Discovering and Understanding the Security Hazards in the Interactions between IoT Devices, Mobile Apps, and Clouds on Smart Home Platforms

Zhou et al.
Slides by Thomas Quig

Announcements

- TracerFIRE, you can probably late register

- Spray Paint Social When2Meet

- :)

Meeting Flag

sigpwny{intern3t_of_thonk}

Summary

Smart home platforms consist of three entities that interact.

1. Cloud infrastructure
2. The devices themselves
3. A management interface (most often a mobile application)

These entities connect with each other, which allows for IoT to work

But it comes with **security risks,** this paper explores those

Novel Contributions

Most other papers only focus on subsections of smart home platforms.

This paper studies IoT smart home platforms wholistically

Existing papers pay attention to traditional security issues

This paper studies IoT specific vulnerabilities (entity-entity interactions)

Other papers define operations differently

This paper defines interactions as "inter-operations" between entities

Background

Smart Home Platforms

- Cloud is the Brain, provides automation services and computation
- IoT devices are the muscle, equipped with sensors that interact with the physical world and send data to the IoT cloud
- Mobile apps are also the brain, but the decision making part of the brain.

Two Types of Devices

- Directly connected to internet (Type 1)
- Connected to internet *through a hub device* (Type 2)

IoT Interactions-Summary

1. Device Discovery (Find the stuff)
2. WiFi Provisioning (Let the stuff connect)
3. Device Registration (Register the stuff with the cloud)
4. Device Binding (Associate the stuff with your account)
5. Device Login (Make sure only you can control the stuff)
6. Device in Use (Do things with the stuff)
7. Device Unbinding/Reset (Goodbye stuff)

Device Discovery

1. Device comes online
2. User “Adds Device” on the control application
3. App connects with the device by using discovery message
   a. Type 2 Platforms must go through a hub device
4. Device reports basic info (MAC addr, model)

WiFi Provisioning

1. User shares internet access with the device
   a. Credentials
   b. AP Mode
   c. “SmartConfig”

2. Device connects to internet and same LAN as app.

ʃ つ•_• ʅつ

“give wifi pls”

Device Registration

1. Device is given an ID
   a. For Type 1 (Direct to cloud), device sends its information to the cloud.
   b. For Type 2, ID is hard-coded into the hardware, and is thus skipped

2. Cloud (Type 1) responds back with specific Device ID, stores this ID

3. Device (Type 1) writes the Device ID to its memory

Device Binding

1. Cloud binds Device ID to owner account.
   a. Type 1: Binding req is sent by the mobile app to the cloud.
   b. Type 2: Device info is sent to the account, then **the account** makes the binding request to the cloud.

Cloud **unconditionally accepts** the binding request because of assumption that customer who physically owns device has full control over it.

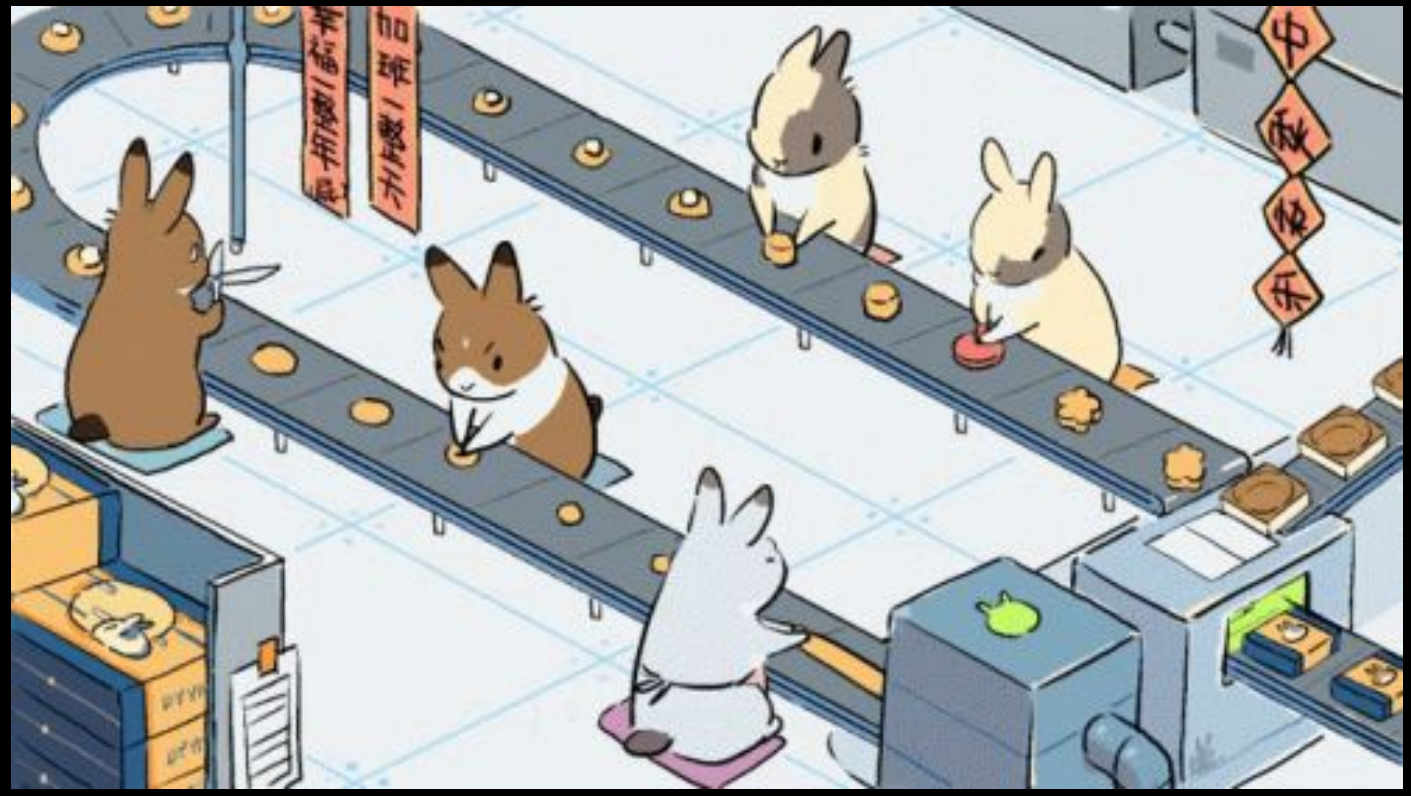
Device Login

1. Device uses its device ID to log into Cloud
2. Device establishes connection to ready itself to execute commands

When devices lose connection, the reconnect automatically

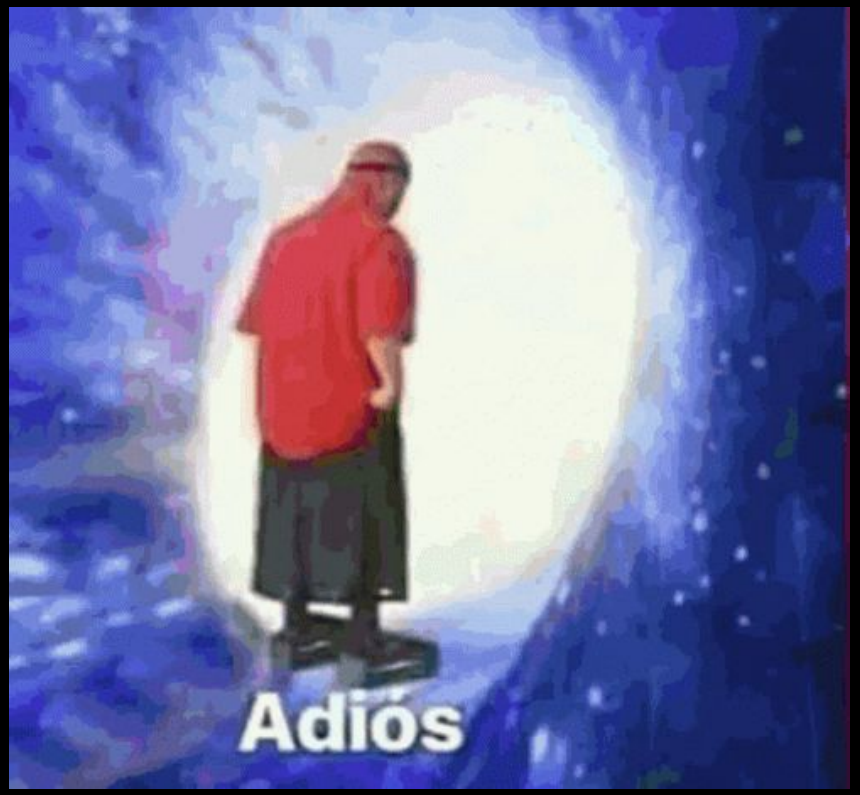
Device In Use

1. Once everything is done… the device operates as intended.

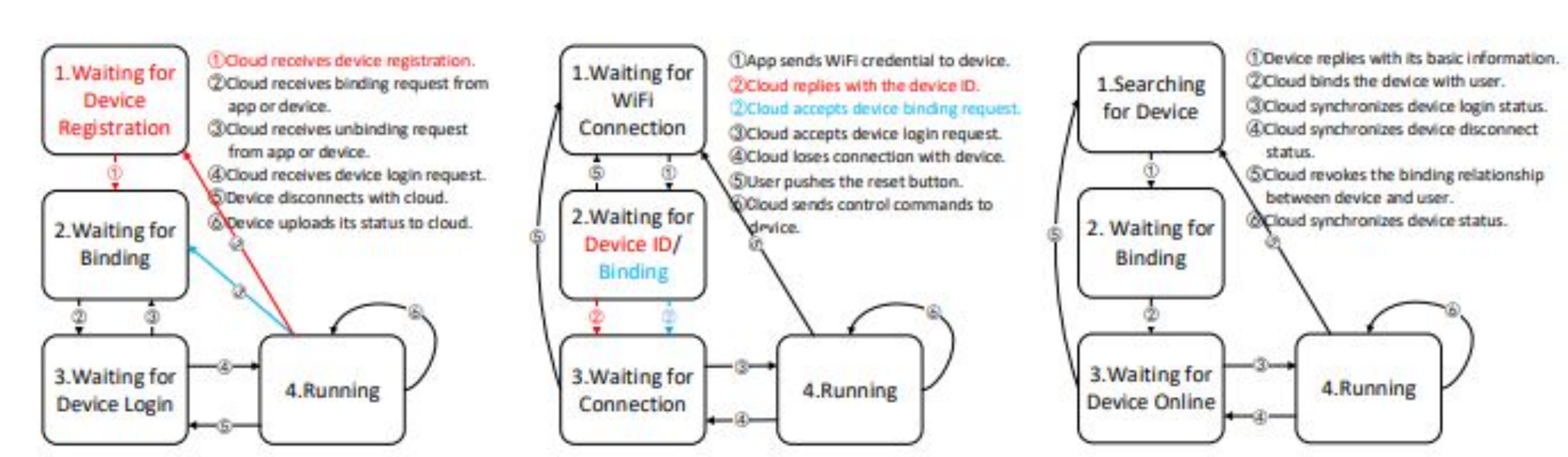


Device Unbind / Reset

1. Devices can be manually unbound
   a. Type 1 - Cloud directly erases binding info
   b. Type 2 - Command sent from cloud to device to erase binding info



IoT Interactions - State Transitions



Type 1-specific is in red

Type 2-specific is in blue

Methods

- Goal information
  - Adversaries intend to get three kinds of information
  - Public vs Guessable vs and Hard-Coded
- Threat model?
  - Physical Access?
  - Identifying information / Legitimacy Information
- Analysis Methods (attack flow)
  - Decipher Communications (Cloud-App, Device-Cloud, Device-App)
    - MITM most common attack vector
  - Understand the interaction messages
  - Create test devices (phantom devices)
    - Devices that use Device-Side SDKs, make IoT device behave like Burp suite

Devices Used

Five widely used smart home platforms.

1. Smart-Things (website)
2. KASA (website)
3. MIJIA (catalogue)
4. Alink (website)
5. Joylink (amazon catalogue)

Results - Summary

Four kinds of vulnerabilities

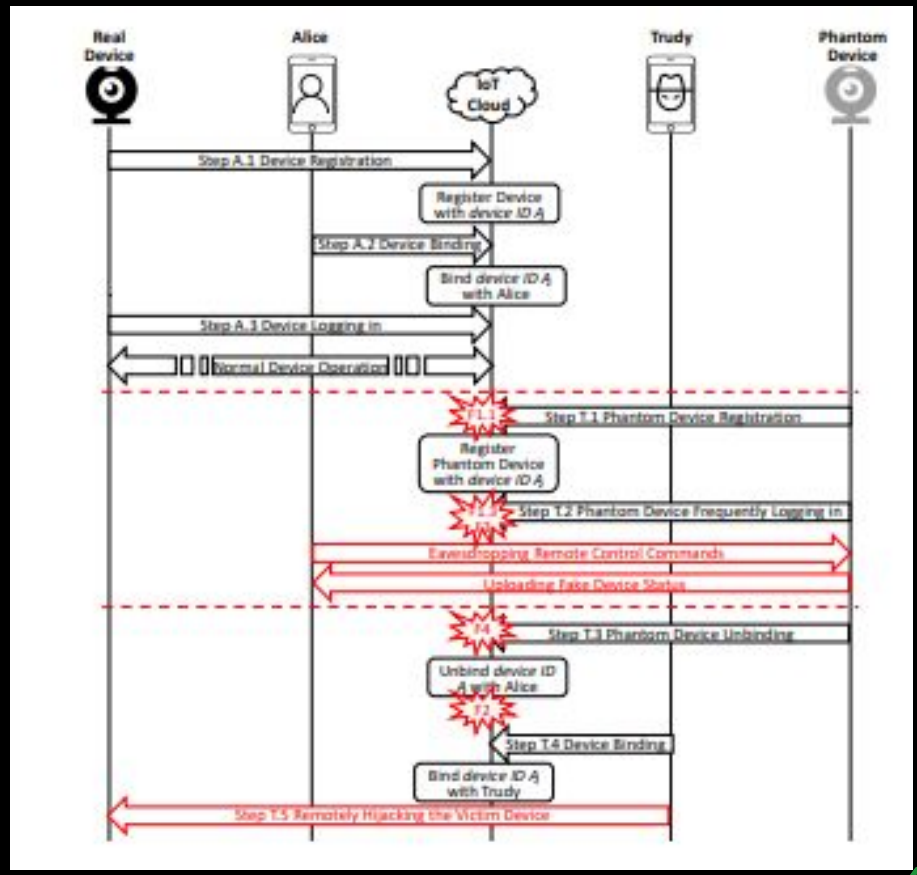
1. Insufficient State Guard
   a. Use strange requests to enter an invalid state.
   b. Cloud often just accepts these requests
2. Illegal State Combination
   a. Combine states to violate security assumptions (hijacking attack)
   b. Synchronization & Race conditions.
3. Unauthorized Device Login
   a. Connect to an account with a fake device
4. Unauthorized Device Unbinding
   a. Send a unbind request from a fake device

Results - Exploitation (Substitution)

1. Obtain legitimate information
   a. Get MAC, CID
   b. TP-LINK led to MITM right away

2. Create phantom device using obtained information

3. Substitute real device with phantom device



Attack Method - Substitution

Type I

1. Sniff the Device ID when target binds the device.
2. Log in with the phantom device WITHOUT trudy's credentials.
3. Have device login rapidly to "beat" the other device.

Type II

1. Sniff the Device ID when target binds the device.
2. Log in with the phantom device WITHOUT trudy's credentials.
3. Have device login rapidly to "beat" the other device.

The difference is how the device IDs are sniffed
Type 1 = from client, Type 2 = from cloud

Implications - Substitution Attack

Privacy Breaches
- Commands meant to go to real device go to adversary

Falsified Data
- Data from real device is intercepted by adversary, can be modified in any desired way.

Stealthy (No way for target to tell)

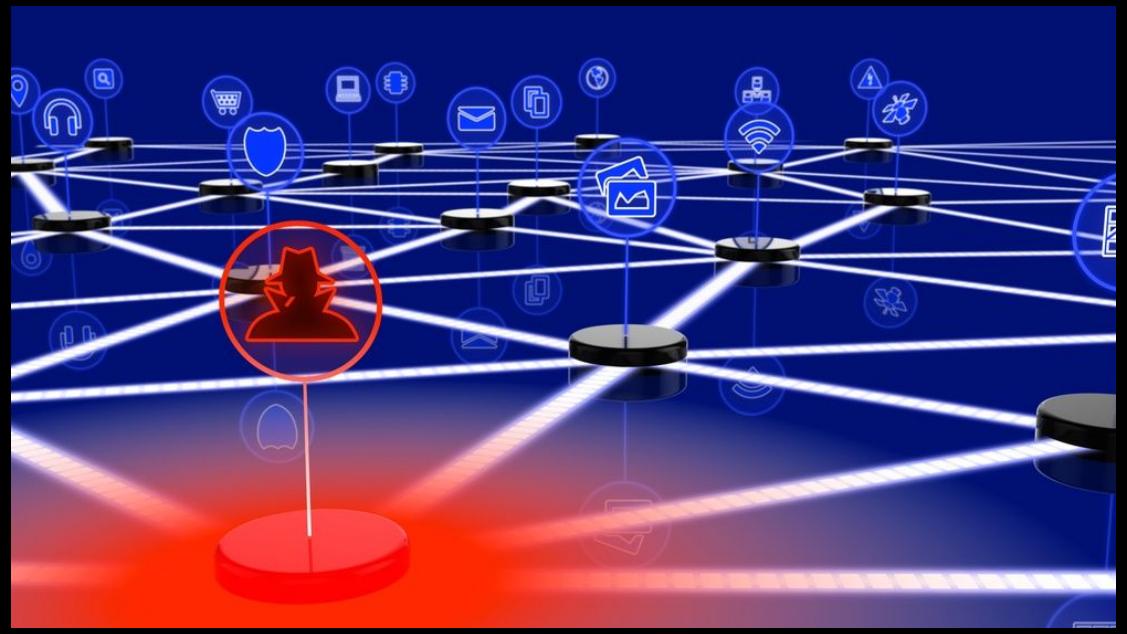
IoT device suffering from falsified data attack

Results - Exploitation (Remote Device Hijacking)

TLDR: Deauthentication attack

Consequence: Adversary rebinds device to their account.

Not as stealthy (Target could notice missing device)



Results - Other Security Hazards

Remote DoS

- Unbind or takeover attacks can lead to a DoS because owner has no control.

Illegal Device Occupation

- Predict device ID's of **unsold devices**, take them over on initial registration.

Firmware Theft

- Leaked firmware can allow for more research into device operations.
- This was done successfully thousands of times in this paper.

Solutions

Cloud-Free Platforms

- Apple HomeKit

DIY IoT Platforms

- Open-Source platforms are becoming more common
- “Security through obscurity”
  - Adversarial shifting

Technical Solutions

- Strict authentication
- Comprehensive authorization
- Enforcing valid state transfer

Commonly Asked Questions

Is device fingerprinting (TLS Pinning etc) a feasible defense?

What about network isolation? Does homekit solve everything?

Is there a way to add additional authentication to key points such as unbinding or ownership transfer?

Discussion Questions

Should regulation be implemented on IoT devices?

- Is regulation possible due to the widespread nature of IoT devices?
- How would the IoT industry change if regulation is enforced?

Next Meetings

Next Thursday: Windows Environments

- The one OS we never like talking about
- How to attack windows boxes and environments
- CME, Hydra, etc.

Sunday Seminar: Open (UIUCTF Planning)

- If you have a topic, reach out to us!
- If nothing, we will do UIUCTF planning
-