TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

Automatic generation of IoT policies based on user preferences and IoT workfows

Presenter: **Ian Welch**

Authors: Mohammed Al-Shaboti, Aaron Chen, Ian Welch

2019 International Workshop on the Internet of Things Cybersecurity and Safety (ITCS 2019)

8-10 August 2019, Massey University, Auckland New Zealand

SCHOOL OF ENGINEERING AND COMPUTER SCIENCE FACULTY OF ENGINEERING

VICTORIA UNIVERSITY OF WELLINGTON

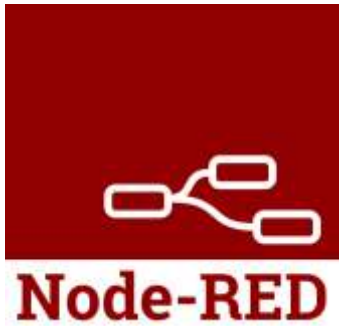
Context: Smart home in the future

STAMFORD, Conn., September 8, 2014

Gartner Says a Typical Family Home Could Contain More Than 500 Smart Devices by 2022

Programming the smart home

- How do write applications for the smart home and other IoT environments?
- Proposed visual programming environments based around ***workflows***.
- Drag and drop available devices in the environment to create a program.
- Examples: Node-RED, Visunio, wia etc.



Problem: dealing with evolution

- Many devices might be installed.
- Many different manufacturers.
- Devices come and go as consumers upgrade and replace installations.
- **Workflows are fragile because bound to specific IoT devices.**

https://au.pcmag.com/lighting/30286/the-best-rgb-keyboards

Problems: security concerns

- Even lightbulbs are general purpose computing devices.
- Might have different levels of trust
- Manufacturer Usage Descriptions (MAC) restrict interactions and permissions accordingly.
- **Problem – still coarse grained access control allowing misuse of permissions to launch attacks**.

IoT lightbulb worm takes over all smart lights until entire city is infected

Nov 10, 2016

NEWS by Danielle Correa

A research team has set up a chain reaction attack that would take over Philips Hue smart lightbulbs across entire cities

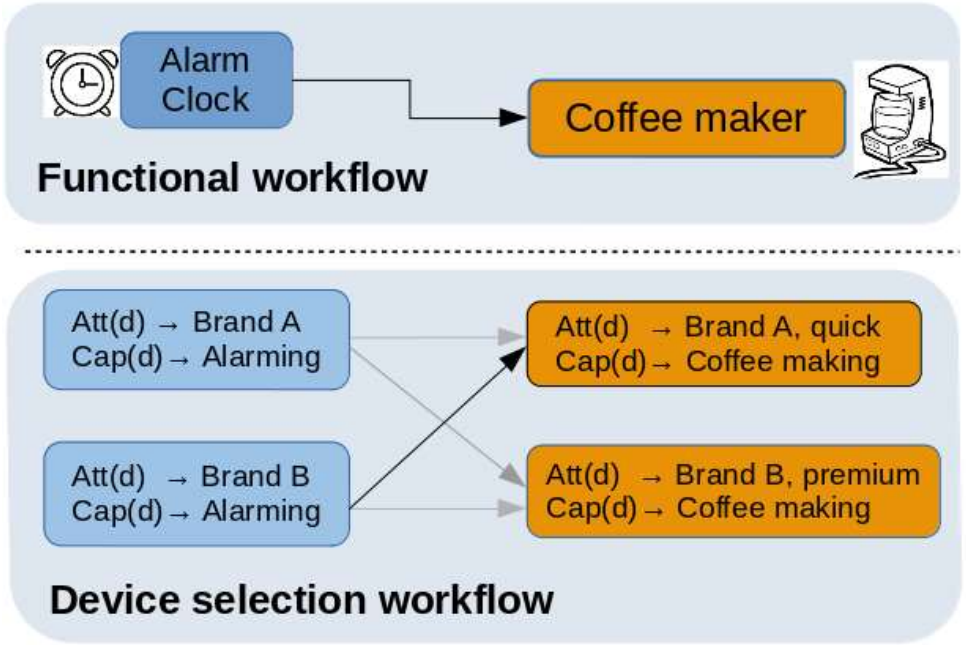
Researchers have developed a proof-of-concept attack on smart lightbulbs that allows them to wirelessly take control over the bulbs from up to 400m.

Goals

1. Abstract away from specific instances of device when specifying workflows.
2. Take customer preference regarding trust etc. into account when choosing particular devices.
3. Reduce risk of misuse of privileges by only granting permissions as required to carry out user tasks.
4. Automatically generate fine-grained access control policies to enforce least privilege on IoT devices.

Problem formulation

- **A device** is descried by a set of *attributes* and a set of *capabilities* each associated with *network requirements*.
- **A network** is a collection of devices.
- **Activity workflow** is a functional representation of the desired activity.
- **A user preference model** to quantify user preference of using any set of devices for a particular set of activity functions.
- **Treat as an optimization problem.**

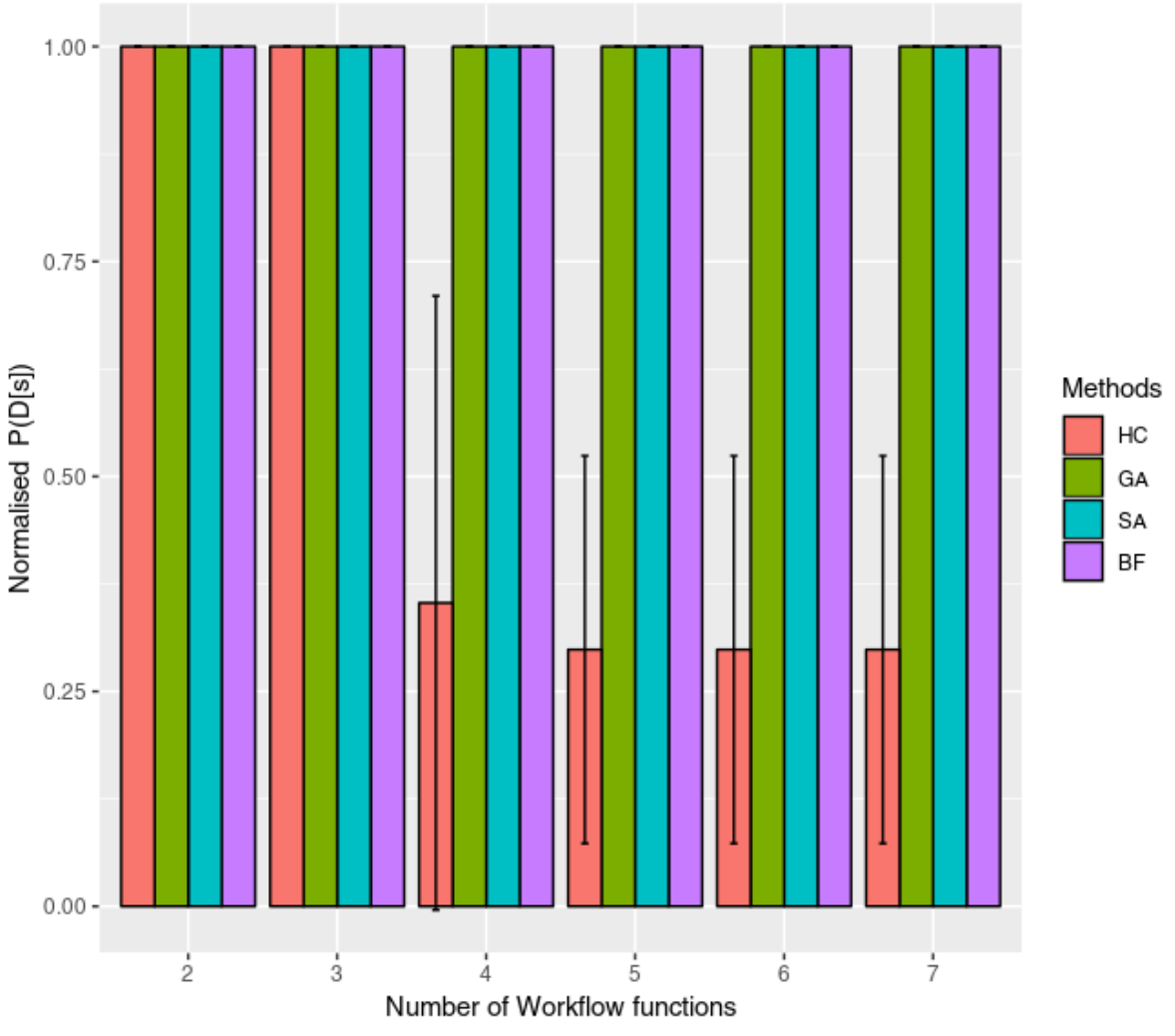


Methodology

- Find the set of devices that fulfill workflow functions and maximize user preference.
- Heuristic search algorithms used to find the set of preferable devices.
  - Hill-Climbing (HC) - Local optimization
  - Simulated Annealing (SA) -Local optimisation
  - Genetic algorithm (GA) – (global optimisation)
- A simple policy generation algorithm used device capability network requirements to generate access policy for the selected devices.

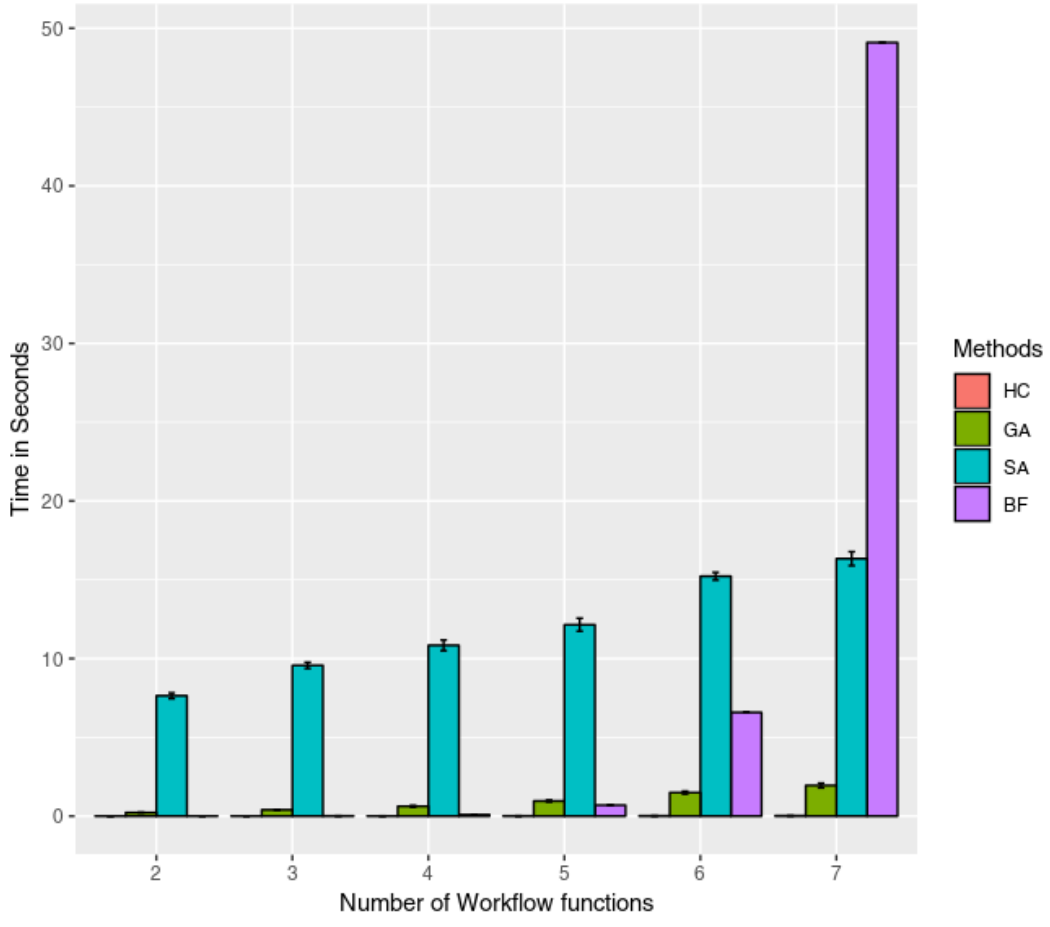
Empirical evaluation (satisfaction)

- Empirical evaluation.
- Multiple workflows randomly chosen (2-7 functions).
- Assume seven alternative devices for each function.
- Randomised preference model with known optimum choice.
- All but Hill Climbing found optimum.



Empirical evaluation (time)

- How long does it take?
- Hill climbing is fastest.
- Hill climbing doesn't find the optimum though.
- Genetic algorithm comes second while finding the optimum.
- Our winner for algorithm to use in this case.



Network access control policy

- Once the devices have been selected.
- Generate ACLs:
  - <srcIP, dstIP, Protocol, dstPort>
- Earlier work we designed SDN-based enforcement mechanism based on Faucet controller.

Summary

- Automatic approach to generation of fine-grained access control policy based upon:
  - User requirements (workflow)
  - User preferences (trust or habit).
- Novel use of optimization techniques to achieve this.
- Future work: implementation, other user context information.

Thank you!