
Earlence Fernandes
Your car is a computer with wheels and an engine

Your refrigerator is a computer that keeps food cold

Your ATM is a computer with money inside

-- Bruce Schneier to the US House Committee on Energy and Commerce

2016
World, saved: Tootz the IoT unicorn farts rainbows at good news

Posted on April 19, 2016 in CONNECTED DEVICES

Tootz integrates with twitter, Facebook, twitch and gmail but will connect to more services in the future. It is Powered by USB or battery for fully untethered fun.
Automated Data Center Cooling Management

Demand Response; Increased Renewables Usage

Smart Cities

Data-Driven Agriculture

FarmBeats Platform, NSDI 2017

Courtesy: Microsoft Genome Project
Hospital Efficiency and Effectiveness

- Track meds for elderly
- Realtime location

Autonomous Vehicles

Wearables

Industrial Internet
We must address security problems in the Internet of Things.
Attacks on the Internet of Things

DDoS attack that disrupted internet was largest of its kind in history, experts say

Dyn, the victim of last week’s denial of service attack, said it was orchestrated using a weapon called the Mirai botnet as the ‘primary source of malicious attack’

Mirai botnet used IP Cameras/DVRs to launch DDoS

Mirai disabled heating for buildings in Finland, 200,000 residences lost power for 3 hours
Attacks on the Internet of Things

CULTURE

Hackers Killed a Simulated Human By Turning Off Its Pacemaker

JASON KEBLIER
Sep 7 2015, 12:45pm

Some humans are already hackable, and, yes, you can do some serious damage by compromising medical implants.

ANDY GREENBERG SECURITY 07.21.15 6:00 AM

Hackers Remotely Kill a Jeep on the Highway—With Me in It

0:07/6:07
Attacks Closer to Home

Remote determine prime time for Burglary [1,2] OR Flooding [1]

Devices

Protocols

[2] FTC Internet of Things Report’15
How might we tackle the IoT security problem?

What are the new intellectual challenges?
Device/Hardware Layer Challenges

Resource Constraints
(Energy, Hardware Features, Computation, ...)

- Privilege Levels, Memory Management Unit,
  Trusted Execution (SGX, TrustZone, ...),
  Secure Randomness, Secure Clocks, ...

How can we measure the passage of time? [1]

[1] A. Rahmati et al., Time and Remanence Decay in SRAM to implement secure protocols on embedded devices without clocks, USENIX Sec 2012
Device/Hardware Layer Challenges

- Core notions of hardware security mechanisms: Similar to other computing paradigms
- Resource Constraints of IoT devices => Affect higher-layer security properties
- Higher-layer security properties => Tuned to manage resource constraints

Hardware-Software Co-Design Approach
Network Layer Challenges

Connectivity Protocol Diversity

- Technology Infancy
- Environmental Constraints (e.g., no additional infrastructure)
- Resource Constraints (e.g., energy)

Affects Network Security Practices
As each protocol has its own notions of how two peers communicate with each other, it is unclear how network security practices such as port scanning translate to networks of devices that use various IoT protocols.
Repurposing Networking Tech. In New Ways

The hub-model of Smart Homes

Re-purpose the WiFi Router [1]

How do we make sure that only a WiFi-enabled presence detector and nothing else affects a WiFi door lock?

Can we patch security vulns at the network layer for unpatchable IoT devices?

Physical Principles for Network Anomaly Det.

Typical Network
General Purpose Computing Devices => Errors in Anomaly Detectors

IoT Network
Specialized Computing Devices => Possibly Less Errors

Physical devices/processes evolve as per physical laws.

Can we leverage this knowledge to build a model and then use it to reduce errors in anomaly detectors?
IoT Platform Layer Challenges

Process Isolation      Access Control      Information Flow Control      Updates      Authentication
IoT Platform Layer Challenges

- Process Isolation
- Access Control
- Information Flow Control
- Updates
- Authentication

Ultra-Resource Constrained Devices. E.g., sensors in a bridge, 64K RAM

Hail Dev Module

IMIX Dev Module

nRF51-DK Dev Module

Language Type Safety + Memory Protection Units = Tock OS [1]

[1] A. Levy et al., Ownership is theft: Experiences building an embedded OS in Rust, in PLOS’15
# IoT Platform Layer Challenges

<table>
<thead>
<tr>
<th>Challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Isolation</td>
</tr>
<tr>
<td>Access Control</td>
</tr>
<tr>
<td>Information Flow Control</td>
</tr>
<tr>
<td>Updates</td>
</tr>
<tr>
<td>Authentication</td>
</tr>
</tbody>
</table>
Analysis of SmartThings [1]

• What is SmartThings?
  • Home automation platform
  • Wirelessly control door locks, motion sensors, music players, ...
  • Supports third-party apps

• Why SmartThings?
  • Relatively Mature (2012)
  • 521 SmartApps
  • 132 device types
  • Shares design principles with other existing, nascent frameworks

SmartThings Primer

SmartThings Cloud Platform

Groovy-Based Sandbox
SmartDevice

Capability System
[Cmd/Attr]
[Events]

Groovy-Based Sandbox
SmartApp

Internet API
SMS API

HTTPS
GET/PUT

WiFi

Configure
Control

SmartThings Companion App

Groovy-Based Sandbox

SmartDevice

Groovy-Based Sandbox

SmartApp

Internet API
SMS API

HTTPS
GET/PUT

WiFi

Configure
Control

SmartThings Companion App

Groovy-Based Sandbox

SmartDevice

Groovy-Based Sandbox

SmartApp

Internet API
SMS API

HTTPS
GET/PUT

WiFi

Configure
Control

SmartThings Companion App
What makes this analysis challenging?

- Design Documents & Technical Reports
- Platform Analysis Toolchains
  - Dynamic Instrumentation
  - Static Analysis of Platform Code
- No public design documents
- Closed source: cannot use existing analysis toolchains
- Cloud platform has limited public interface
Analysis Methodology & Threat Model

SmartThings Cloud Platform

- Groovy-Based Sandbox
- Capability System
  - [Cmd/Attr]
  - [Events]
- Groovy-Based Sandbox

SmartDevice

SmartApp

HTTPS
GET/PUT

Internet API
SMS API

Black-box API Testing w/ Apps + Crash-Log Analysis (along 5 principles)

Static Code Analysis of SmartApps (our toolchain, our dataset)
## Security Eval. of SmartThings: Our Results

<table>
<thead>
<tr>
<th>Security Analysis Area</th>
<th>Finding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overprivilege in Apps</td>
<td>Two Types of <strong>Automatic Overprivilege</strong></td>
</tr>
<tr>
<td>Event System Security</td>
<td>Event <em>Snooping and Spoofing</em></td>
</tr>
<tr>
<td>Third-party Integration Safety</td>
<td>Incorrect OAuth Can Lead to Attacks</td>
</tr>
<tr>
<td>External Input Sanitization</td>
<td>Groovy <em>Command Injection</em> Attacks</td>
</tr>
<tr>
<td>API Access Control</td>
<td>No Access Control around SMS/Internet API</td>
</tr>
</tbody>
</table>

### Empirical Analysis of 499 Apps

- > 40% of apps exhibit overprivilege of at least one type (55%, 43%)

### Proof of Concept Attacks

- Pincode Injection and Snooping, Disabling Vacation Mode, Fake Fire Alarms
## Capability System

<table>
<thead>
<tr>
<th>Capability</th>
<th>Commands</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>capability.lock</td>
<td>lock(), unlock()</td>
<td>lock (lock status)</td>
</tr>
<tr>
<td>capability.battery</td>
<td>N/A</td>
<td>battery (battery status)</td>
</tr>
</tbody>
</table>

- **Send commands**
- **Read/set attributes**
- **Receive events**

**ZWave Lock SmartDevice**

- `capability.lock`
- `capability.lockCodes`
- `capability.battery`
  ...

### Usability
- Simpler Coarser Capabilities

### Ease of Development
- Expressive Functionality

### Security
- Fine-Grained Capabilities
Exploiting Design Flaws in SmartThings

- Command Injection
- OAuth Compromise
- Overprivilege
- Unrestricted SMS API
- Event Spoofing

Pincode Injection

Popular Existing SmartApp with Android companion app; Unintended action of setCode() on lock
Backdoor Pincode Injection Attack

**Mappings**

```python
Mappings = {
    path("/devices/:id"):
        action:
            [PUT: "updateDevice"]
}

def updateDevice():
    def cmd = request.JSON.command
    def args = request.JSON.arguments
    // code truncated
    device."$cmd"(*args)
```

**Dynamic Method**

```python
{command: 'setCode',
 arguments: [3, '3456']
}
```
Exploiting Design Flaws in SmartThings

- **Command Injection**
- **OAuth Compromise**
- **Overprivilege**
- **Unrestricted SMS API**
- **Event Spoofing**

**Pincode Injection**
- Popular Existing SmartApp with Android companion app; Unintended action of setCode() on lock

**Pincode Snooping**
- Stealthy malware SmartApp; ONLY requests capability.battery

**Disabling Vacation Mode**
- Malware SmartApps with no capabilities; Gives impression of reduced reliability

**Fake CO Alarm**
What did we learn from the attacks/analysis?

- **App-Device bindings** can be more precise without changing UX
  [Coarse SmartApp-SmartDevice Binding Overprivilege]
  - Fixing of event system overprivilege is a by-product

- Risk-based Capabilities/Permission => Fundamental Risk Asymmetry

- Permissions are only useful as a first line of defense for IoT platforms, can we do better?
IoT Platform Layer Challenges

<table>
<thead>
<tr>
<th>Process Isolation</th>
<th>Access Control</th>
<th>Information Flow Control</th>
<th>Updates</th>
<th>Authentication</th>
</tr>
</thead>
</table>

FlowFence [1] flow tracking is a first-class primitive

- Restructure apps in terms of information flows
  - Apps request point-to-point flows instead of individual permissions

  Camera data only used to activate door lock

- Language-level primitive to isolate and flow-track sensitive code

✓ Dynamic labeling scheme
✓ Programmer-defined tracking granularity
✓ Supports existing tools, languages, IDEs; no changes to OS

A Spectrum of Information Flow Tracking

Architecture Level
(Instructions, Gates)
Resource Overhead; Special Hardware
RIFLE, Execution Leases, ...

OS-Based DIFC
(Page/Process Level Tracking)
May Overaint; Coarse-Control
HiStar, Asbestos, Flume, ...

Language-Based DIFC
(Type Systems, Variable-Level Tracking)
Dev. Learning Curve; Limited Control over External Resources
Jif, Jeeves, ...

Challenge: Applying flow tracking principles to a specific domain

“Component-Level” DIFC
(Well-defined component-level tracking)
Combines PL & OS Techniques
Laminar, COWL, Aeolus ...
definition{
  name: "DemoApp", namespace: "com.testing",
  author: "IoTPaper", description: "Test App",
  category: "Utility"
}

//query the user for capabilities
preferences {
  section("Select Devices") {
    input "lock1", "capability.lock", title: "Select a lock"
    input "sw1", "capability.switch", title: "Select a switch"
  }
}
def updated() {
  unsubscribe()
  initialize()
}
def installed() {
  subscribe sw1, "switch.on", onHandler
  subscribe sw1, "switch.off", offHandler
}

def onHandler(evt) {
  lock1.unlock()
}
def offHandler(evt) {
  lock1.lock()
}
IoT Platform Layer Challenges

Updates should be careful and planned => Economic Impact or Worse

Cyber Incident Blamed for Nuclear Power Plant Shutdown

By Brian Krebs

A nuclear power plant in Georgia was recently forced into an emergency shutdown for 48 hours after a software update was installed on a single computer.

The incident occurred on March 7 at Unit 2 of the Hatch nuclear power plant near Baxley, Georgia. The trouble started after an engineer from Southern Company, which manages the technology operations for the plant, installed a software update on a computer operating on the plant’s business network.
IoT Platform Layer Challenges

Updates should be careful and planned => Economic Impact or Worse

IoT devices in the field could be intermittently powered => How to update during power losses?

IoT devices may not be updateable fundamentally [1] => no infrastructure was built by manufacturer

[1] T. Yu et al., Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things, HotNets-XIV.
IoT Platform Layer Challenges

- Process Isolation
- Access Control
- Information Flow Control
- Updates
- Authentication

- Weak Passwords
- Default Password (Mirai)
- Password Re-use

Client Side Password Strength Estimators
  e.g., https://github.com/dropbox/zxcvbn

GIZMODO
TV Report on Accidental Amazon Orders Triggers
Attempted Amazon Orders Across San Diego

Hudson Hongo
1/06/07 8:30pm - Filed by ALEXA
Application Layer Challenges

• Physical Co-Relations
  • E.g., Garage door closes, nearby speaker picks up acoustic pattern
  • E.g., Vehicle speed increases, change in engine vibration patterns

• Machine Learning [1] for Control
  • E.g., Robots
  • E.g., Autonomous Vehicles

IoT Security Research: A Rehash of Old Ideas or New Intellectual Challenges?
E. Fernandes, A. Rahmati, K. Eykholt, A. Prakash
arXiv 2017

https://web.eecs.umdich.edu/~earlence/

Consider Submitting

https://iotsecurity.eecs.umdich.edu

https://www.safethings.info/