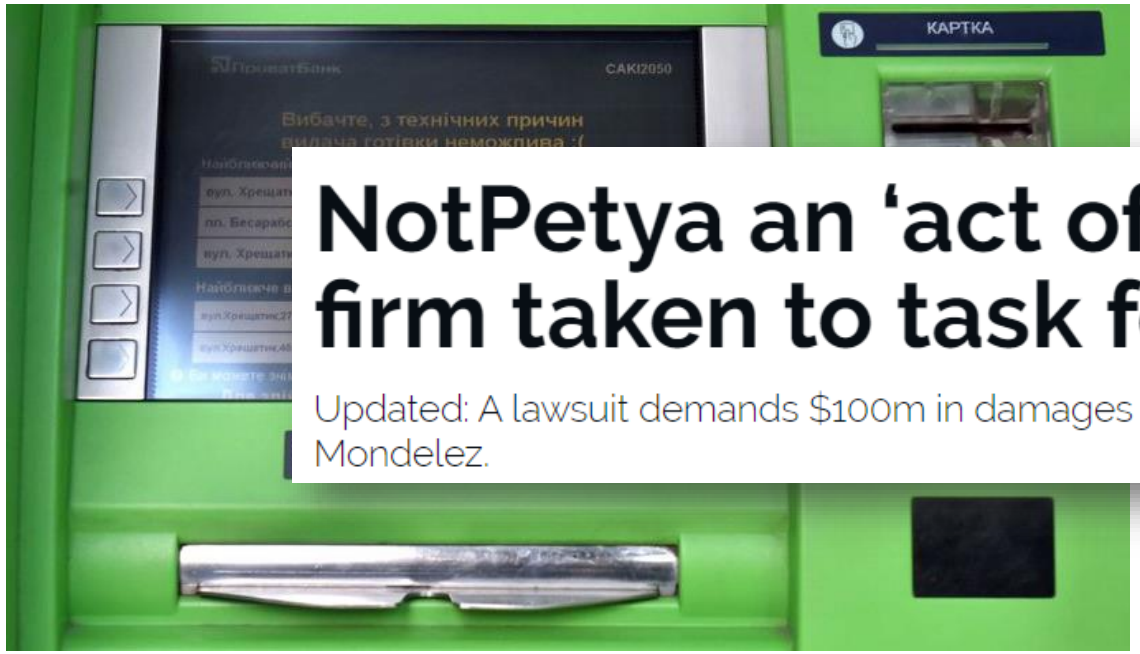
- Has been an a
- Has been an a



https://theintercept.com/2019/01/24/computer-supply-chain-attacks/

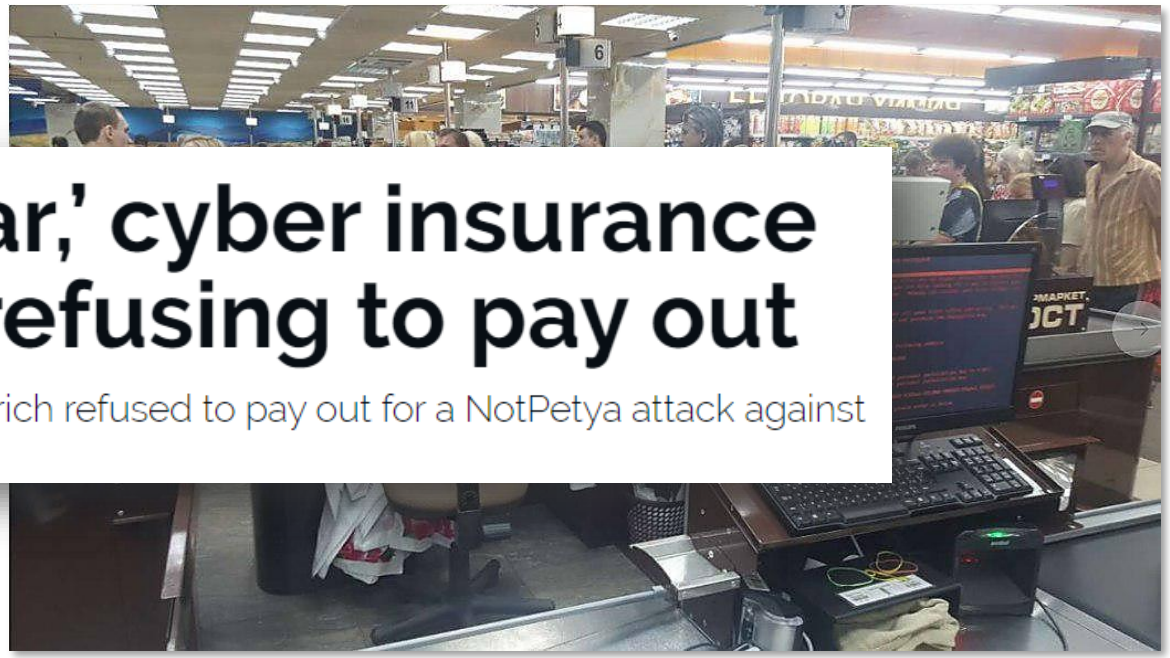
Most devastating supply chain attack

- NotPetya attack in Ukraine, June 27 2017 (Constitution day)
- An update for MeDoc tax software was pushed out by the update server
  - All vital functions in the whole country were paralyzed in less than 24hrs



NotPetya an 'act of war,' cyber insurance firm taken to task for refusing to pay out

Updated: A lawsuit demands $100m in damages after Zurich refused to pay out for a NotPetya attack against Mondelez.

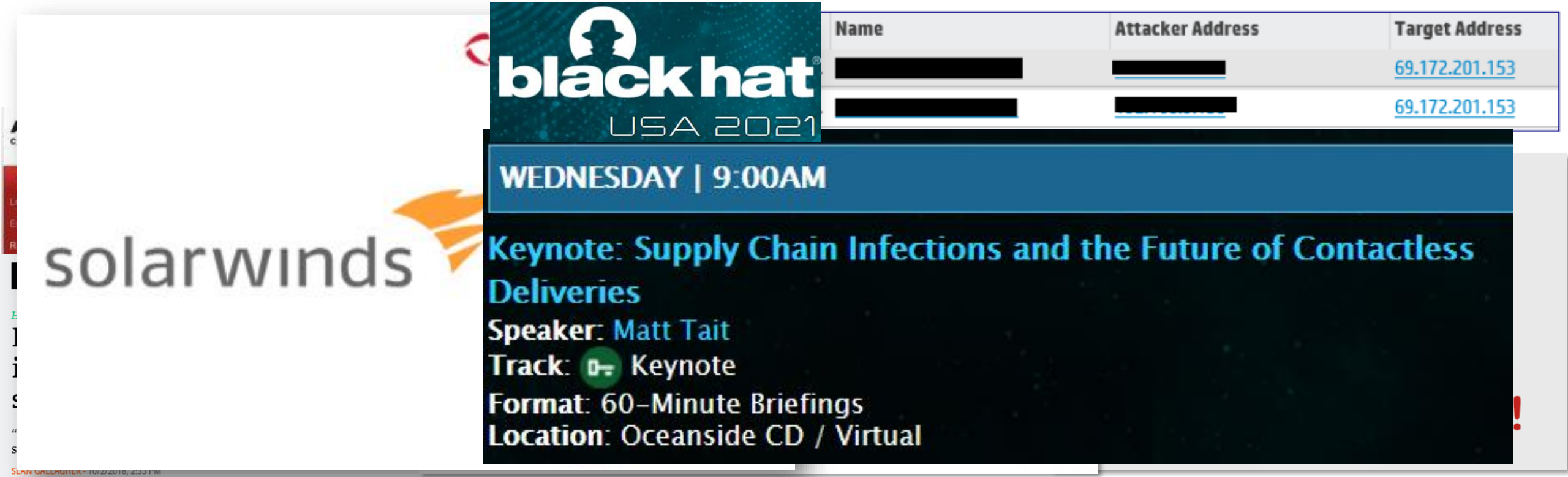


https://www.bbc.com/future/article/20170704-the-day-a-mysterious-cyber-attack-crippled-ukraine

https://twitter.com/MaximEristavi/status/879712719535996928

Most recent supply chain attacks

- Complexity and impact of supply chain attacks are increasing
- Mostly state-sponsored level of attack vector: both execution & management



Two sides of a coin

Supply Chain Security

“Attack delivery” vector

“Attack surface” vector

HUAWEI

KASPERSKY lab

https://threatpost.com/wipro-confirms-hack/143826/

https://www.csoonline.com/article/3543838/cisco-and-palo-alto-networks-appliances-impacted-by-kerberos-authentication-bypass.html

Two sides of a coin

Supply Chain Security

“Attack delivery” vector

“Attack surface” vector

Wipro Confirms Hack and Supply Chain Attacks on Customers

https://threatpost.com/wipro-confirms-hack/143826/

Cisco and Palo Alto Networks appliances impacted by Kerberos authentication bypass

https://www.csoonline.com/article/3543838/cisco-and-palo-alto-networks-appliances-impacted-by-kerberos-authentication-bypass.html

Two sides of a coin

Supply Chain Security

“Attack delivery” vector

“Attack surface” vector

Software Bill of Materials Elements and Considerations

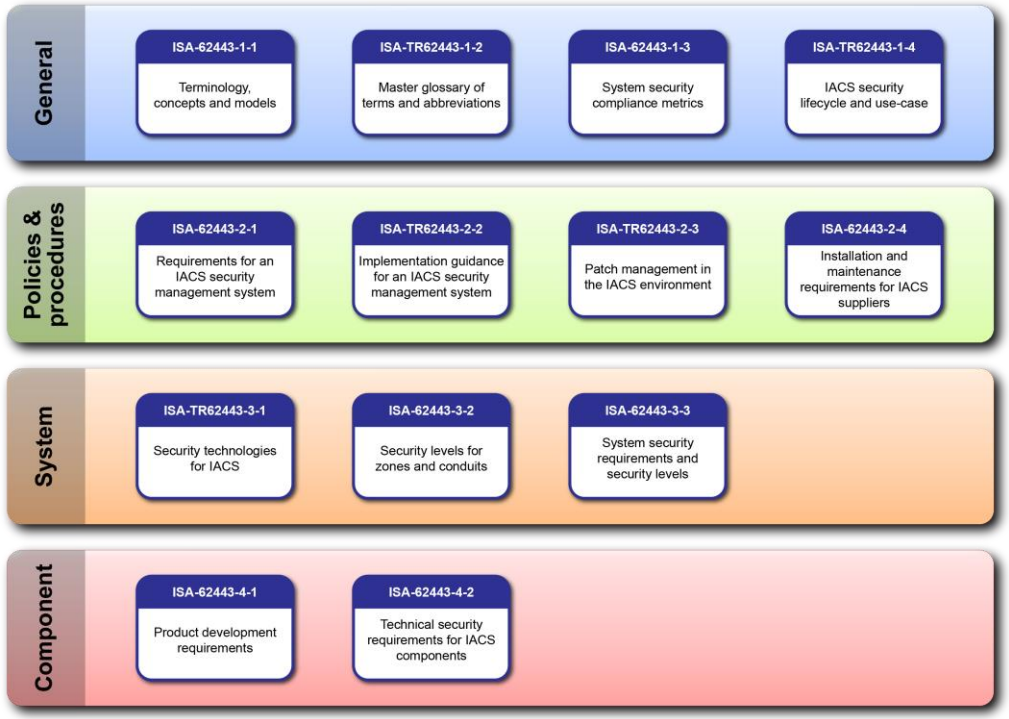
A Notice by the National Telecommunications and Information Administration on 06/02/2021

https://threatpost.com/wipro-confirms-hack/143826/

https://www.federalregister.gov/documents/2021/06/02/2021-11592/software-bill-of-materials-elements-and-considerations

Supply chain security in OT/ICS/CI

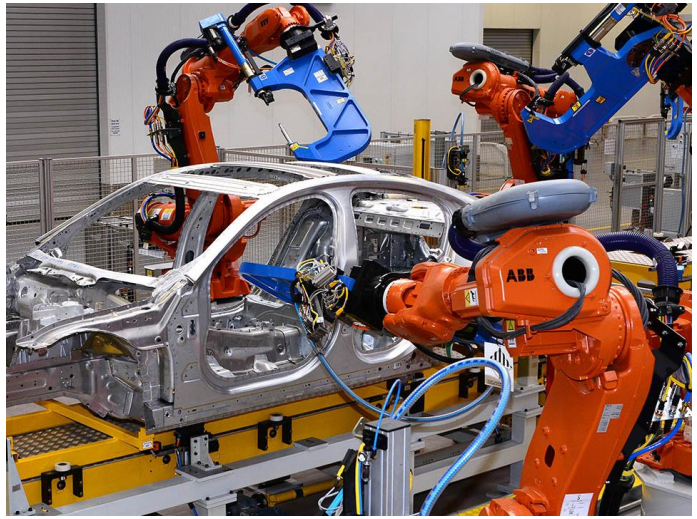
- **IEC 62443** is international series of standards which specifies comprehensive requirements for the secure development, integration and maintenance of assets used in Industrial **Automation & Control Systems (IACS)**environments
- Targets at:
  - **Vendor**
  - Integrator
  - Asset owner



Examples of industrial controllers

https://vecer.mk/files/article/2017/05/02/485749-saudiska-arabija-ja-kupi-najgolemata-naftena-rafinerija-vo-sad.jpg

http://www.jfwhite.com/CollateralImages/English-US/Galleries/middleboro91 15kv breakers. jpg



https://www.roboticsbusinessr eview.com/wp-content/uploads/2016/05/jaguar-factory.jpg

https://www.oilandgasproduct news.com/files/slides/locale_image/medium/0089/22183_en_16f9d_8738_honeywell-process-solutions-rtu2020-process-controller. jpg

https://selinc.com/uploadedImages/Web/Videos/Playlists/Playlist_RTAC_1280x720.png?n=635847581260000

http://www02.abb.com/global/seitp/seitp202.nsf/0/06 01d25ed243cfb0c1257d7e0043e50e/$file/7184_lvl2.jpg

Device security vector

SL = (Identification & Authentication, Control, Use control, System integrity, Data confidentiality, Restricted data flow, Timely response to events, Resource availability) = (2, 2, 0, 1, 3, 1, 3)

Security Level	Target	Skills	Motivation
SL1	Casual or coincidental violations	No Attack Skills	Mistakes
SL2	Cybercrime, Hacker	Generic	Low
SL3	Hacktivist, Terrorist	ICS Specific	Moderate
SL4	Nation State	ICS Specific	High

Foundational Requirements (FR)

Security certification of industrial assets

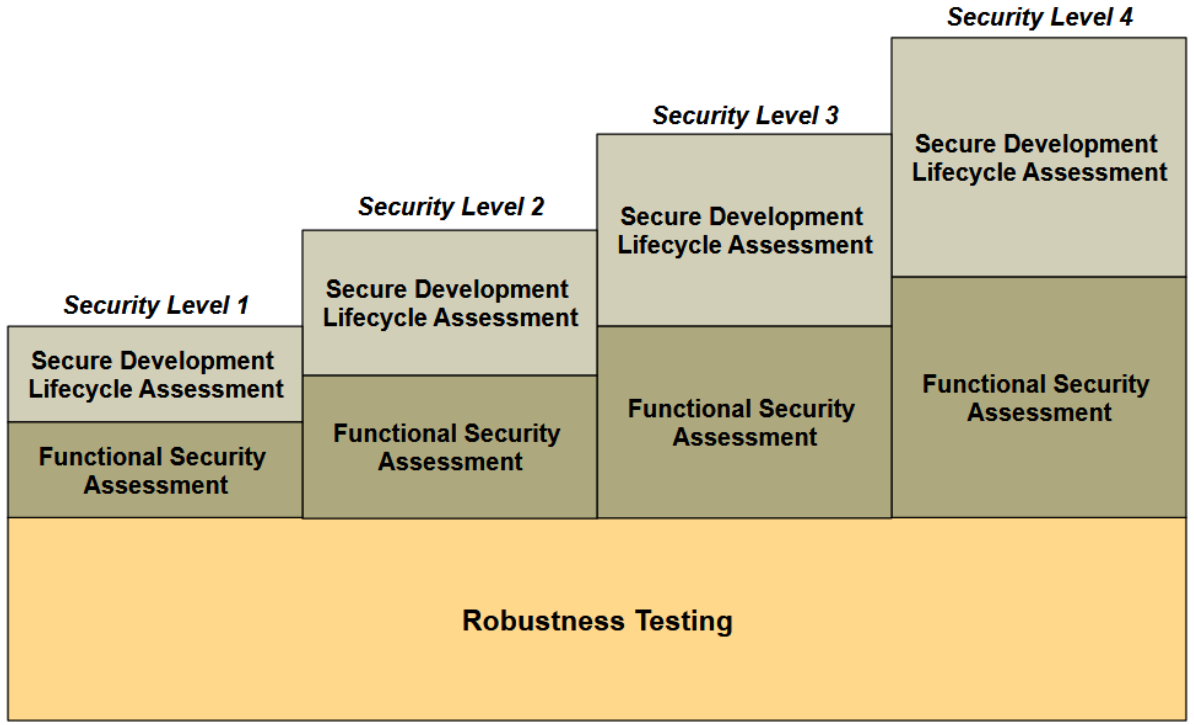
- Certification was developed to attest that devices meet IEC-62443 requirements:
  - "*Asset owners have confidence that the IACS products they purchase are robust against network attacks and are free from known security vulnerabilities*"
- Most commonly certified:
  - Security Development Lifecycle Assurance Program (SDLA)
  - Embedded Device Security Assurance Program (EDSA)

	Honeywell Process Solutions	DCS Controller	Experion C300	R430	EDSA 2010.1 Level 1	10/27/2016
	Honeywell Process Solutions	PLC	ControlEdge PLC	R140	EDSA 2.0.0 Level 2	7/3/2017

https://www.isasecure.org/en-US/End-Users/ISASecure-Certified-Devices

Security certification efforts

- Is mostly about functional testing
- Long hanging fruits things



Typical Chartered Lab Level of Effort in Man Weeks

	Level 1	Level 2	Level 3
1. CRT test all accessible TCP/IP interfaces	1 - 2 weeks	1 - 2 weeks	1 - 2 weeks
2. Perform FSA on device and all interfaces	< 1 week	1 week	1 – 2 weeks
3. Audit supplier's software development process	1 week	1 – 2 weeks	1 – 2 weeks
4. Perform ITA and issue report	1 week	1 week	1 week
	3 – 5 weeks	4 – 6 weeks	4 – 10 weeks

Vulnerabilities in device supply chain

- Urgent/11 (July 2019)
- Ripple20 (June 2020)
- Amnesia:33 (December 2020)

Black Hat talks

Attack surface is not evaluated

- Authentications schemes in industrial PLCs are regularly broken by (not very advanced) researchers

Empirical Study of PLC Authentication Protocols in Industrial Control Systems

Adeen Ayub
Department of Computer Science
Virginia Commonwealth University
Richmond, United States of America
ayuba2@vcu.edu

Hyunguk Yoo
Department of Computer Science
The University of New Orleans
New Orleans, United States of America
hyoo1@uno.edu

Irfan Ahmed
Department of Computer Science
Virginia Commonwealth University
Richmond, United States of America
iahmed3@vcu.edu

https://ieeexplore.ieee.org/document/9474296

Rogue7: Rogue Engineering-Station attacks on S7 Simatic PLCs

Eli Biham[1] Sara Bitan[1] Aviad Carmel[1] Alon Dankner[1] Uriel Malin[2]
Avishai Wool[2]

https://www.blackhat.com/us-19/briefings/schedule/index.html#rogue-rogue-engineering-station-attacks-on-s-simatic-plcs-16049

PLC Access Control: A Security Analysis

Haroon Wardak
Information and Computer Science Department
KFUPM, Dhahran, 31261, KSA
Email: g201302150@kfupm.edu.sa

Sami Zhioua
Information and Computer Science Department
KFUPM, Dhahran, 31261, KSA
Email: zhioua@kfupm.edu.sa

Ahmad Almulhem
Computer Engineering Department
KFUPM, Dhahran, 31261, KSA
Email: ahmadsm@kfupm.edu.sa

https://ieeexplore.ieee.org/document/7882935

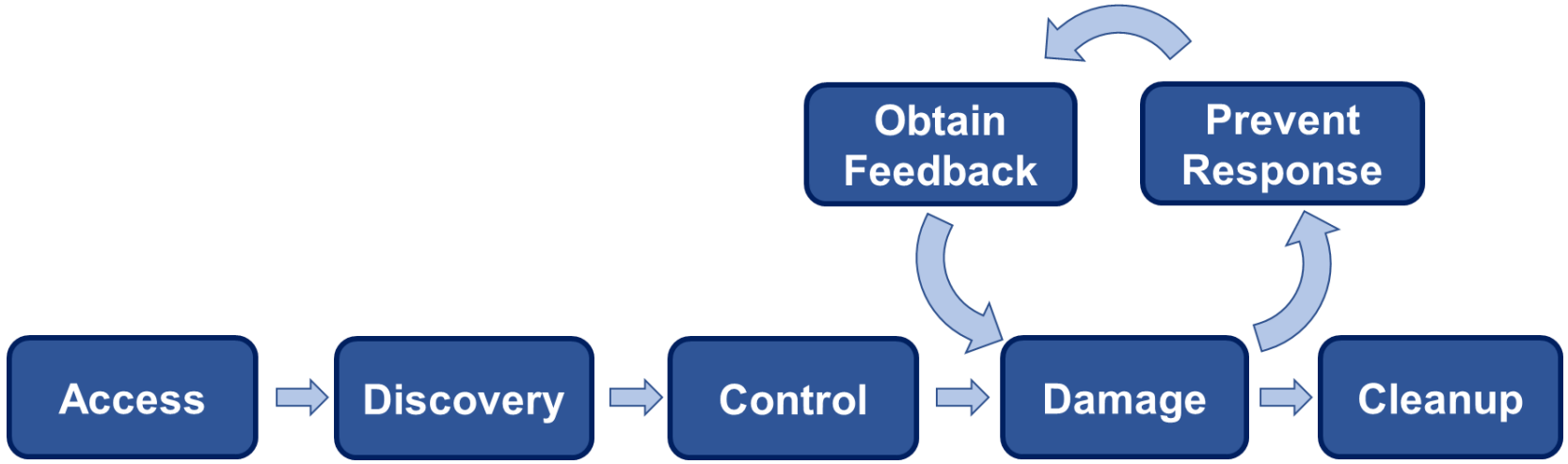
What‘s my interest in device attack surface?

- **Research specialization:** Offensive cyber-physical security in Critical Infrastructures

Focus:

- Physical damage or how to make something going bad, wrong, crash or blow up by means of cyber-attacks

Cyber-physical attack lifecycle



Using asset design for attacker needs

- Assist with attack activities, e.g. reconnaissance
- Exploit asset designs for attack execution

A Rising Tide: Design Exploits in Industrial Control Systems

Alexander Bolshev
IOActive, Inc.
Madrid, Spain

Jason Larsen
IOActive, Inc.
Seattle, WA 98104, USA

Marina Krotofil
Honeywell
Duluth, GA 30097, USA

Reid Wightman
Digital Bond
Indianapolis, IN 46220 USA

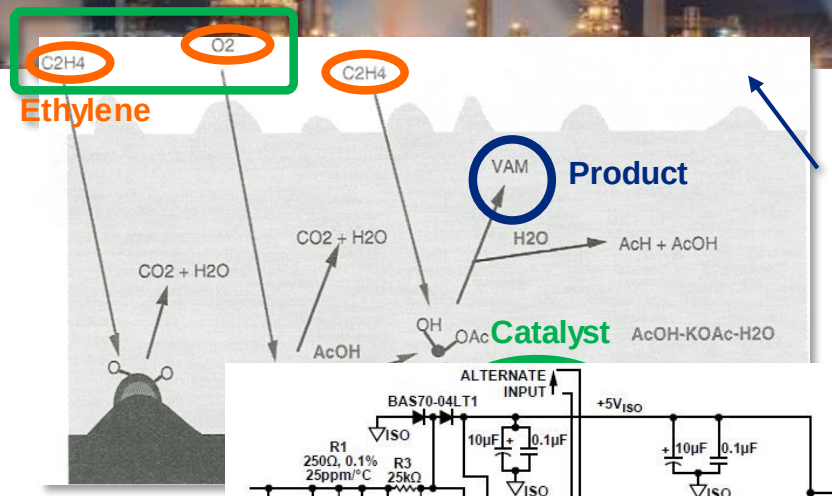
On the Significance of Process Comprehension for Conducting Targeted ICS Attacks

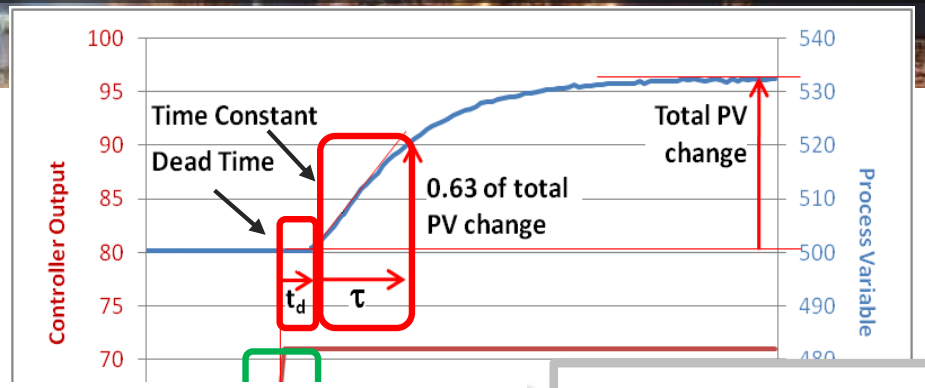
Benjamin Green
Lancaster University
Lancaster, United Kingdom
b.green2@lancaster.ac.uk

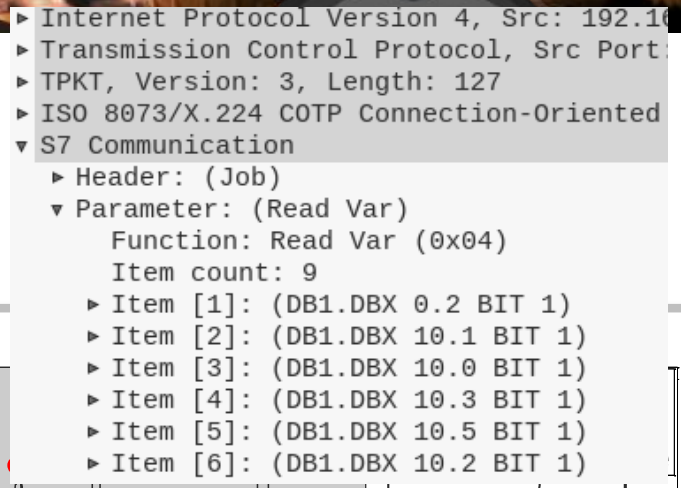
Marina Krotofil
Hamburg University of Technology
Hamburg, Germany
marina.krotofil@tuhh.de

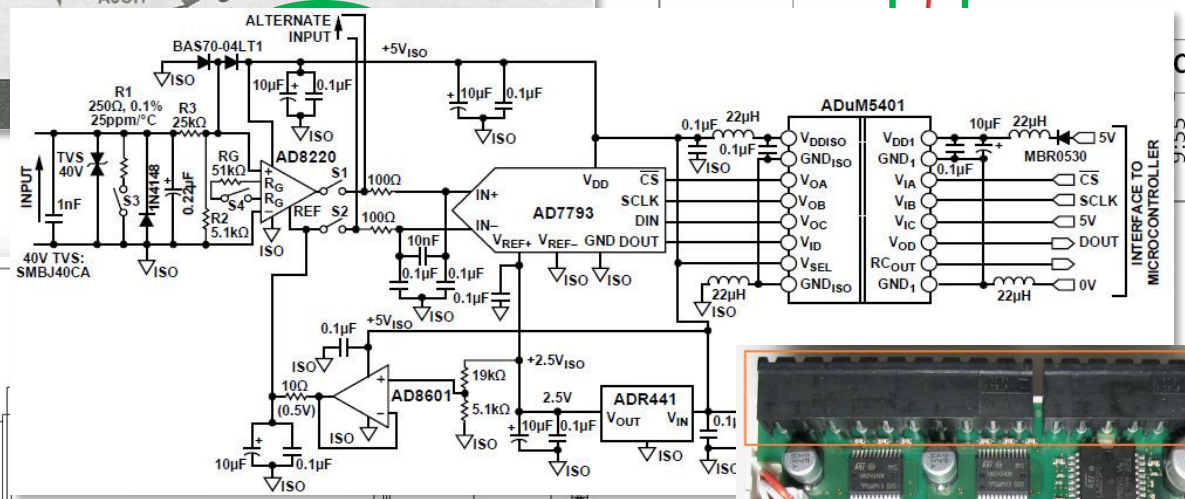
Ali Abbasi
University of Twente
Enschede, Netherlands
a.abbasi@utwente.nl

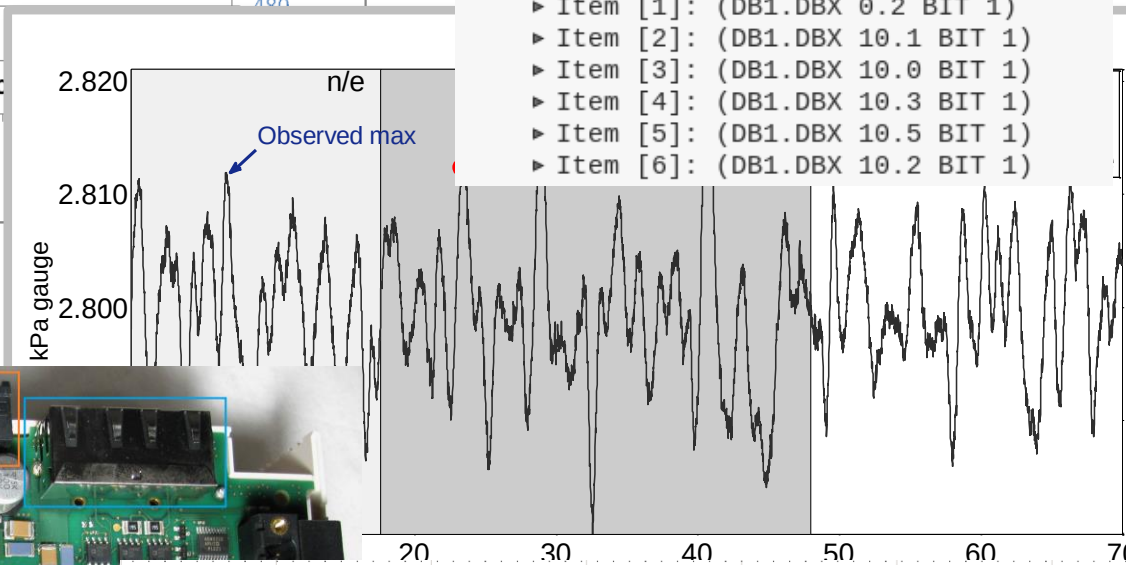
Knoweldge involved into exploit development

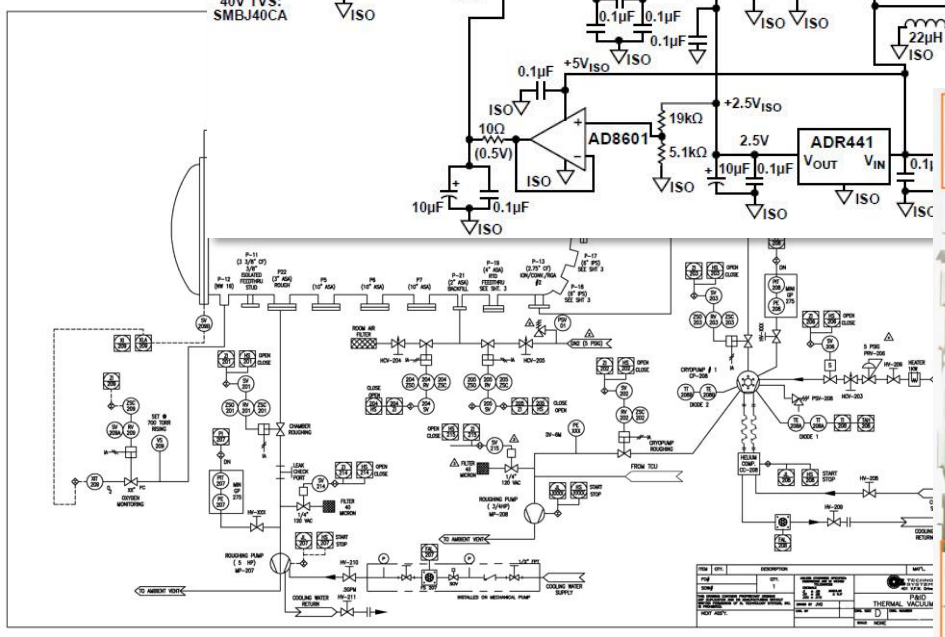


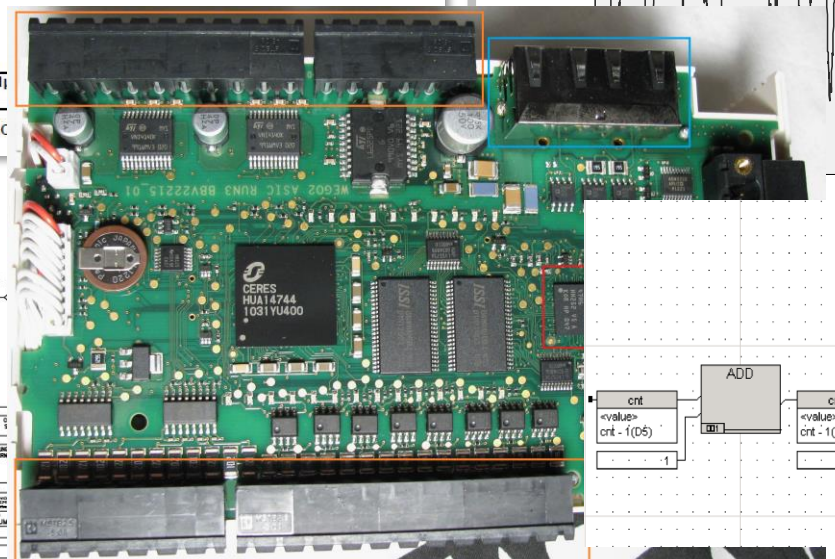


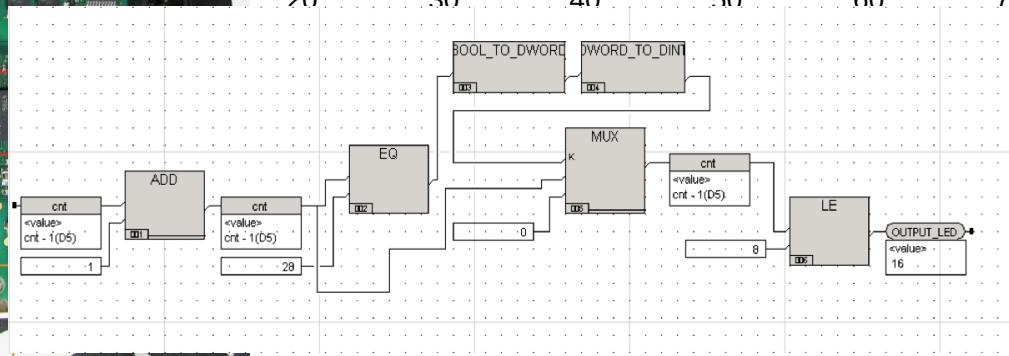




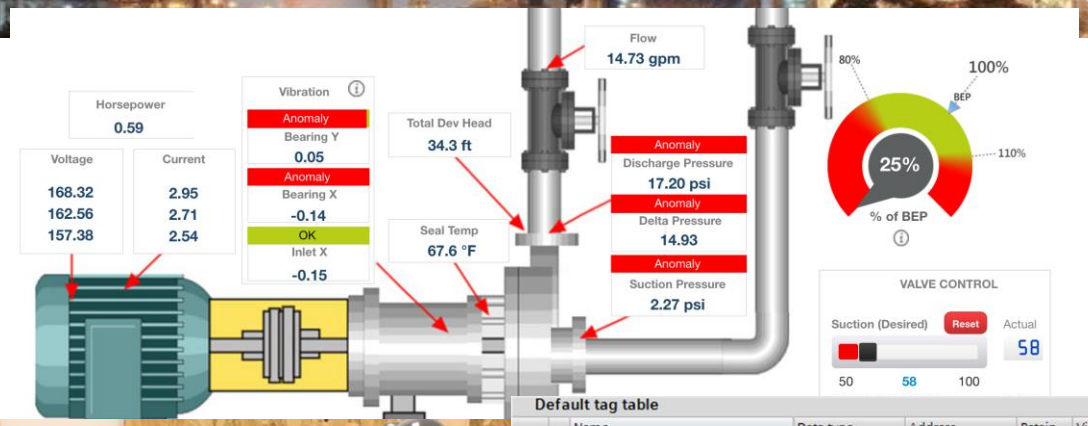






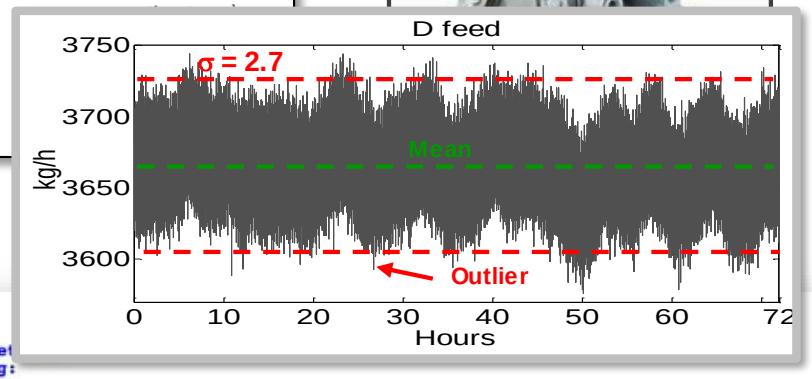


Knoweldge involved into exploit development



Algorithm 1 Runs Analysis

```
procedure EXPLORE                                   ▷ 1: analyse phase
    signal ← signal to analyse

    while not an end of signal do
        while moving up do
            runs++                                  ▷ count positives moves
            value = sum(changes)                    ▷ positive steps change
            if direction change then                ▷ save results
                positivesruns(runs)++
                positivesvalues(runs) = value
        while moving down do
            runs++                                  ▷ count negatives moves
            value = sum(changes)
            if direction change then
                negativesruns(runs)++
                negativesvalues(runs) = value
        if no change then
```



Default tag table

	Name	Data type	Address	Retain	Visible..	Acces...	Comment
1	Emerg-OFF	Bool	%I1.0		☑	☑	Emergency-OFF (nc contact)
2	S3	Bool	%M0.3		☑	☑	pushbutton START S3 (no contact)
3	B1	Bool	%I0.1		☑	☑	sensor safety fence closed (no contact)
4	B2	Bool	%I0.2		☑	☑	sensor cylinder A moved out (no contact)
5	M0	Bool	%Q0.0		☑	☑	move out cylinder A
6	S1	Bool	%M0.1		☑	☑	pushbutton manual mode S1 (no contact)
7	S2	Bool	%M0.2		☑	☑	pushbutton automatic mode S2 (no conta..
8	S4	Bool	%M0.5		☑	☑	pushbutton ON S4 (no contact)
9	S5	Bool	%I0.5		☑	☑	pushbutton OFF S5 (no contact)
10	Motor1	Bool	%Q0.2		☑	☑	motor conveyor belt M01
11	B0	Bool	%I0.3		☑	☑	sensor bottle counting
12	S6	Bool	%I0.6		☑	☑	reset counter / new box

Algorithm 2 Triangles

```
procedure EXPLORE
    signal ← signal to analyse
    window ← learning window
    noiselvl ← noise parameter

    step = window * 10
    topslope = −999.99
    bottomslope = 999.99
    while not an end of signal do
        if first elements then
            current = value
            index = 1
        while index < window do                     ▷ learni
            upperslope = (current − (last + noiselvl))/index
            lowerslope = (current − (last − noiselvl))/index
            if upperslope > topslope then
                topslope = upperslope
            if lowerslope < bottomslope then
```

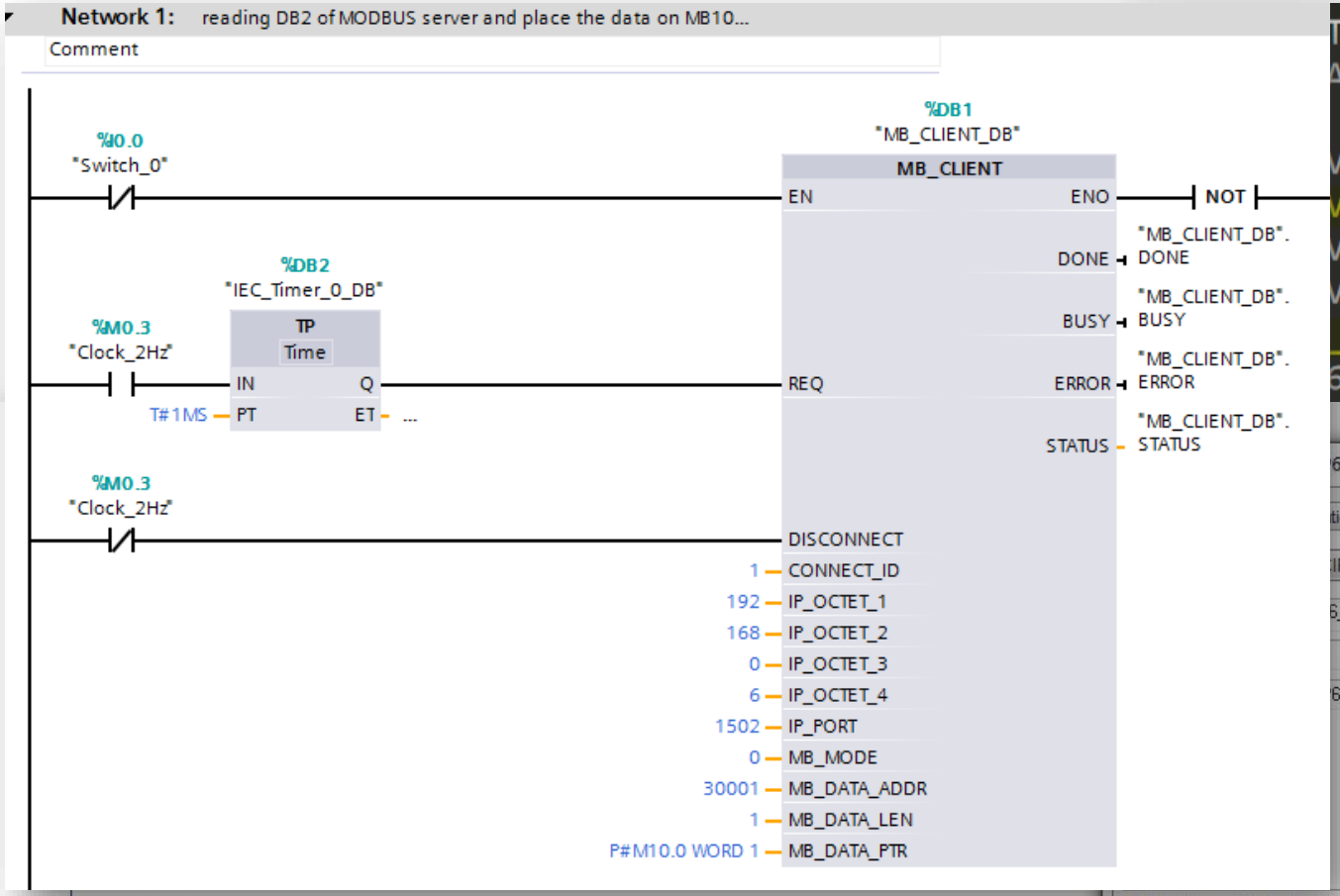
```
.def CalcSomet
CalcSomething:
push.w  R4
mov.w   SP, R4
incd.w  R4
add.w   #0FFFAh, SP
mov.w   R15, 0FFFCh(R4)
clr.w   0FFF8h(R4)
clr.w   0FFFAh(R4)
jmp     loc_22
```

```
loc_22:
cmp.w   0FFFCh(R4), 0FFFAh(R4)
jl      loc_18
```

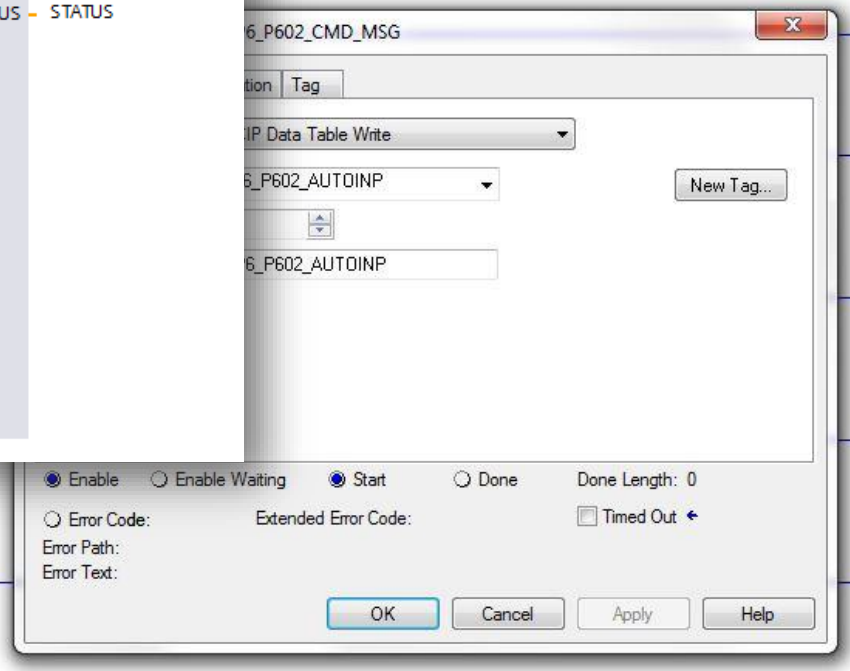
```
loc_18:
add.w   0FFFAh(R4), 0FFF8h(R4)
inc.w   0FFFAh(R4)
```

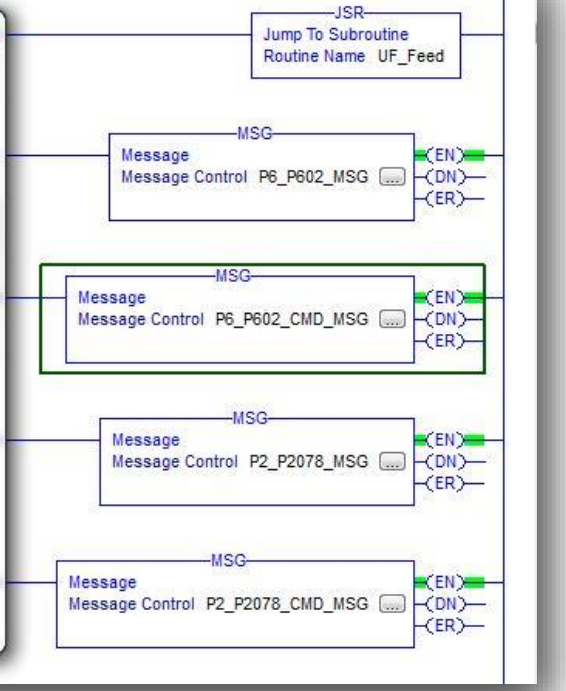
```
mov.w   0FFF8h(R4), R15
add.w   #6, SP
pop     R4
ret
; End of function CalcSomething
```

Control logic is a key component



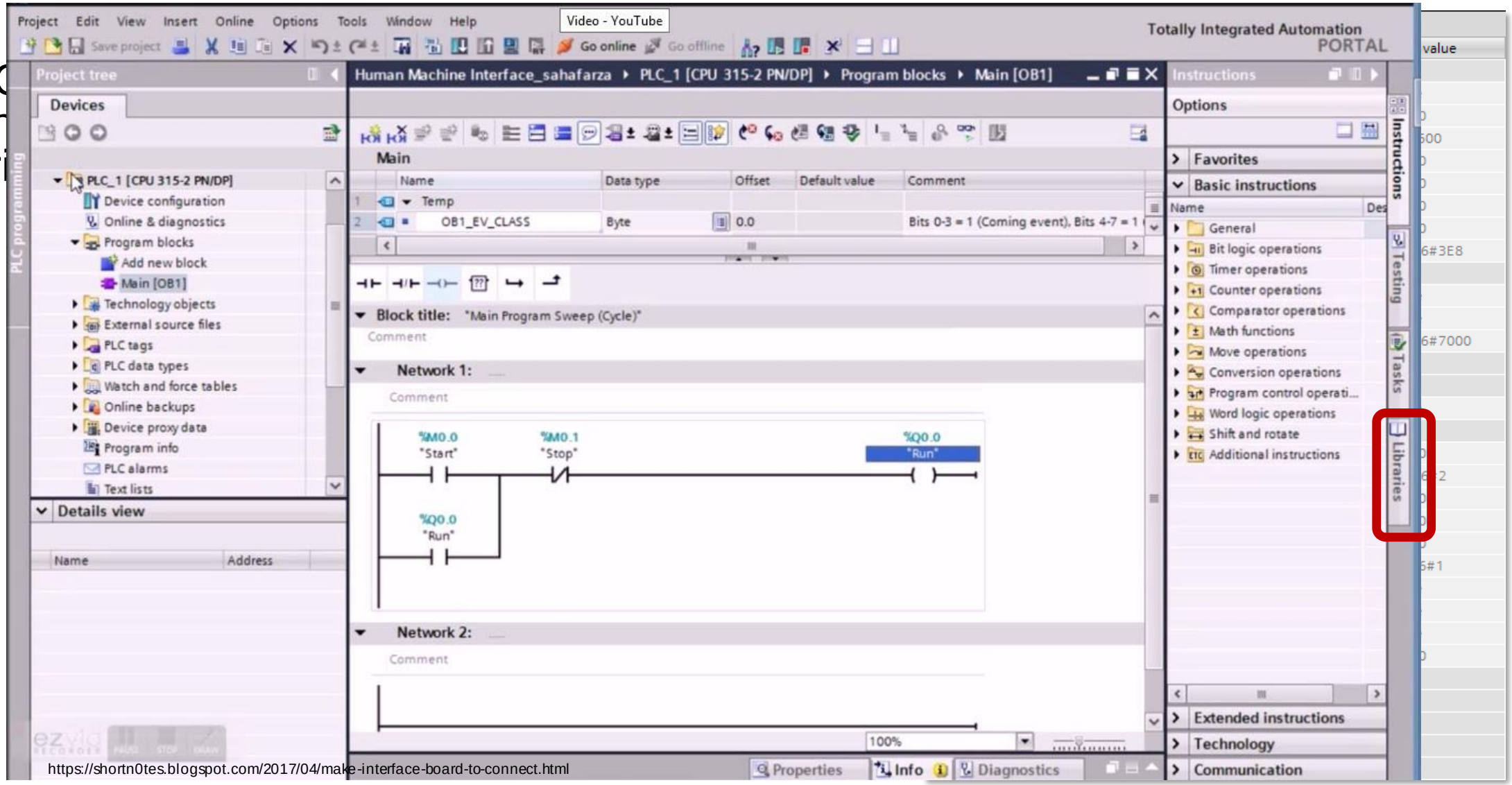

```
TION FOR PRESET TIMER*)
AST_STATE:= HMI_P3_STATE;

V301_AutoInp          :=0;
V302_AutoInp          :=1;
V303_AutoInp          :=0;
V304_AutoInp          :=0;
_UF_FEED_DUTY_AutoInp :=1;
602_AutoInp           :=0;
```





Static memory addressing

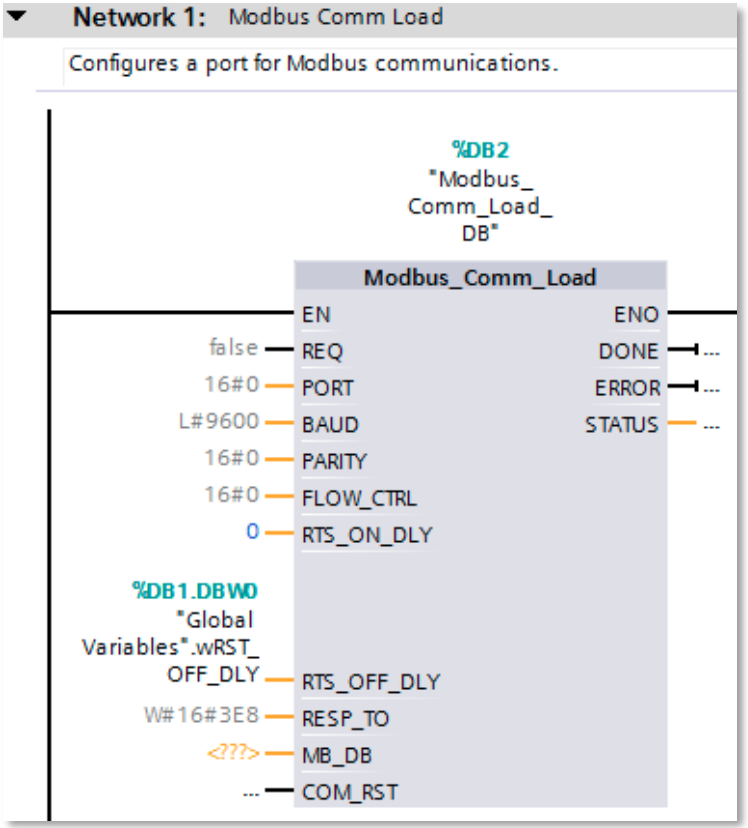
- PLC
  Fun
  Vari



https://shortn0tes.blogspot.com/2017/04/make-interface-board-to-connect.html

Static memory allocation & addressing

- PLC vendors offer libraries of standard Function Blocks (FB) with associated Variable/Data Blocks



Modbus_Comm_Load_DB

		Name	Data type	Offset	Start value
1		Input			
2		REQ	Bool	0.0	false
3		PORT	Word	2.0	16#0
4		BAUD	DInt	4.0	L#9600
5		PARITY	Word	8.0	16#0
6		FLOW_CTRL	Word	10.0	16#0
7		RTS_ON_DLY	Word	12.0	16#0
8		RTS_OFF_DLY	Word	14.0	16#0
9		RESP_TO	Word	16.0	W#16#3E8
10		Output			
11		DONE	Bool	18.0	false
12		ERROR	Bool	18.1	false
13		STATUS	Word	20.0	W#16#7000
14		InOut			
15		MB_DB	Struct	22.0	
16		COM_RST	Bool	28.0	false
17		Static			
18		ICHAR_GAP	Word	30.0	16#0
19		RETRIES	Word	32.0	W#16#2
20		MODE	Byte	34.0	16#0
21		LINE_PRE	Byte	35.0	16#0
22		BRK_DET	Byte	36.0	16#0
23		STOP_BITS	Byte	37.0	B#16#1
24		EN_DIAG_ALARM	Bool	38.0	false
25		EN_SUPPLY_VOLT	Bool	38.1	false
26		b_e_REQ	Bool	38.2	false
27		y_state	Byte	39.0	16#0
28		Send_Config	Send_Config	40.0	
29		Receive_Config	Receive_Config	126.0	
30		Receive_Conditions	Struct	202.0	
31		WRREC	WRREC	270.0	
32		RDREC	RDREC	296.0	

DB/FB enumeration methods 1

- **Metadata**
  - Get Block Info (DB.1, etc.) or List Blocks
  - Detectable as rare command
- **Bulk transfer**
  - Block Upload (DB.1, etc.)
  - Detectable as rare command
- **Bytecode read**
  - Read (DB.1, etc.)
  - Stealth/not easily detectable due to usage of regular command

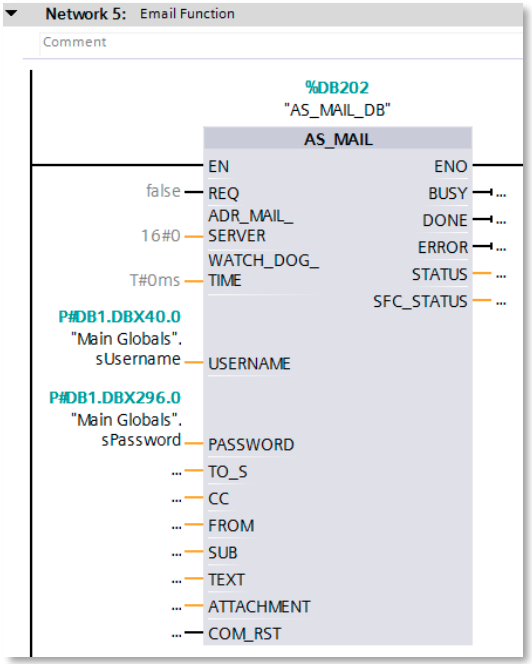
Information leakage vulnerability

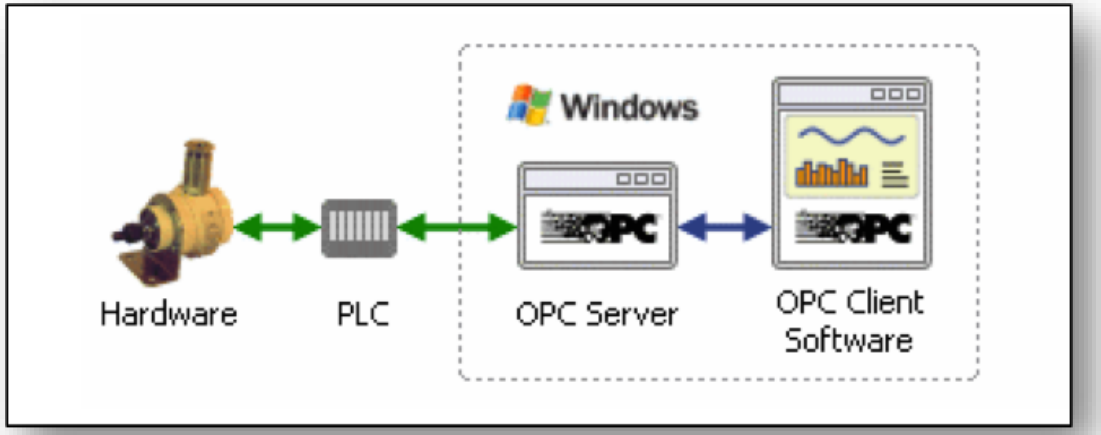
Modbus_Comm_Load_DB

		Name	Data type	Offset	Start value
1		Input			
2		REQ	Bool	0.0	false
3		PORT	Word	2.0	16#0
4		BAUD	DInt	4.0	L#9600
5		PARITY	Word	8.0	16#0
6		FLOW_CTRL	Word	10.0	16#0
7		RTS_ON_DLY	Word	12.0	16#0
8		RTS_OFF_DLY	Word	14.0	16#0
9		RESP_TO	Word	16.0	W#16#3E8
10		Output			
11		DONE	Bool	18.0	false
12		ERROR	Bool	18.1	false
13		STATUS	Word	20.0	W#16#7000
14		InOut			
15		MB_DB	Struct	22.0	
16		COM_RST	Bool	28.0	false
17		Static			
18		ICHAR_GAP	Word	30.0	16#0
19		RETRIES	Word	32.0	W#16#2
20		MODE	Byte	34.0	16#0
21		LINE_PRE	Byte	35.0	16#0
22		BRK_DET	Byte	36.0	16#0
23		STOP_BITS	Byte	37.0	B#16#1
24		EN_DIAG_ALARM	Bool	38.0	false
25		EN_SUPPLY_VOLT	Bool	38.1	false
26		b_e_REQ	Bool	38.2	false
27		y_state	Byte	39.0	16#0
28		Send_Config	Send_Config	40.0	
29		Receive_Config	Receive_Config	126.0	
30		Receive_Conditions	Struct	202.0	
31		WRREC	WRREC	270.0	
32		RDREC	RDREC	296.0	

What we can enumerate

- Hundreds of standard function blocks
  - Communications
  - Remote administration
  - Control algorithms
  - Safety functions
  - Alerting
  - Etc., etc. (a good engineer would know better!)
- Closest analogy previously seen in the wild
  - **Havex** recon campaign, 2013

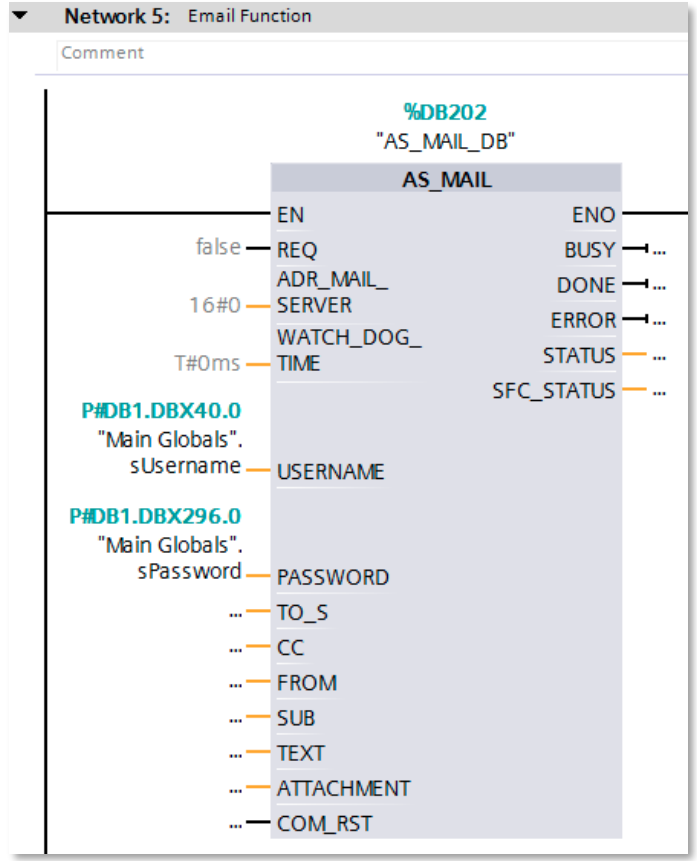




DB content exfiltration

2

- Location of each variable within DB is known
  - Read request
  - *DB.1, offset 4, read 32 bits*
- Large variables stored in global database
  - Locatable via pointers
  - Exfiltrate pointer address
  - Decode address *p#DB.1DBx40.0*
  - Exfiltrate content at the decoded address (read 256 bytes for strings)




```
      DB(1)                                    Offset(40)
{0000000000000001}1000010000000{0000000000101000}000
```

42	sUsername	String	40.0	'test@test.com'
43	sPassword	String	296.0	'mypassword'

DB content manipulation

3

- Use write commands at target addresses
  - Variable values assigned directly
  - Default values
- Some variables are stored in global DB (via pointers)
  - “Pushed” to local DB every scan cycle (e.g., every 10 ms or 1sec)
  - Race condition situation for the attacker
  - Use smart tricks

IEC_Counter_0_DB

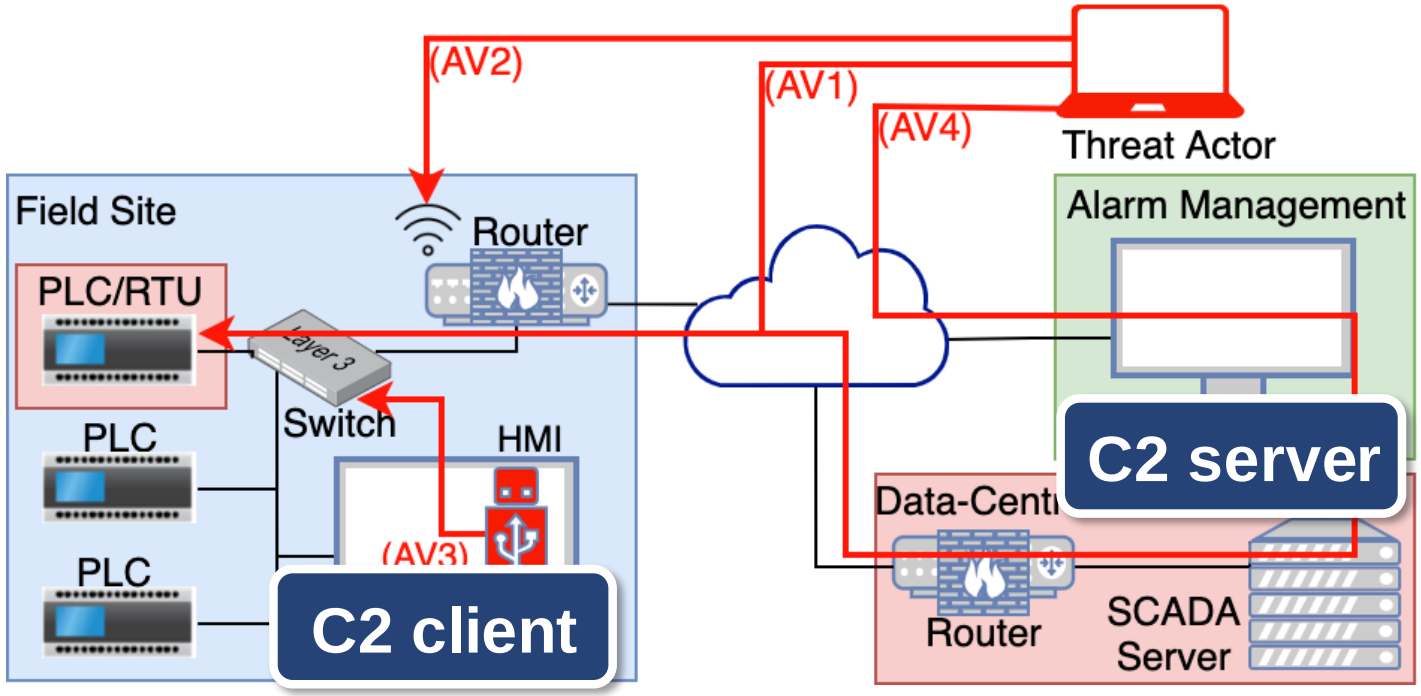
	Name	Data type	Offset	Start value
1	Input			
2	CU	Bool	0.0	FALSE
3	R	Bool	0.1	FALSE
4	PV	Int	2.0	0
5	Output			
6	Q	Bool	4.0	FALSE
7	CV	Int	6.0	0
8	InOut			
9	Static			
10	CUO	Bool	8.0	FALSE

C2 channel to segregated environments

- Violates network segmentation defense/best practice (IEC 62443)
- Up to 10 bytes of unused memory with multiple incomplete bytes per DB
- Allows execution of commands at console level
  - E.g., *ping 192.168.0.1*

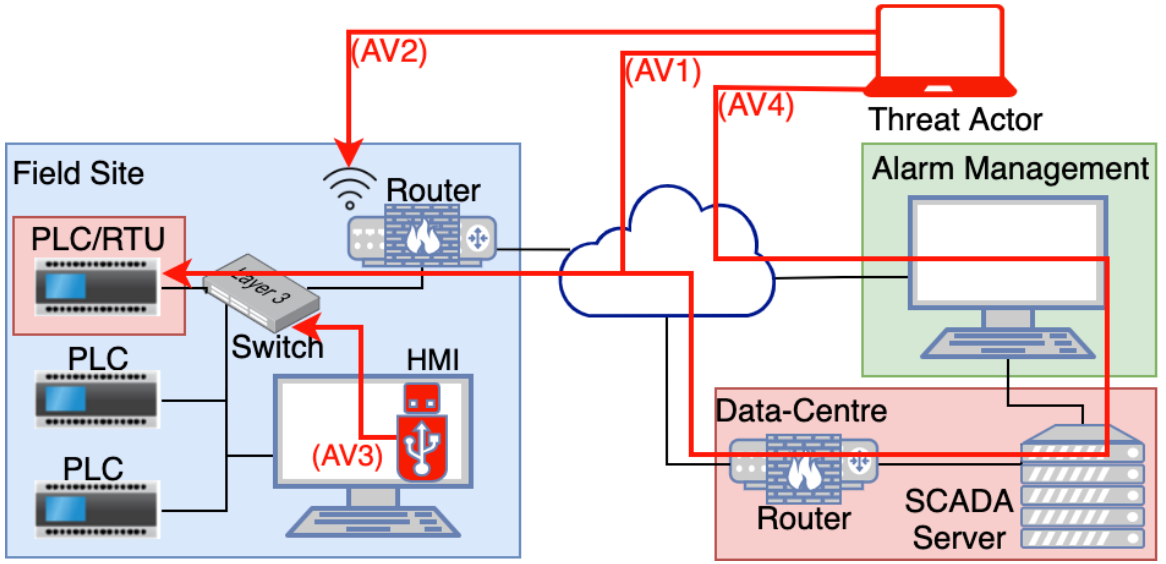
Function	C2-Server	C2-Client
Hello		00000001
Hello Ack	00000011	00000000
Write	01000000	11100000
Reading	11110000	01100000
Read	00000000	00000000
Final Write	11111111	11111110
On Hold	00011000	00011000

Table 1: Synchronization Byte



Detectability of attack techniques

- C2 communication is preventable/detectable by perimeter firewalls
- C2 based on *Read/Write* commands from trusted devices are not detected

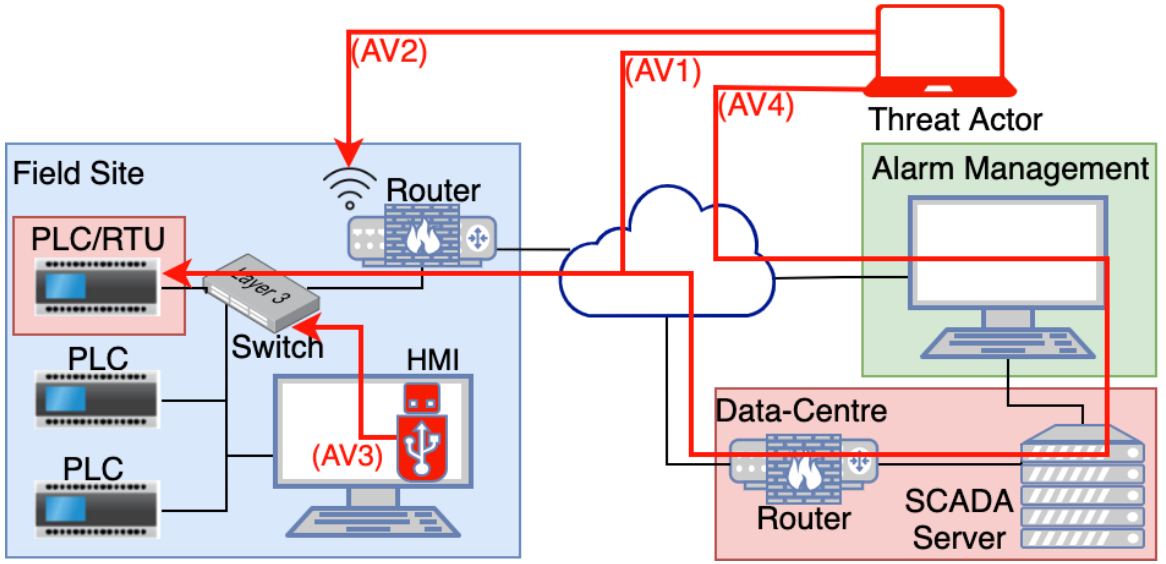


Prevention Results.

Prevention										
Vendor/Device	Trusted/Untrusted	T1	T2	T3	T4	T5	T5s	T6	T7	T8
Siemens S623	Untrusted	N	N	N/A	N	N	N/A	N	N	Y
	Trusted	N	N	N/A	N	N	N/A	N	N	Y
Tofino Xenon	Untrusted	Y	Y	N/A	Y	Y	N/A	Y	Y	Y
	Trusted	N	N	N/A	N	N	N/A	N	N	Y
Westermo Redfox	Untrusted	Y	Y	N/A	Y	Y	N/A	Y	Y	Y
	Trusted	N	N	N/A	N	N	N/A	N	N	N
Checkpoint 1570R	Untrusted	Y	Y	N/A	Y	Y	N/A	Y	Y	Y
	Trusted	N	N	N/A	N	N	N/A	Y	Y	Y

Detectability of attack techniques

- Network monitoring solution with traffic baselining detect baseline deviation (Claroty)
  - Generates Event
  - *”Baseline deviation change, not risky change”*
  - No security Alert



Detection										
Vendor/Device	Trusted/Untrusted	T1 (A/E)	T2 (A/E)	T3 (A/E)	T4 (A/E)	T5 (A/E)	T5s (A/E)	T6 (A/E)	T7 (A/E)	T8 (A/E)
Claroty CTD	Untrusted	Y (A)	Y (A)	N/A	Y (A)	Y (A)	N/A	Y (A)	Y (A)	Y (A)
	Trusted	Y (E)	Y (E)	Y (E)	Y (E)	Y (E)	Y (A)	Y (A)	Y (A)	Y (A)

Broader applicability of attack technique

- Allen Bradley SLC 500
  - Uses similar memory allocation approach
- ABB variable frequency drive
  - Provides library functions for e.g. Siemens PLC for drive control
  - Vulnerable to the same exploitation approach

Conclusions

- By exploiting memory allocation and addressing we developed approach to enumerate & manipulate function blocks/control logic on PLC
  - Applicable to arbitrary industrial environments
  - Using stealth techniques/undetectable (**only read & write commands!!)**
  - Fully automated exploit of high targeting precision
  - Establishment of covert channel to isolated network segments

- Exploitation of supply chain to attack supply chain
  - Profiling custom functions/FBs
  - Delivery of exploitation code

SCADA PROJECTS FROM THE POINT OF VIEW OF HACKERS

https://2018.zeronights.ru/en/reports/scada-projects-from-the-point-of-view-of-hackers/

Conclusions

Currently asset owner is blamed in all occurrences of asset exploitation

The blame should be shared with asset vendor

Q & A

Marina Krotofil
@marmusha
marmusha@gmail.com