Problem Statement and Goals

**Problem:**
- Home IoT device boom:
  - Future smart homes will be highly Internet-connected
  - Developers & users have no understanding of security
  - Vendors have no incentive to provide security
  - Influx of IoT devices leads to decreased security

**Goal:**
- Analyze home IoT devices to demonstrate security & privacy risks

Methodology

**Devices Analyzed:**
- Geeni Lux 800 Smart Light Bulb
- iBaby Monitor M6S
- Swift Robo Buddy

**Analysis For Each Device:**

**Geeni Lux 800 Smart Light Bulb:**
- Open port 6668 (Tuya API)
- Overinclusion of Tuya API in app
- Unused source code for other IoT devices
- Unchangeable LAN mode login password 12345678
- Default WiFi mode login admin:123
- Username & remote ID are static
- Information sent in plain text
- Nearby SSIDs
- Configuration data
- acSMTP login creds for Chinese email account

**iBaby Monitor M6S:**
- Many improvements over earlier M3S model
- Network traffic HTTPS & AES-encrypted
- Port 23 (telnet) open
- credentials not trivially brute-forceable
- APK source heavily protected with Tencent packer
- Found update daemon
- Firmware not captured from traffic (sockets)
- No hard-coded URIs

**Swift Robo Buddy:**
- Port 23 (telnet) open and login is root:jvbzd
- From log file on the device
  - Type: [jv] User: [admin] IP: [45.33.92.153]

**Possible attacks:**
- Enumerate IDs and brute-force login
- Arbitrary execution via wget
- Possible vulnerabilities on the communication server 45.33.92.153

Results

**Future Work**

**General Future Work:**
- Use methodology for other IoT devices
- Quantify the overinclusion of unused source code

**Further Research topics:**
- How can an attacker take advantage of the interactions of IoT devices on a home network?
- How could an attacker take advantage of overinclusion of unused source code or hardware in IoT devices?
- Analyze communications between lightbulb and APK
- Develop Tencent unpacker
- Develop remotely-exploitable attacks