Vigilia: Securing Smart Home
Edge Computing

Symposium on Edge Computing 2018
Oct 25-26, 2018 – Bellevue, WA

Rahmadi Trimananda, Ali Younis, Bojun Wang, Bin Xu, Brian Demsky | UC Irvine
Harry Xu | UCLA
Vulnerable Smart Home IoT Devices

CVE-2016-5053
Access via TCP port 4000

CVE-2017-6520
Access via UDP port 5353

CVE-2018-3911
HTTP header injection

CVE-2012-3002
Bypassed authentication

CVE-2013-6949
Improper STUN/TURN

http://support.smartthings.com/,
http://www.amazon.com/
SmartThings
Platform for Smart Home IoT Devices

[Diagram showing a Smart Hub connected to various devices like Smart Speaker, Motion Sensor, SmartApps, and Device Handlers]
Enhanced Auto Door Lock
SmartApp Example

Enhanced Auto Lock Door

by Arnaud
Automatically locks a specific door after X minutes when closed and unlocks it when open after X seconds.

Kwikset SmartCode 910
ZigBee

SmartThings Multipurpose Sensor
ZigBee

Enhanced Auto Door Lock
SmartApp Example

SmartThings Cloud Server

Front door

Enhanced Auto Door Lock
SmartApp Example
Enhanced Auto Door Lock
SmartApp Example

- Door Lock (Device Handler)
- Open/Close Sensor (Device Handler)

ST Cloud Server

Front door

Example:
Open/Close Sensor
Enhanced Auto Door Lock SmartApp Example

- Install SmartApp
- Enhanced Auto Lock Door
- Choose door lock and sensor
Enhanced Auto Door Lock
SmartApp Example

preferences{
    page name: "mainPage", install: true, uninstall: true
}
def mainPage() {
    dynamicPage (name: "mainPage") {
        section("Select the door lock:" ) {
            input "lock1", "capability.lock", required: true
        }
        section("Select the door contact sensor:" ) {
            input "contact", "capability.contactSensor", required: true
        }
        section("Automatically lock the door when closed..."") {
            input "minutesLater", "number", title: "Delay (in minutes):", required: true
        }
        ...
    }
}

- capability for security
- SmartApp can only bind with and control certain devices, e.g., capability.lock
Enhanced Auto Door Lock SmartApp Example

```python
def lockDoor():
    log.debug "Locking the door."
    lock1.lock()
...

def unlockDoor():
    log.debug "Unlocking the door."
    lock1.unlock()
...

def doorHandler(evt):
    if ((contact.latestValue("contact") == "open") && (evt.value == "locked")) {
        ...
    }
...
```

- **capability** for security
- **SmartApp** can only control certain device features, e.g., `lock1.lock()`
Enhanced Auto Door Lock SmartApp Example

ST Cloud Server

Door Lock (Device Handler)

Open/Close Sensor (Device Handler)

Door Lock Controller App

Front door

Example
SmartThings Is Not Secure!

• Capability model breaks down
  • It is easily subverted!

• SmartThings prone to attacks
  • WiFi device attack
  • Cloud server attack
  • Bad SmartThings code attack
# WiFi Device Attack

<table>
<thead>
<tr>
<th>Device</th>
<th>Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blossom sprinkler</td>
<td>Unauthenticated API access via port 80</td>
</tr>
<tr>
<td>LIFX light bulb</td>
<td>Unauthenticated access via port 56700</td>
</tr>
<tr>
<td>iHome speaker</td>
<td>Unauthenticated access via port 80</td>
</tr>
<tr>
<td>Amcrest camera</td>
<td>Weak authentication for video stream via port 80</td>
</tr>
<tr>
<td>D-Link siren</td>
<td>Brute-force-able PIN guessing via port 80</td>
</tr>
</tbody>
</table>

Cloud Server **Attack**

- Excessive access to cloud servers
- Arbitrary network access
- Smart hubs bypass router firewall “legally”!
Bad SmartThings Code Attack

• Device capability has security in mind, but flawed!
  • It restricts access based on capabilities
  • But, not enforced at network level

• Device handler code could be made to conspire with SmartApp
  • Spy on SSDP traffic of other devices
  • Communicate with arbitrary IP and ports
  • Send commands to arbitrary devices
Enhanced Auto Door Lock SmartApp Example

- Door Lock (Device Handler)
- Open/Close Sensor (Device Handler)
- Door Lock Controller App
- ST Cloud Server

Front door
Enhanced Auto Door Lock
SmartApp Example

- Door Lock (Device Handler)
- Open/Close Sensor (Device Handler)
- Door Lock Controller App
- ST Cloud Server

This device handler now CONTROLS the camera!
Enhanced Auto Door Lock SmartApp Example

Security Problem!

Threat Model

• Devices have vulnerabilities

• Attackers have full knowledge of the system

• Attackers have access to the home network via compromised device
  • Not physical access
Vigilia

• Why not just fix SmartThings?
  • SmartThings is a closed solution
  • None of its source code is available
  • SmartApps run on SmartThings cloud

• Vigilia is an open-source implementation of SmartThings
  • Improved security aspect of SmartThings
  • Managed communication through cross-layer techniques
Vigilia Handles Excessive Access

SmartThings has

- **Configuration**
  - Install/register device
  - Binding with device handler

- **Capabilities**
  - Which specific device handler?
  - Which specific feature?
  - Binding with app

- **Restrict** communication at network level!
Typical Home Network

Problems

- Devices have no unique secrets
  - Can spy on packets sent to other devices
  - Can masquerade as other devices or even router
  - Can lie about MAC or IP
- Devices send packets directly to other devices
  - without going through the firewall

23/10/2018  https://www.amazon.com/
Vigilia Network

• Assigns
  • a unique WiFi password
  • to each WiFi device

Result
• Devices can’t spy on traffic between devices!
Vigilia Network

• Vigilia uses hostapd
  • to lock MAC address
  • to specific WiFi password

Result

• Devices can’t lie about MAC addresses!

23/10/2018 https://www.chittagongit.com/
Vigilia Network

• Vigilia isolation + hairpin
  • force all communications to go through firewall
  • firewall locks IP to MAC

Result

• Devices can’t communicate unless firewall allows
• Devices can’t lie about IP addresses
SmartThings Device Model

Problem

• Device handlers have excessive network access
  • TCP/IP handlers can specify and connect to any IP + port
  • Zigbee handlers can specify and connect to any Zigbee device address
  • All handlers can see SSDP traffic
Vigilia Device Model

- Vigilia tracks local devices’ IP addresses
- Vigilia gives drivers access to devices via capability
- Capabilities only allow communication with specific devices
  - Drivers only specify which devices
  - Runtime assigns driver IP + port / Zigbee address
  - Runtime can confidently enforce firewall rules without breaking
Vigilia Configuration

• Configuration contains two types of binding

(1) App to device handler/driver

(2) Device handler/driver to device
Vigilia Configuration

• Configuration contains two types of binding

(1) App to device handler/driver

(2) Device handler/driver to device
Vigilia Configuration

- Configuration contains two types of binding

1. App to device handler/driver

2. Device handler/driver to device
Securing App to Device Driver Binding

- **Isolate** components in sandbox
  - Lock to files + IP + port

- **Filter request**
  - At destination for capability access

- **Use firewall rules**
  - Allow specified communications
  - Block everything else
Securing App to Device Driver Binding

• **Isolate** components in sandbox
  • Lock to files + IP + port

• **Filter request**
  • At destination for capability access

• **Use firewall rules**
  • Allow specified communications
  • Block everything else
Securing TCP/IP Devices

• Device driver capability
  • Use firewall rules
  • Allow specified TCP/IP communications
  • Block everything else
Securing TCP/IP Devices

• Device driver capability
  • Use firewall rules
  • Allow specified TCP/IP communications
  • Block everything else
Securing Zigbee Devices

• **Device driver capability**
  • Zigbee packet filter on Zigbee gateway
  • Multiple Zigbee drivers can talk to gateway
  • Only the right one can send packets to device
Securing Zigbee Devices

• Device driver capability
  • Zigbee packet filter on Zigbee gateway
  • Multiple Zigbee drivers can talk to gateway
  • Only the right one can send packets to device
Vigilia Guarantees

• All communications from non-malicious apps will be allowed

• All communications not explicitly configured are blocked
### Experience

<table>
<thead>
<tr>
<th>Vigilia App</th>
<th>Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
<td><strong>Output</strong></td>
</tr>
<tr>
<td>Irrigation</td>
<td>Soil moisture sensor (Zigbee)</td>
</tr>
<tr>
<td></td>
<td>Weather report website</td>
</tr>
<tr>
<td></td>
<td><a href="https://openweathermap.org/">https://openweathermap.org/</a></td>
</tr>
<tr>
<td>Lights</td>
<td>Cameras</td>
</tr>
<tr>
<td>Music</td>
<td>GPS (smartphone)</td>
</tr>
<tr>
<td>Home security</td>
<td>Motion, water-leak, multipurpose sensors (Zigbee)</td>
</tr>
<tr>
<td></td>
<td>Camera</td>
</tr>
</tbody>
</table>
# Attacks

<table>
<thead>
<tr>
<th>Attack</th>
<th>Application</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprinkler</td>
<td>Sprinkler</td>
<td>Run API via port 80 (HTTP)</td>
</tr>
<tr>
<td>Light bulb</td>
<td>Lights</td>
<td>Turn on/off via port 56700</td>
</tr>
<tr>
<td>Speaker</td>
<td>Music</td>
<td>Play music via port 80 (HTTP)</td>
</tr>
<tr>
<td>Camera</td>
<td>Home Security</td>
<td>View camera via port 80 (HTTP)</td>
</tr>
<tr>
<td>Siren/Alarm</td>
<td>Home Security</td>
<td>Brute-force PIN &amp; access via port 80 (HTTP)</td>
</tr>
<tr>
<td>Deauthentication</td>
<td>All</td>
<td>Jam WiFi access &amp; let device join a malicious WLAN router</td>
</tr>
</tbody>
</table>
# Attacks

<table>
<thead>
<tr>
<th>Attack</th>
<th>Normal*</th>
<th>IoTSec</th>
<th>Vigilia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprinkler</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Light bulb</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Speaker</td>
<td>✓</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Camera</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Siren/Alarm</td>
<td>✓</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Deauth. + Sprinkler</td>
<td>N/A</td>
<td>N/A</td>
<td>X</td>
</tr>
<tr>
<td>Deauth. + Light bulb</td>
<td>N/A</td>
<td>N/A</td>
<td>X</td>
</tr>
<tr>
<td>Deauth. + Speaker</td>
<td>N/A</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Deauth. + Camera</td>
<td>N/A</td>
<td>N/A</td>
<td>X</td>
</tr>
<tr>
<td>Deauth. + Siren/Alarm</td>
<td>N/A</td>
<td>✓</td>
<td>X</td>
</tr>
</tbody>
</table>

✓ = attack success  
X = attack thwarted  

*Normal = standard router, including Norton Core and Bitdefender Box 2
Public IP Experiment

• 16 smart home devices  
  • Exposed to the Internet – public IP  
  • Duration of 10 days  

• Total of 38,296 access attempts  
  • TCP (e.g., TCP SYN/ACK)  
  • UDP  
  • ICMP
Public IP Experiment – Cameras

• Four Amcrest cameras – 14 hours of exposure

• With Vigilia – only 551 attempts
• With password only – 31,230 attempts

• No protection
  • All 4 disabled in 15 minutes!
  • 172 – 362 packets per camera
  • XML-RPC attack via HTTP (port 80)
Conclusions

• Smart home IoT devices have vulnerabilities
• Cannot manage security for individual (simplistic) devices
• Manage the communications!

• Download: http://plrg.eecs.uci.edu/vigilia/

Please find more details in the paper!
Thank you! 😊