PAPER SUMMARY PRESENTATION
IoT and Medical Device Sec

GROUP 6:

JASON CHEN, KAIYU DONG, ALLEN LIU

Medical Device Security in the IoT Age

What's a Medical Device?

- Pacemaker
- Surgical Robot
- Vital Sign Monitor

What if a med device is hacked?

Patient data can be stolen

Hospital network compromise

Device functionality interrupted

Attack Vector 1

Bluetooth Low Energy

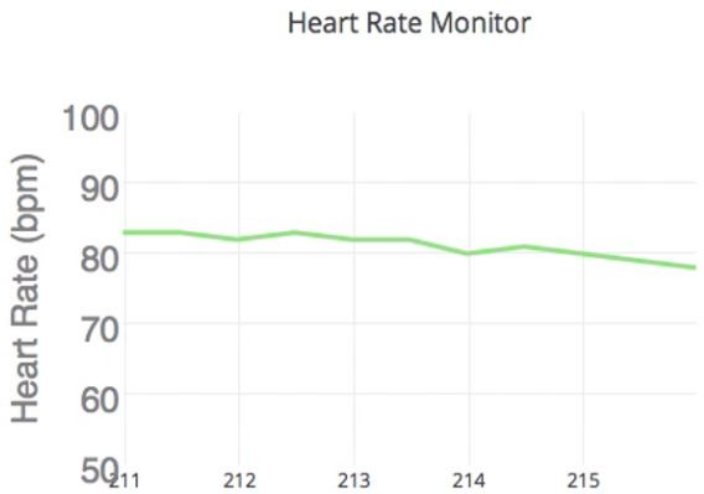
Terminal Shell Edit View Window Help

100% Wed 12:43 PM

file:///tmp/output.html

Patient Monitor

Heart Rate: 78



2_bluetooth_security_demo — pi@MC-RP3-A: ~ — ssh pi@192.168.7.192 — 130×12

...th_security_demo — -bash | pi@MC-RP3-A: ~ — ssh p... | pi@MC-RP3-B: ~ — ssh p... | pi@MC-RP3-C: ~ — ssh p...

```
Sending payload: 14FF004D0142B12BD4C854664C35715F1A1F03DBB6
Sending payload: 14FF004D01E759B2C0C50CECAD16B3DE8B3B0DB2CA
Sending payload: 14FF004D01308F02C632F108F302FDD2894038D9DB
Sending payload: 14FF004D01E71D7AAEEF23464AEE5F6FA67F9A252D
Sending payload: 14FF004D01A59CA5A124C87FCD0D7E14E84D8ED462
Sending payload: 14FF004D01A3F498E19C4D4C9B19C09855C7416CED
Sending payload: 14FF004D015589619 5AEB251F7FD558AD9C0A7D967
Sending payload: 14FF004D013997D15DAA64735AEBFB85594FE2EB24
Sending payload: 14FF004D01CF88F4DC8035F5D83E40EF4B32FECC5E
Sending payload: 14FF004D0198ACEF1047D1A4537A5E01285385E346
Sending payload: 14FF004D011CAE132F298B7AA2A6A83621351E0B0E
```

iPhone 7 – iOS 10.3 (14E269)

Heart Rate

78

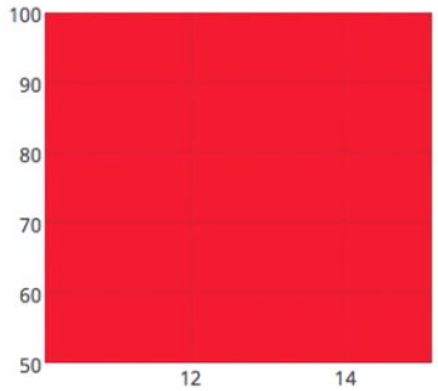
Terminal Shell Edit View Window Help 100% Wed 12:44 PM

file:///tmp/output.html

Patient Monitor

Heart Rate: err

Monitor Disconnected!



Hacker

[TX]14FF004D013B3FFFEA7A1A6DD3BBCFEBA90EFDB1FB
[TX]14FF004D016BB43565582CCF008AD81A04216F6A98
[TX]14FF004D01BF1465FEF6C3AF5B3D3355ABDC3B4F04
[TX]14FF004D01C58ED5DCBBC915B2C073500B9E9860C5
[TX]14FF004D0170F72C400F8C4FB665890BEFC58026B5
[TX]14FF004D01DC6B3F21A38D0E9412E9BB71E224801B
[TX]14FF004D01E6BF67ECAEFE94C90115A92475B05737
[TX]14FF004D016CE10CE778F53DA434C64BC1DFB7EBFE
[TX]14FF004D01AD92D82F5938C83968CED48C15F5AD91
[TX]14FF004D011B1E6A68603C7028DFCDD7CCF1E68509
[TX]14FF004D01AD92D82F5938C83968CED48C15F5AD91
[TX]14FF004D016CE10CE778F53DA434C64BC1DFB7EBFE
[TX]14FF004D01E6BF67ECAEFE94C90115A92475B05737
[TX]14FF004D01DC6B3F21A38D0E9412E9BB71E224801B
[TX]14FF004D0170F72C400F8C4FB665890BEFC58026B5
[TX]14FF004D01C58ED5DCBBC915B2C073500B9E9860C5
[TX]14FF004D01BF1465FEF6C3AF5B3D3355ABDC3B4F04
[TX]14FF004D016BB43565582CCF008AD81A04216F6A98
[TX]14FF004D013B3FFFEA7A1A6DD3BBCFEBA90EFDB1FB
[TX]14FF004D016BB43565582CCF008AD81A04216F6A98

2_bluetooth_security_demo — pi@MC-RP3-C: ~ — ssh pi@192.168.7.194 — 130×12

...th_security_demo — -bash | pi@MC-RP3-A: ~ — ssh p... | pi@MC-RP3-B: ~ — ssh p... | pi@MC-RP3-C: ~ — ssh p...

```
Sending payload: 14FF004D011B1E6A68603C7028DFCDD7CCF1E68509
Sending payload: 14FF004D01AD92D82F5938C83968CED48C15F5AD91
Sending payload: 14FF004D016CE10CE778F53DA434C64BC1DFB7EBFE
Sending payload: 14FF004D01E6BF67ECAEFE94C90115A92475B05737
Sending payload: 14FF004D01DC6B3F21A38D0E9412E9BB71E224801B
Sending payload: 14FF004D0170F72C400F8C4FB665890BEFC58026B5
Sending payload: 14FF004D01C58ED5DCBBC915B2C073500B9E9860C5
Sending payload: 14FF004D01BF1465FEF6C3AF5B3D3355ABDC3B4F04
Sending payload: 14FF004D016BB43565582CCF008AD81A04216F6A98
Sending payload: 14FF004D013B3FFFEA7A1A6DD3BBCFEBA90EFDB1FB
Sending payload: 14FF004D016BB43565582CCF008AD81A04216F6A98
^Cpi@MC-RP3-C:~ $
```

iPhone 7 – iOS 10.3 (14E269)

Carrier 12:44 PM

localhost

Heart Rate

79

What would happen?

False vitals -> wrong diagnosis

Interrupted monitoring -> missing critical event

False vitals -> tainted data analysis algorithm

Apply:

Don’t rely solely on BLE security

Don’t rely solely on “proprietary protocol”

If you’re a manufacturer, consider data integrity checks

If you’re a hospital, ask if there are data integrity checks

Attack Vector 2

Man in the Middle

Control Console 673f2aeb-df6a-42dd-babb-3aaaa0cac9c5

20% Sun 11:30 AM Mike Kijewski

Treatment Console

Radiotherapy
Treatment Plan

Mode: Treatment
Energy: MV
Doserate: 600 MU/min
Time: 80.00 sec
Dose: 800 cGy
Ident:
Login

MCS | Execute | Beam On

Linac

Beam Off

MedCrypt Debug

Include Signature | NO_SSL

```
 fileData {
 }
}

Treatment Time in Millis: 4000.0
Sending Event : Client [NORMAL] 'Command Successful (127.0
 "doserate": 600.0,
 "deliveredDose": 800.0
}' (demo.medcrypt.co:8080)
"Beam On" Command Complete.
Sending Beam Settings and "Beam On" Command to Linac...
```

Malicious Actor - Man-in-the-middle (MITM) Attack

```
Mar 04, 2018 11:30:46 AM io.netty.handler.logging.LoggingHandler channelReadComplete
INFO: [id: 0x484de7af, L:/0:0:0:0:0:0:0:0:8462] READ COMPLETE
---> Altering the command from [{
 "doserate": 600.0,
 "deliveredDose": 800.0
}] to [{
 "doserate": 2400.0,
 "deliveredDose": 20000.0
}]
---> Altering the response from [Signature could not be verified.  Cancelling Execution of Comman
---> Altering the status from [ERROR] to [OK]
```

8462 | 127.0.0.1 | 8463 | R | Active

```
.LoggingHandler channelRegistered
.LoggingHandler bind
.LoggingHandler channelActive
TIVE
.LoggingHandler channelRead
```

What would happen?

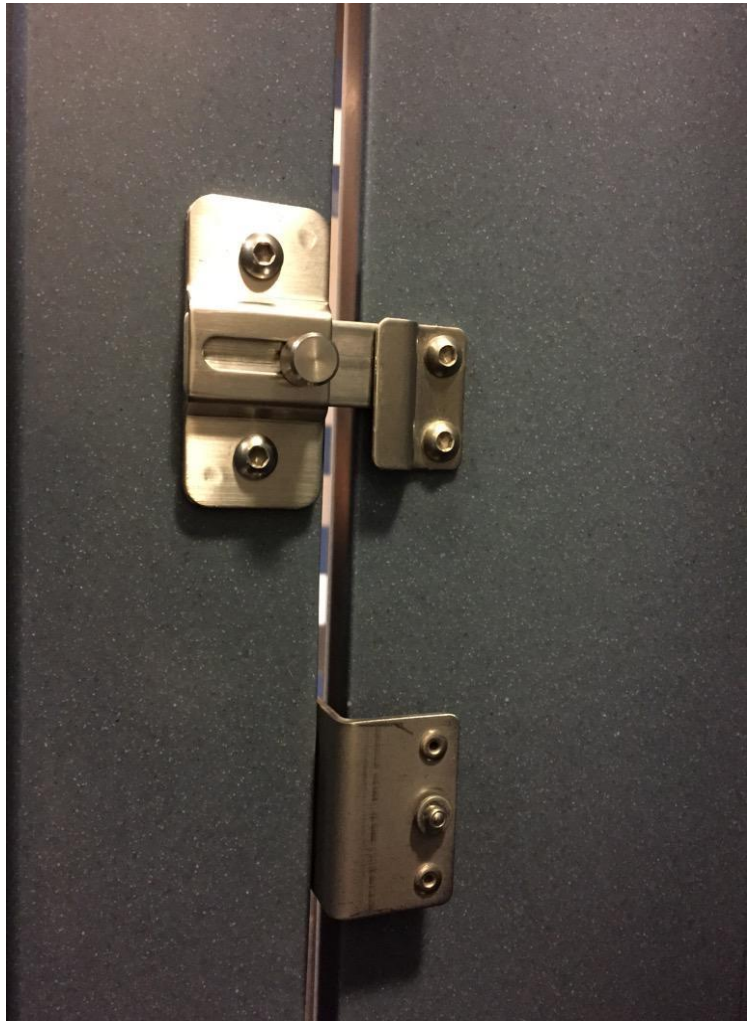
Too much radiation

Not enough radiation

Misplaced radiation dose

Availability disruption -> patient relocation?

Who would do this?



Apply:

Don’t rely solely on perimeter security

If you’re a manufacturer, consider security by design

If you’re a hospital, ask if the device is secure by design

Other Attack Vectors

OS Vulnerability (e.g. XP)

User Authentication (or lack of)

Remote software updates (no verification)

Apply: How to mitigate these vulns during design

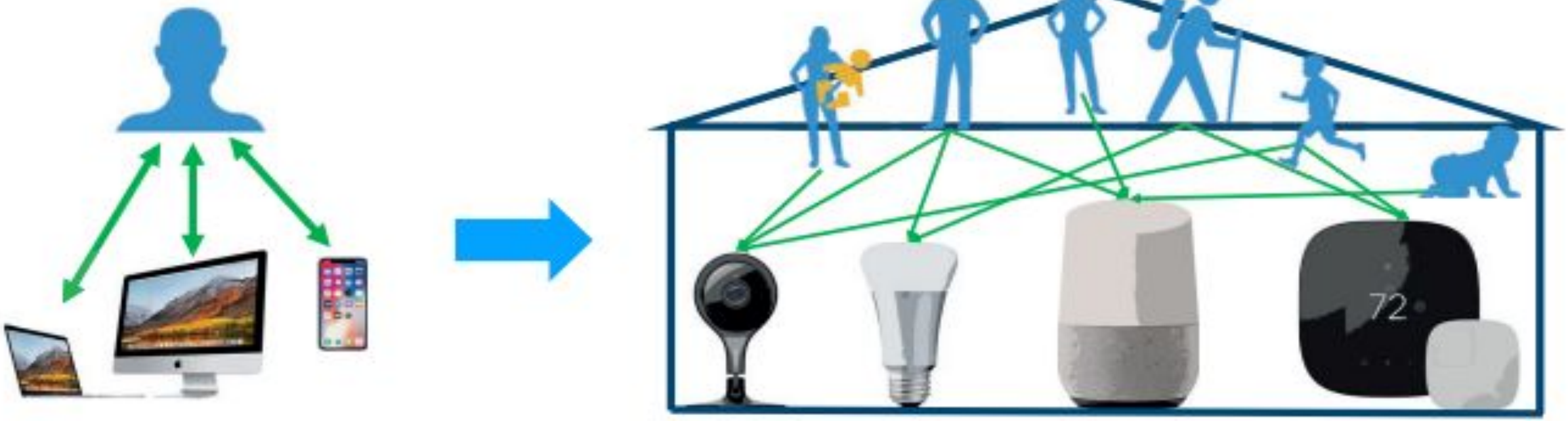
Unique keys on each device / endpoint

Encrypt stuff locally

Sign stuff locally

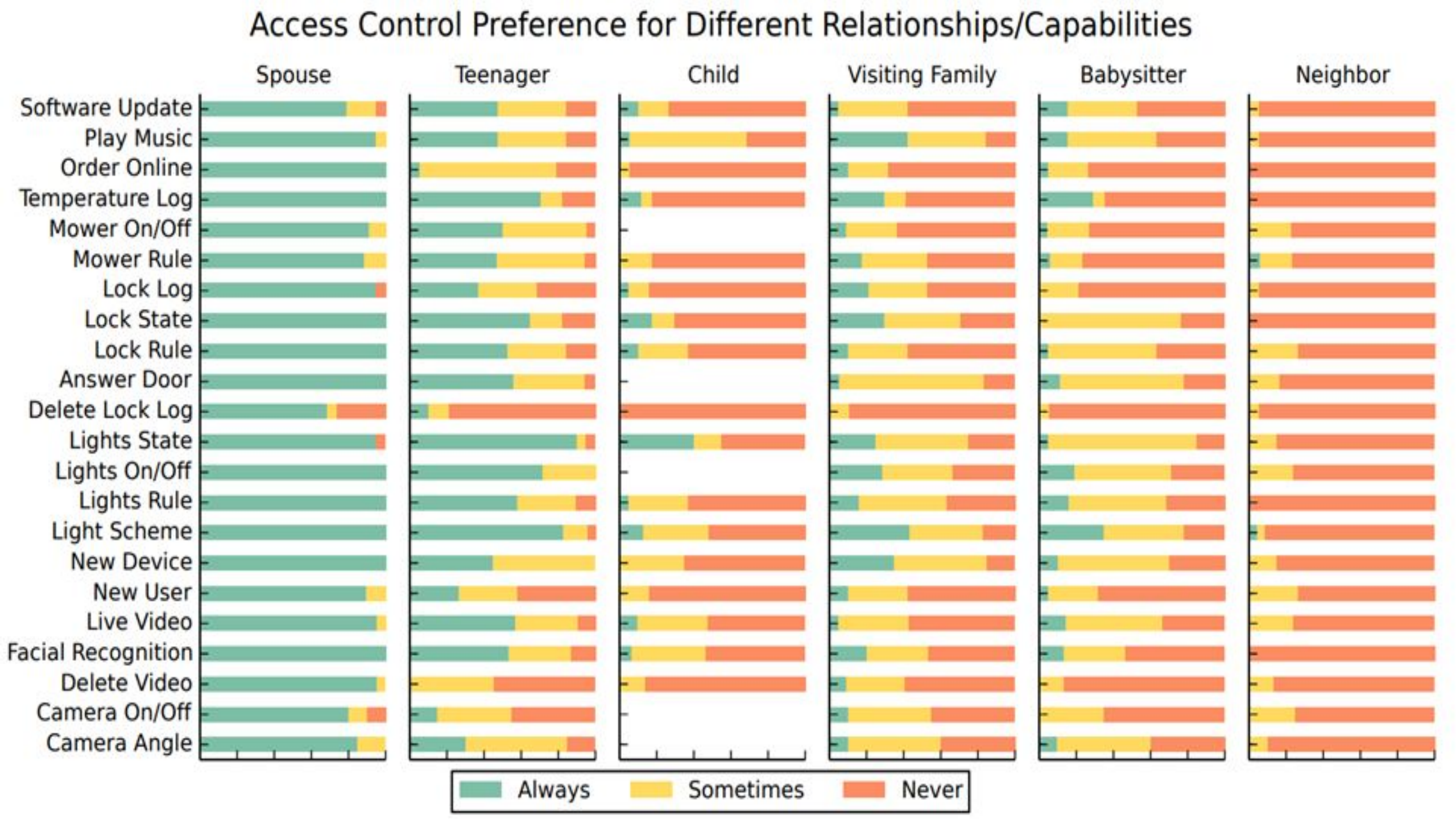
Verify signatures on data / commands before acting

Rethinking Access Control and Authentication for the Home Internet of Things

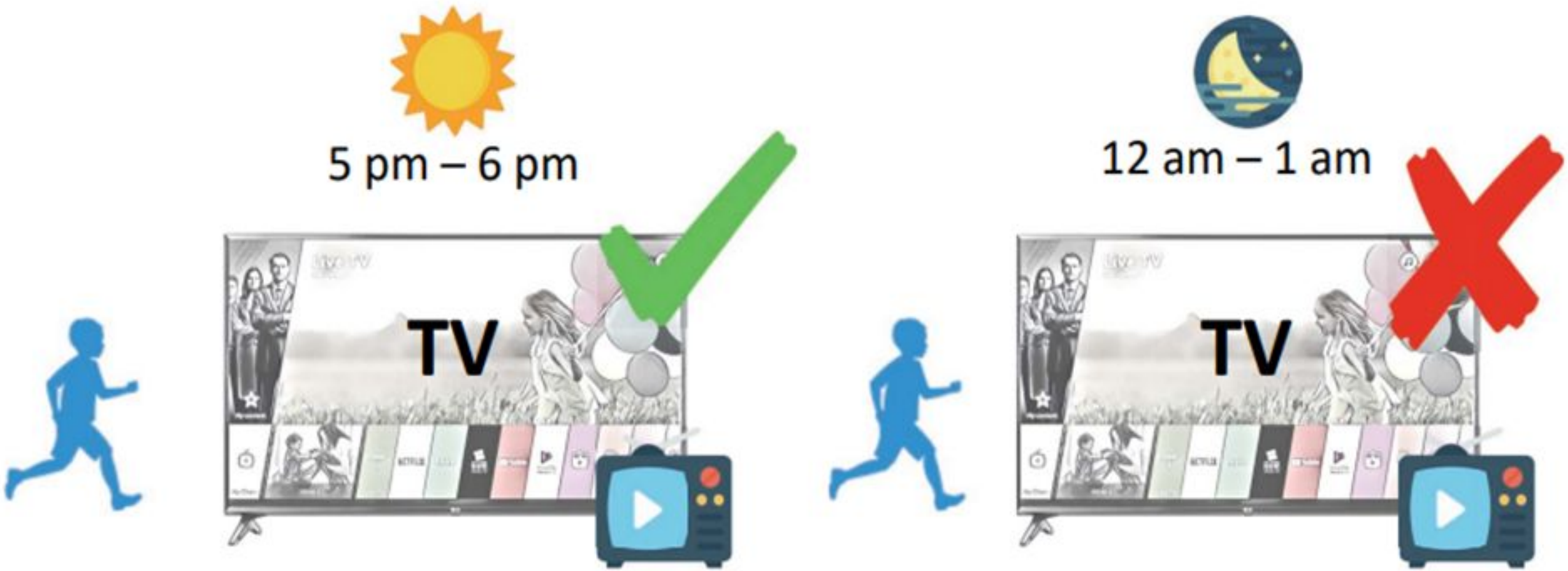


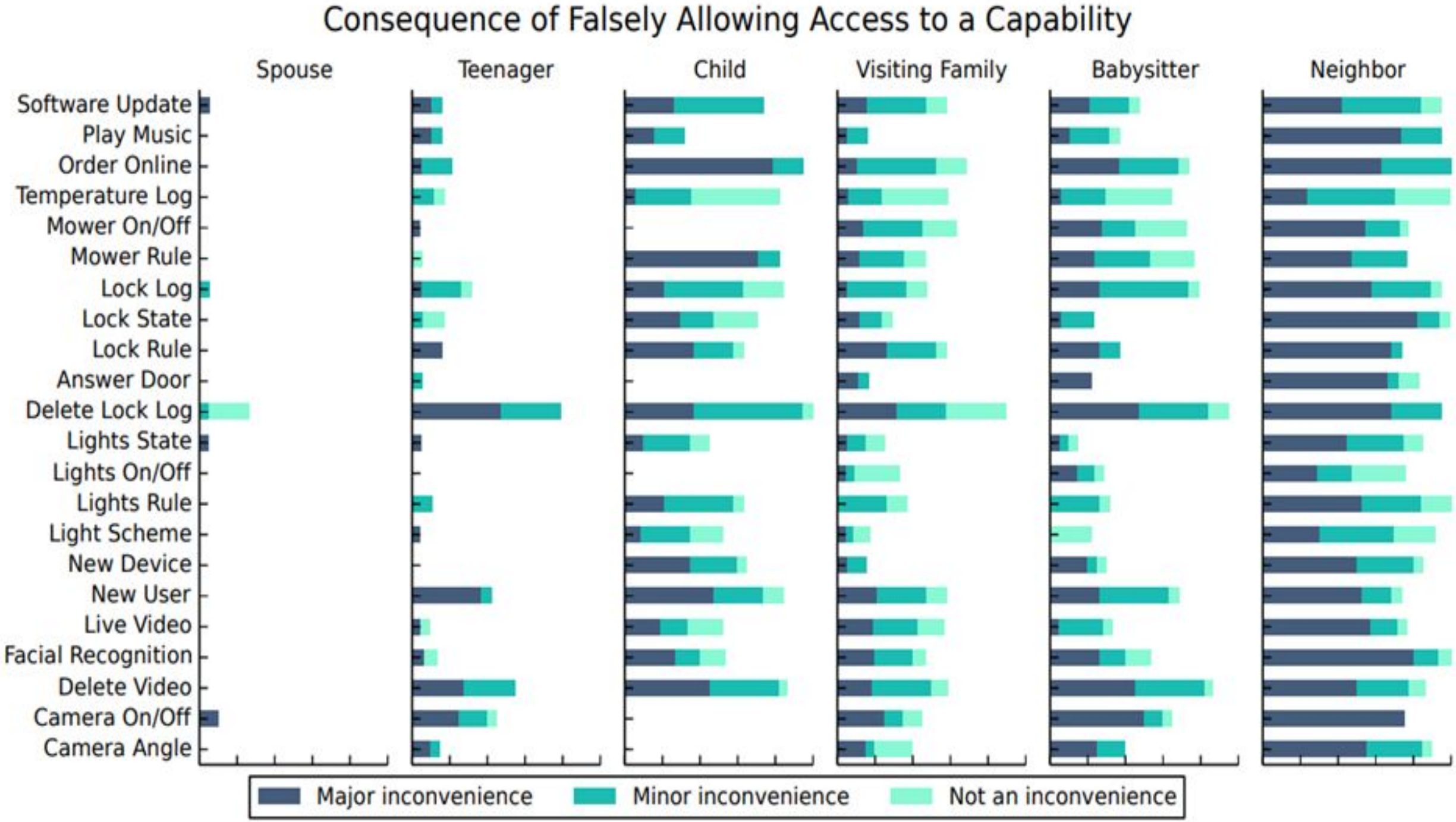
Single User

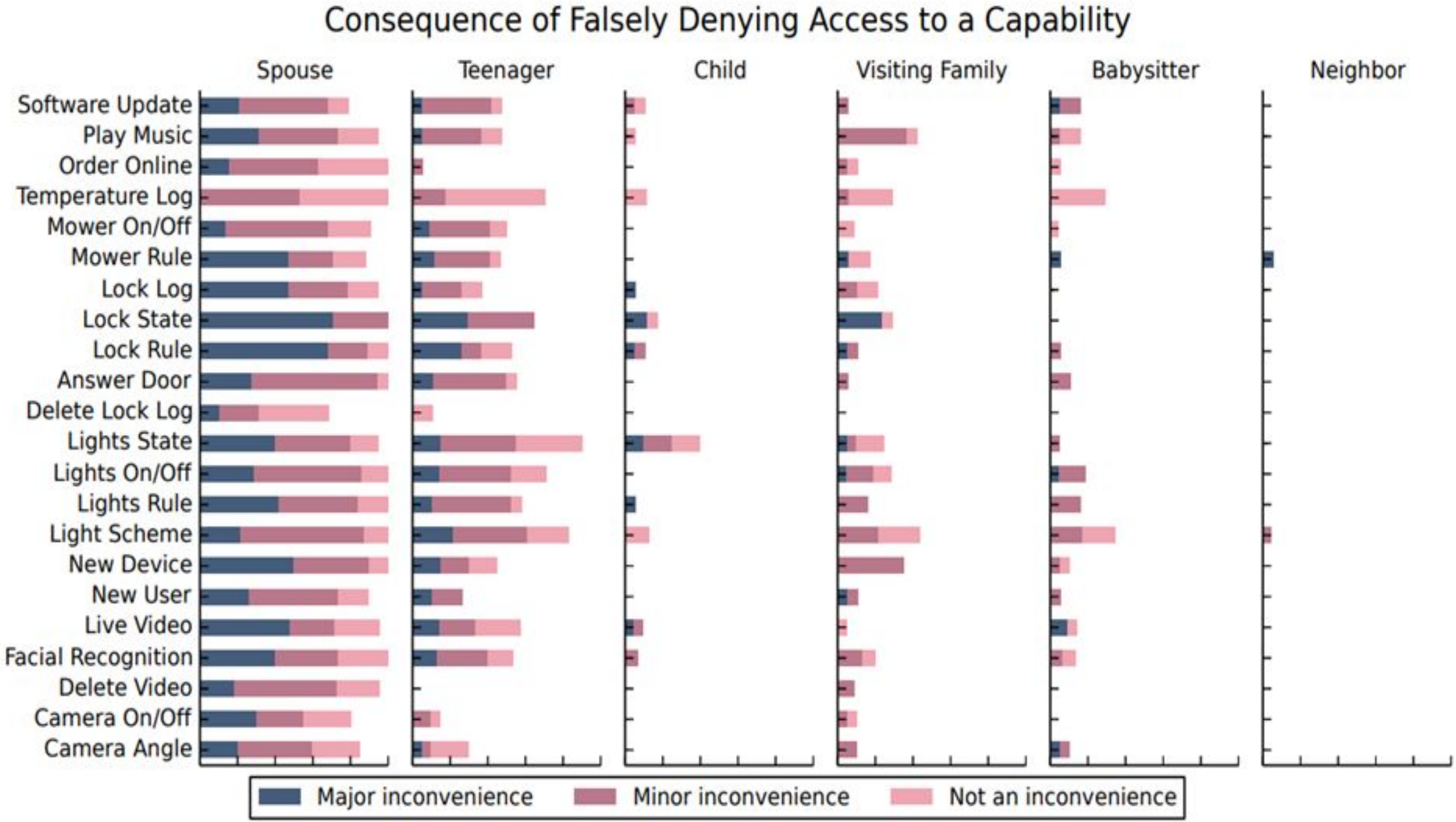
Multi User

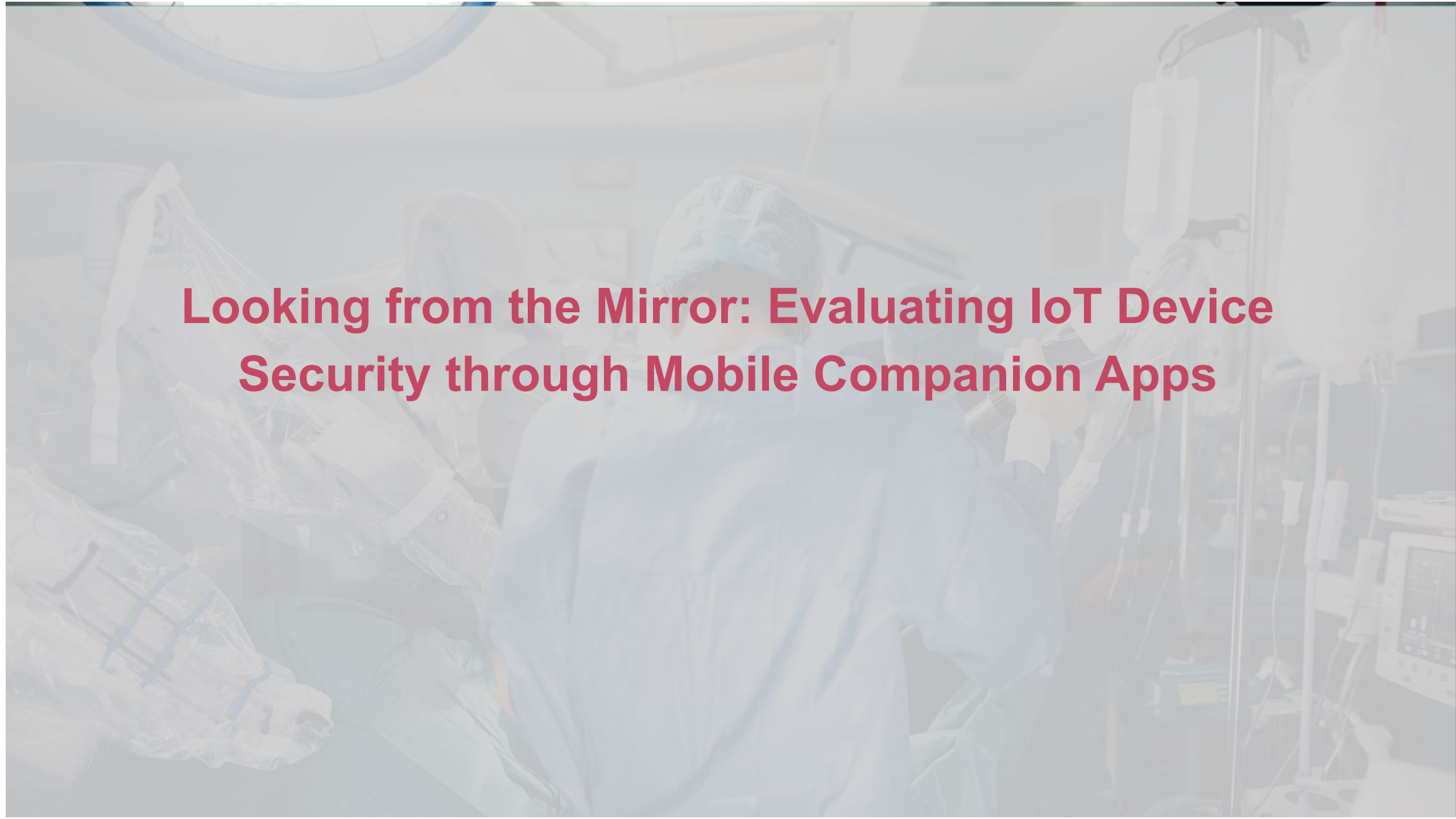

Access Control Preference for Different Relationships/Capabilities
Spouse
Teenager
Child
Visiting Family
Babysitter
Neighbor
Software Update
Play Music
Order Online
Temperature Log
Mower On/Off
Mower Rule
Lock Log
Lock State
Lock Rule
Answer Door
Delete Lock Log
Lights State
Lights On/Off
Lights Rule
Light Scheme
New Device
New User
Live Video
Facial Recognition
Delete Video
Camera On/Off
Camera Angle
Always
Sometimes
Never

Different capabilities may be different within a single device

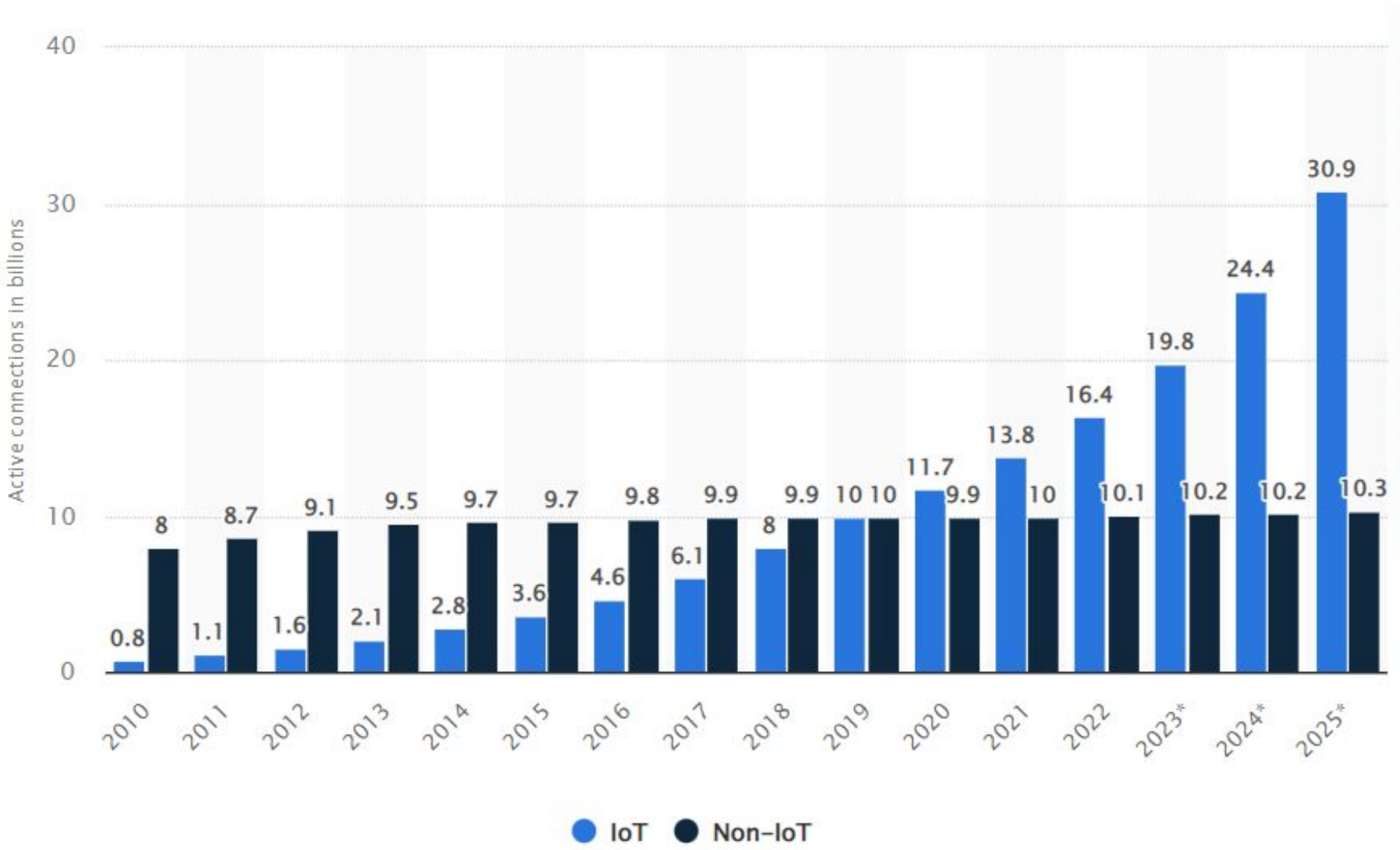



Consequence of Falsely Allowing Access to a Capability
Spouse
Teenager
Child
Visiting Family
Babysitter
Neighbor
Software Update
Play Music
Order Online
Temperature Log
Mower On/Off
Mower Rule
Lock Log
Lock State
Lock Rule
Answer Door
Delete Lock Log
Lights State
Lights On/Off
Lights Rule
Light Scheme
New Device
New User
Live Video
Facial Recognition
Delete Video
Camera On/Off
Camera Angle
Major inconvenience
Minor inconvenience
Not an inconvenience


Consequence of Falsely Denying Access to a Capability
Spouse
Teenager
Child
Visiting Family
Babysitter
Neighbor
Software Update
Play Music
Order Online
Temperature Log
Mower On/Off
Mower Rule
Lock Log
Lock State
Lock Rule
Answer Door
Delete Lock Log
Lights State
Lights On/Off
Lights Rule
Light Scheme
New Device
New User
Live Video
Facial Recognition
Delete Video
Camera On/Off
Camera Angle
Major inconvenience
Minor inconvenience
Not an inconvenience



Looking from the Mirror: Evaluating IoT Device Security through Mobile Companion Apps



IoT and non-IoT active device connections worldwide from 2010 to 2025

HELPNETSECURITY

CRITICALITY AND IMPACT OF XIOT VULNERABILITIES IN 1H 2022

46.05% High

32.26% Medium

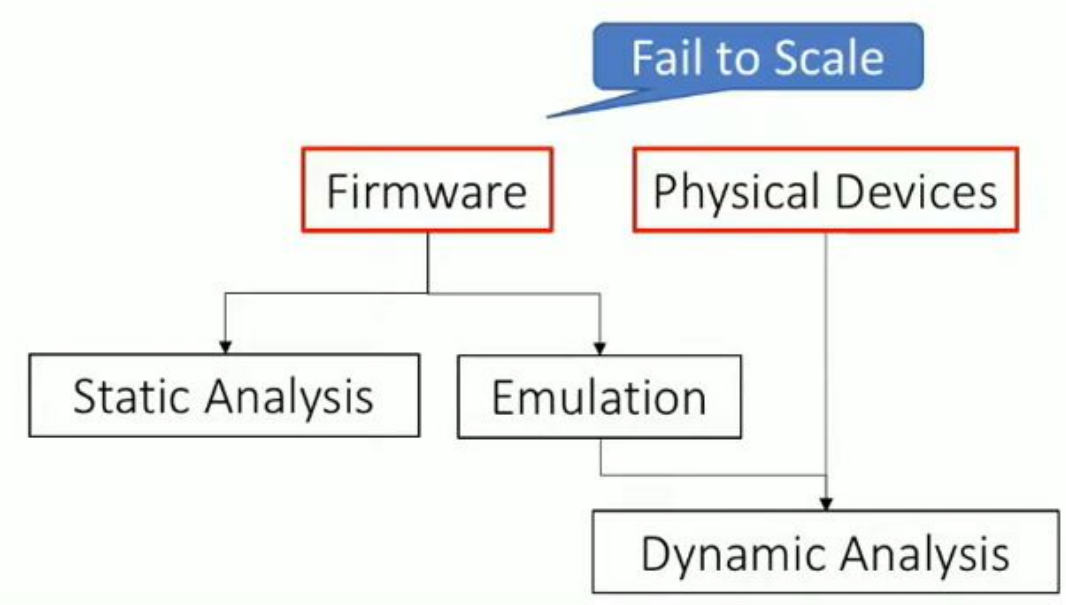
18.61% Critical

3.08% Low

SOURCE: CLAROTY

Identifying Vulnerable IoT Devices

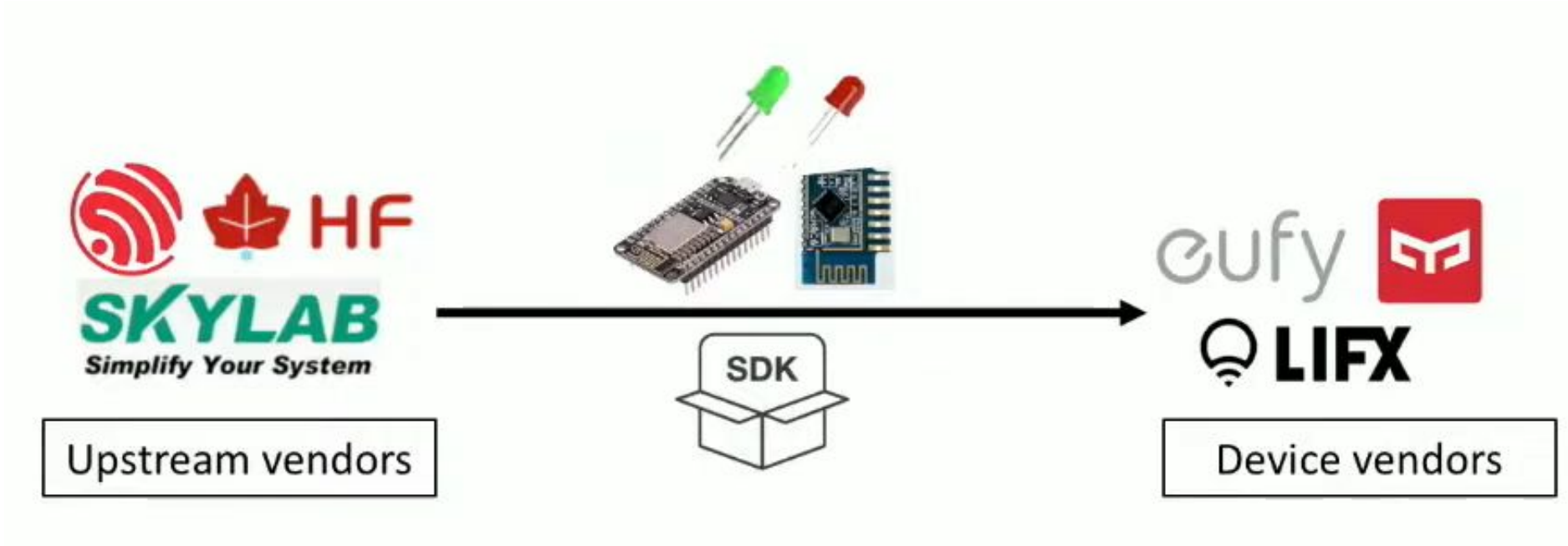
- Static Analysis
  - Vulnerability search
  - Symbolic execution
- Dynamic Analysis
  - Physical/emulated devices
  - Fuzzing, etc.



Question: How to identify the vulnerable IoT devices in a scalable way?

Insights

- IoT devices share HW/SW components.
- **Vulnerability propagates** between devices!



Insights

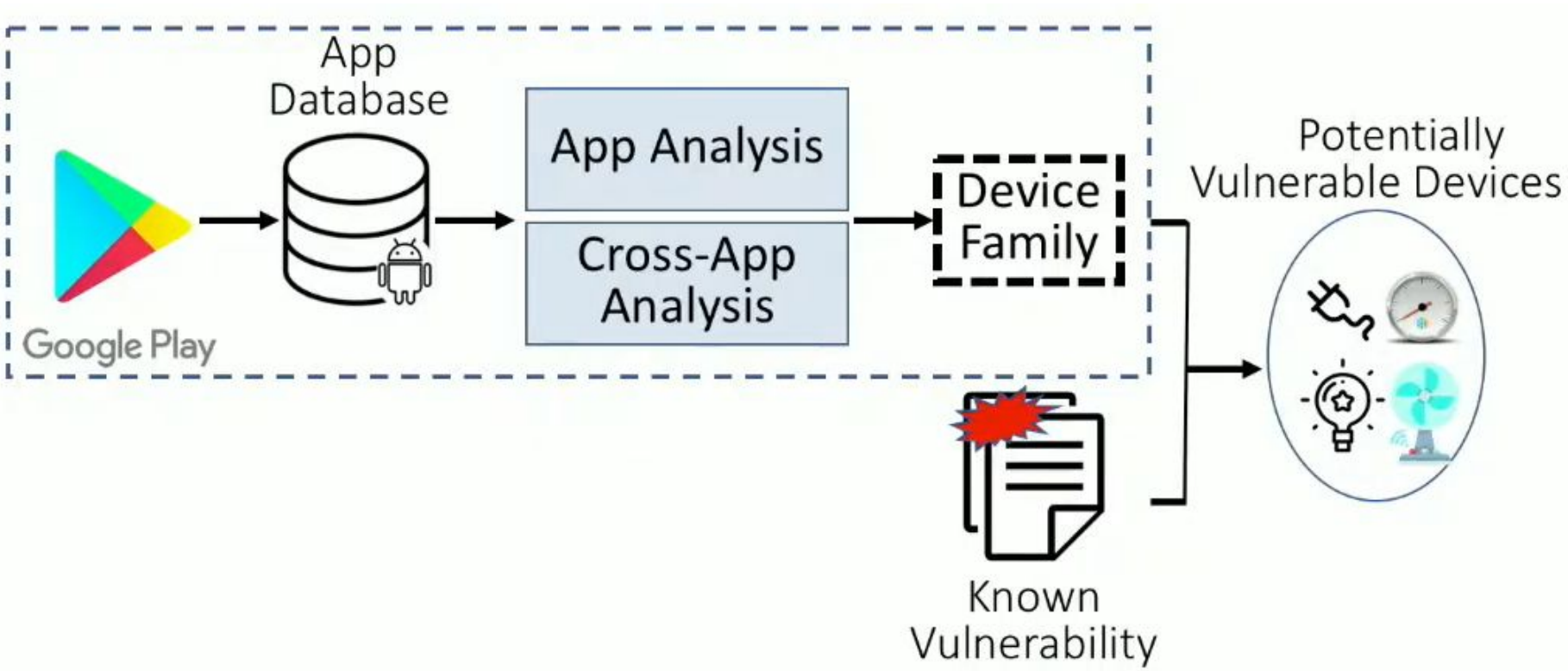
- Mobile companion apps are usually good estimation of the IoT devices.
- Similarity of IoT devices are reflected in their mobile companion apps.

Approach

A platform that helps to identify new vulnerable devices:

- Instead of analyzing devices directly, we use cross-app similarities to identify vulnerabilities that are transferrable among devices.
- No access to physical devices or their firmware, and thus scales better.

Architecture



App Analysis

Building **device profile** by analyzing **mobile apps**

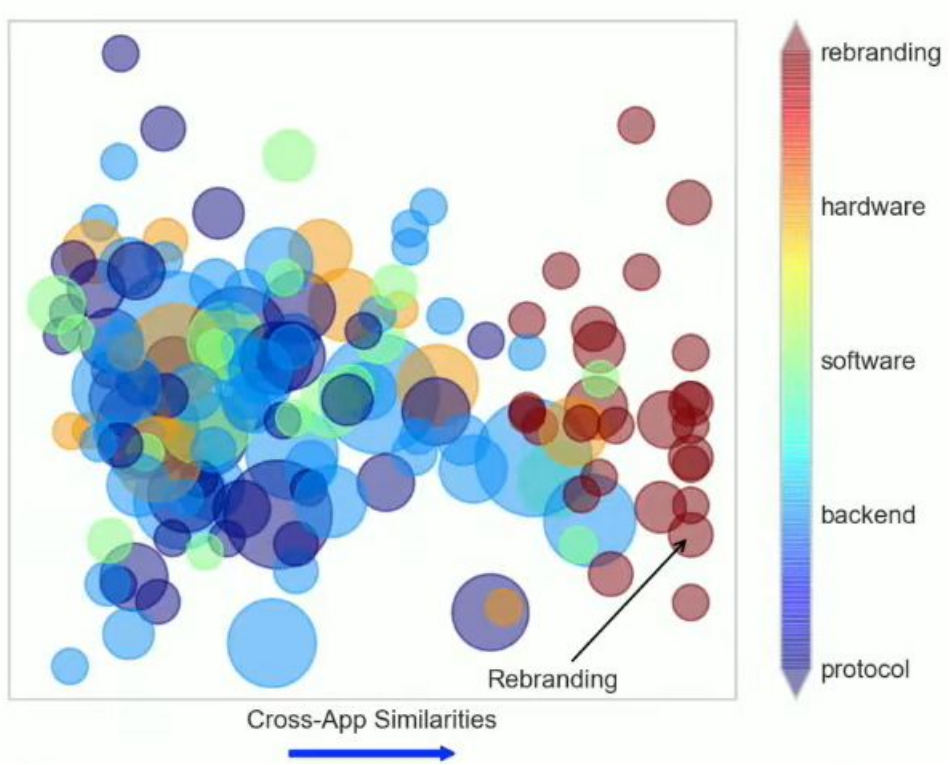
- Device interface (Network interface between device and app)
- Device imprints (Unique strings in apps to identify the device)
- Fuzzy hash (Code signature that calculated for the app code)

Cross-app Analysis

- Modular Similarity
  - Devices are not exactly identical, but shares components
  - Cluster the devices based on functional components of the apps

Device Families

- Each of the family shares a same component: software, hardware, protocol, backend, etc.



Takeaways

- IoT devices share components
  - Vulnerabilities transferrable among different devices

- Device similarities are reflected in their mobile companion apps.
  - App analysis provides an effective means to quickly identify vulnerable devices (and decides if a device maybe vulnerable to a specific vulnerability).

Current Problem

Functionalities of the devices are concerned more instead of software security to manufacturers, regardless the significantly increasing of usage of IoT devices.

Solutions in Future

Apply to Medical Devices:

- Apply access-control policy
- Cross-app analysis

Potential changes to FDA

- Forcing device manufacturers to take a security-based design approach to device development.
- Taking significantly quicker action in approving modifications to devices.
- Increasing data-sharing between manufacturers and health care organizations.