Towards Secure Smart Home IoT: Manufacturer and User Network Access Control Framework

Mohammed Al-Shaboti, Ian Welch, Aaron Chen, Muhammed Adeel Mahoud
**Smart home IoT**

**IoT** is a paradigm where things have *identification*, *sensing*, *processing* capabilities and able to *communicate* with each other and with the Internet to perform a task (Thakar et al, 2016).

**Smart home IoT** form 61% of the total number of IoTs in 2017 (Gartner, 2017).

**Smart home IoTs** help to facilitate different daily home activities.

- Security system (e.g. IP-camera, door locks)
- Home appliances (e.g. smart fridge/bulbs)
- Home entertainment (e.g. stereo system)
Security problems (smart home)

➢ **Vulnerable and easy to be exploit**
  - Out of 32 examined smart home IoT devices for security and privacy risk, result shows 9 with low and 4 with high and 19 with moderate risks (Jacobsson et al., 2016).
  - HP report shows 70% of IoT devices used unencrypted network protocols, and 60% of IoT user interface are vulnerable and 90% collect personal information (HP ent., 2015).

➢ **Misuse of IoT devices**
The Mirai botnet is mainly formed from IoT devices (i.e. CCTV run telnet), is used to launch a DDOS attack against Dyn DNS provider with a peak of 665 Gbps (KREBSONSECURITY, 2016).

➢ **Compromise user privacy**
RAPID7 examined baby monitors IoT from seven vendors (Philips, iBaby Labs, Summer Infant) and found that all of them are vulnerable (e.g. unauthorized access/default credentials) (Mark et al., 2015).
Example: IP-cameras!

Camaras Online NZ - harveynorman.co.nz

Watch live surveillance online IP cameras in Un

Watch live surveillance online IP cameras in Ne

Thousands of cameras and security systems available to view fo
Causes of vulnerabilities

- **IoT market**
  - No regulation
  - Some vendors are new to the security
  - Fragmented IoT supply chain (whom to blame!)

- **Current status**
  - No automatic update
  - *Unpatchable devices*
  - Default/hardcoded credentials

- **Security solutions challenges**
  - Diverse standards and technology
  - Resource constrained devices
  - Social factor
  - Lack of network administrator
Manufacturer Usage Description

The goal of MUD is to provide a means for Things to signal to the network what sort of access and network functionality they require to properly function.

IETF standard in draft state (IETF, 2018).
Manufacturer Usage Description

- Reduce threat surface on device entering a network to communications intended by the manufacturer
- Provide means to address some vulnerabilities faster than time to update systems
- Keep cost of implementation to a bare minimum
- Provide means of extensibility for manufacturer to express device capabilities or requirements
Example

A light bulb is intended to light a room. It may be remotely controlled through the network, and it may make use of a rendezvous service of some form that an application on a smart phone.

What we can say about that light bulb, then, is that all other network access is unwanted.

It will not contact a news service, nor speak to the refrigerator, and it has no need of a printer or other devices.

It has no social networking friends.

Therefore, an access list applied to it that states that it will only connect to the single rendezvous service will not impede the light bulb in performing its function, while at the same time allowing the network to provide both it and other devices an additional layer of protection.
What MUD doesn’t do?

What if our light bulb user does want to open up communications with another device?
What if our light bulb needs restrictions related to time?
What if our light bulb needs restrictions related to how often it turns on and off?

MUD needs to supplemented by some form of discretionary access control allowing user to provide fine-grained control and override where necessary some of the rules.
Beyond Static Access Control

What is we want more than a yes/no answer?

We also want to restrict connectivity to network or turn on logging etc. when suspected attack.

Might want to prevent certain types of attack (for example, ARP spoofing – prevent man-in-middle attack at the MAC level).
Enforcement of MUD + DAC

MUD is a proposed standard, it's not a technology. Technology agnostic.

Host-centric solutions require changes to the devices.

Network-centric solutions:

- Use a proxy for the device (heavyweight)
- Firewall controls at the perimeter (traditional perimeter approach)
- Apply software defined networking for control (lightweight + enforced at the access layer)
Role of Software-defined Networking (SDN)

- SDN separates control plane from data plane and provides standardised API between them.

- SDN helpful features:
  - On demand access control
  - Network-wide visibility.

Enforces fine-grained policy on the network perimeter and within the network.
SDN Challenges

➢ Control channel between SDN controller and switches are limited
➢ Performance degrades quickly the more the controller is involved (reactive mode)
➢ Can be focus of a denial-of-service attack

➢ Our focus is reactive and off load to servers providing network function virtualisation.

➢ Use Dataplane development kit (DPDK) for NFV.
Prototype Architecture

SDN-based security architecture

User control panel → Security agent → SDN Controller

MUD → Security agent

Process/Dispatch → Data plane

NFV → security services
**Goal**: to evaluate the functionality of the architecture.

The malicious IoT device scan the network in three scenarios:

1. Without any access control
2. With MUD policy
3. With DAC access control
No policy

Results (without policy)

➢ TCP port **LAN scan** (nmap -sS -Pn 192.168.10.14,58,64,254)

<table>
<thead>
<tr>
<th>Target</th>
<th>Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optiplex 9020)</td>
<td>22/ssh, 139,445, 5000/(HTTP)</td>
</tr>
<tr>
<td>Net-PC</td>
<td>135, 139,445, 6000</td>
</tr>
<tr>
<td>Android</td>
<td>None</td>
</tr>
<tr>
<td>piDHCPserver</td>
<td>22/ssh, 53/domain</td>
</tr>
</tbody>
</table>

➢ TCP port **WAN scan** (nmap -sS 198.148.81.138)

<table>
<thead>
<tr>
<th>Target</th>
<th>Services</th>
<th>MUD</th>
</tr>
</thead>
<tbody>
<tr>
<td>hackthissite.org</td>
<td>22/ssh, 443/HTTPS, 80/HTTP</td>
<td>None</td>
</tr>
</tbody>
</table>
TCP port **WAN scan** (nmap -sS 198.148.81.138)

<table>
<thead>
<tr>
<th>Target</th>
<th>Services</th>
<th>MUD</th>
</tr>
</thead>
<tbody>
<tr>
<td>hackthissite.org</td>
<td>(22, 80) filtered, 443/HTTPS open</td>
<td>ip-cam.json</td>
</tr>
</tbody>
</table>

TCP port **LAN scan** (nmap -sS -Pn 192.168.10.14,58,64,254)

<table>
<thead>
<tr>
<th>Target</th>
<th>Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optiplex 9020)</td>
<td>22/ssh, 139,445, 5000/(HTTP)</td>
</tr>
<tr>
<td>Net-PC</td>
<td>135, 139,445, 6000</td>
</tr>
<tr>
<td>Android</td>
<td>None</td>
</tr>
<tr>
<td>piDHCPServer</td>
<td>22/ssh, 53/domain</td>
</tr>
</tbody>
</table>
MUD + DAC

Results (with MUD & user policy)

➢ TCP port LAN scan after DAC (nmap -ss -Pn 192.168.10.14,58,64,254)

<table>
<thead>
<tr>
<th>Target</th>
<th>Status</th>
<th>Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optiplex 9020)</td>
<td>Up</td>
<td>5000/(HTTP)</td>
</tr>
<tr>
<td>Net-PC</td>
<td>Up</td>
<td>None</td>
</tr>
<tr>
<td>Android</td>
<td>Up</td>
<td>None</td>
</tr>
<tr>
<td>piDHCPServer</td>
<td>Up</td>
<td>None</td>
</tr>
</tbody>
</table>

➢ TCP port WAN scan (nmap -ss 198.148.81.138)

<table>
<thead>
<tr>
<th>Target</th>
<th>Services</th>
<th>MUD</th>
</tr>
</thead>
<tbody>
<tr>
<td>hackthissite.org</td>
<td>(22, 80) filtered, 443/HTTPS open</td>
<td>ip-cam.json</td>
</tr>
</tbody>
</table>
Related work

- IETF, MUD
- Xu et. al, 2016, in-bound filtering.
- Feamser, 2010, outsourcing.
- Lorenz et. al., 2017, SDN & NFV firewall
- Yu et.al, 2015, Micro VM proxy
- Lara et. al. 2016, opensec
- Luo et. al, 2016, NFV agent
- Cyber reboot, 2017, Poseidon

- Gharakheili, 2016, QoS control
- Chetty, 2015, uCap control
- Sivaraman et.al 2015, security control.

- Skarmeta et.al, 2014. Lightweight token for CoAP
- TLS/DTLS
- NETCONF
Results (Router performance)

OpenFlow-enabled home router **throughput.**
- Router: TP-link router runs open vSwitch on OpenWRT.
- Tool: Iperf.

<table>
<thead>
<tr>
<th>Connection type</th>
<th>w/o NAC</th>
<th>w/ NAC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TCP</td>
<td>UDP</td>
</tr>
<tr>
<td>Eth to Eth</td>
<td>94.1 MB</td>
<td>1.05 MB</td>
</tr>
<tr>
<td>WiFi to WiFi</td>
<td>21 MB</td>
<td>1.05 MB</td>
</tr>
</tbody>
</table>
Future work

➢ What if the MUD doesn’t exist?

➢ User device characterisation to apply closest matching MUD

➢ Asking the user to fill in a webform to define DAC is not usable for home users

➢ Use machine learning around context to infer policies from partial information provided by users (directly or via user behaviour analysis)

➢ Can we give users control over choosing between scenarios?
Stretch goal

➢ How can we crowd source information across multiple household?

➢ How do we maintain privacy?

➢ How do we guard against sybll attacks?
More information

➢ faucet.nz: opensource SDN controller for production networks.

Problem of the Heroic lone defender
IoT Honeypit

➢ Honeypot = attract attackers and monitor them, looks like a real system

➢ Tarpit = accepts requests and does nothing slowing attacker

➢ Honeypit = doesn’t look like a real system, provides rich and changing attack surface, *slow down hackers giving time to respond*

➢ Focusing on ssh initially
Each attacker gets custom set of vulnerabilities

Focusing on password vulnerabilities
- Common passwords, default passwords, credential stuffing...

Provide the attack /etc/passwd

Give them hashing challenges
Evaluation

➢ Does the attacker reverse the hash and post it somewhere (“honeyhash”)

➢ Does the attacker use the information to attack another device nearby

➢ Can we use this information also to feed into IDS? Vulnerability models?
Evaluation

➢ Does the attacker reverse the hash and post it somewhere ("honeyhash")

➢ Does the attacker use the information to attack another device nearby

➢ Can we use this information also to feed into IDS? Vulnerability models?
More information

➢ https://youtu.be/t5XBF4LApo0

➢ Talk on proof-of-concept for web applications at OWASP2018