Standardizing IoT Network Security Policy Enforcement

David Barrera*, Ian Molloy, Heqing Huang
IBM Research

* Polytechnique Montreal

20 Billion IoT devices online by 2020

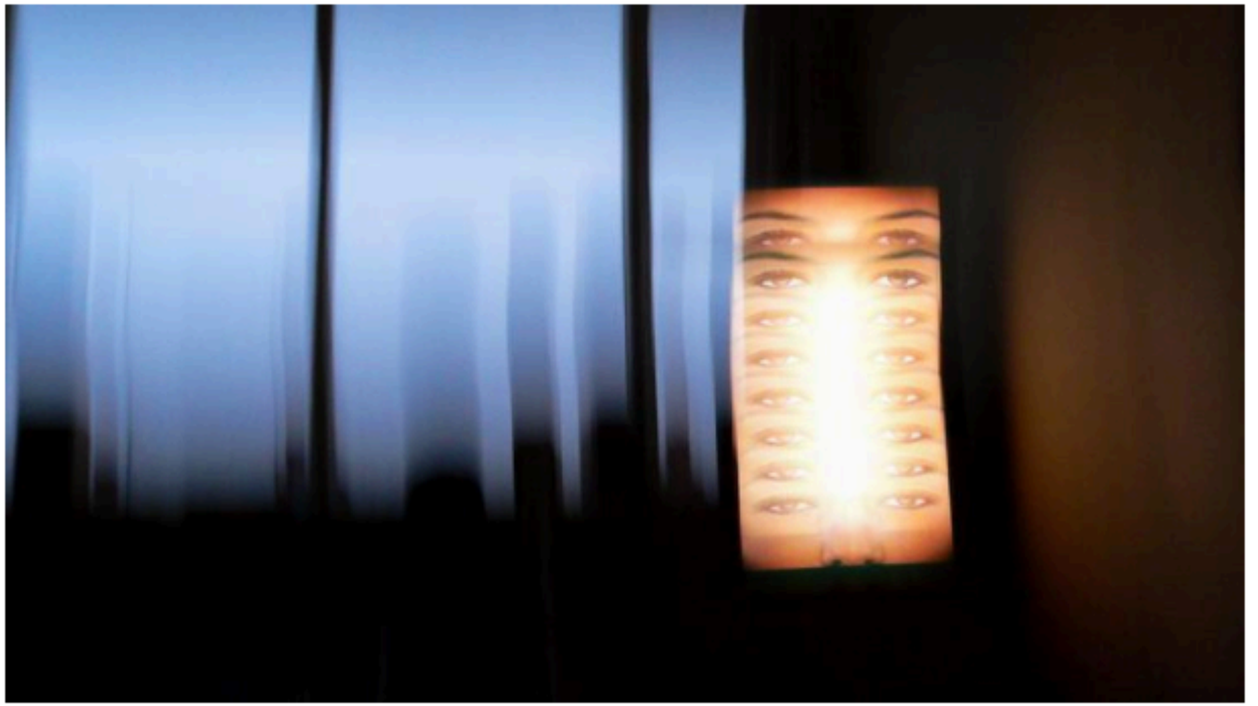
13.5 billion (65%) devices in the ***consumer*** space

http://www.gartner.com/newsroom/id/3165317

Security Challenges of consumer IoT devices

- Transparency – What are the devices doing?
- No screens/displays, communicate status via LEDs
- No keyboard, cannot debug easily
- Currently require full trust in vendor

THE INTERNET OF HACKABLE THINGS



The CIA Spied on People Through Their Smart TVs, Leaked Documents Reveal

Hackers from the CIA found a way to keep Samsung Smart TVs on “Fake-Off mode.”

Security Challenges of consumer IoT devices

- Security
- Devices can run arbitrary code
- Often use weak credentials
- Do not/cannot run anti-malware on-device

- Weak and default credentials
- SSH keys and backdoors

BBC Sign in News Sport Weather Shop Earth Travel

NEWS

Home Video World US & Canada UK Business Tech Science Magazine

Technology

'Smart' home devices used as weapons in website attack

22 October 2016 | Technology Share

Net-connected cameras are helping attackers in large-scale attacks

Widespread cyberattack takes down sites worldwide

by Sara Ashley O'Brien @saraashleyo

October 21, 2016: 8:11 PM ET

Massive cyber attack takes down major websites

A number of popular websites like Twitter and Netflix went down for some users on Friday in a massive cyberattack with international reach.

Affected sites included Twitter (TWTR), Etsy (ETSY), Github, Vox, Spotify, Airbnb, Netflix (NFLX) and Reddit.

Social Surge - What's Trending


Jimmy Kimmel to Trump after school shooting: 'You've literally done nothing'


The case for canceling all student debt

Cisco: We're moving our $67 billion cash pile to the U.S.

Features!

- Set bulb state: `on` or `off`
- Get bulb state
- Allow three year old to yell at Alexa to turn on the lights

- ~~DoS Dyn~~
- ~~Exfiltrate data~~
- ~~Send spam~~
- ~~Meddle in US elections~~

Overview

- IoT devices often serve a single purpose (lightbulb on/of, upload video footage, collect temperature data)
- The network profile of IoT devices is simpler than desktops/servers
- Idea: restrict network behavior of IoT devices to **only what is required for essential functionality**
- Avoid requiring installation an agent on the IoT device
- **Deployability, Extensibility, Simplicity**

Comparison to related technologies

Consumer firewalls

- Basic network filtering and blocking of unsolicited inbound traffic
- Allow outbound traffic by default
- No support for application-layer filtering

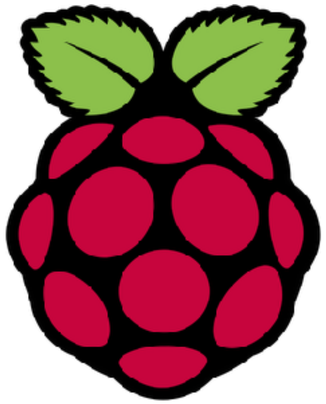
Enterprise solutions:

- Network Access Control (NAC) – most effective when used with an agent on the device
- Next-generation firewalls and Unified Threat Management
  - Incorporate DPI, IDS/IPS, anti-malware, VPN, etc.
  - Heavyweight solutions
  - Expensive

Comparison to related technologies

IDIoT brings enterprise-like security features to the consumer space, focusing on simplicity of policy management.

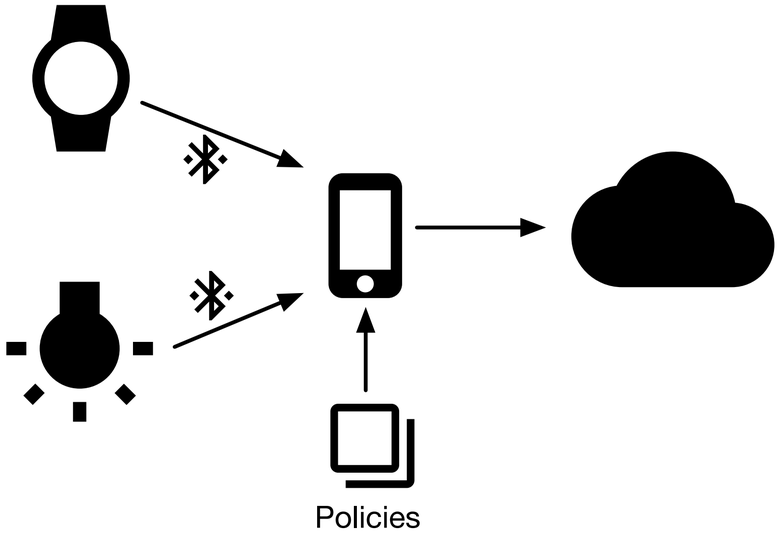
- IoT devices don’t significantly change their behavior over time – allows for simple policies and lightweight filtering
  - Our development board is a Raspberry Pi
- IoT devices don’t support installation of agents – focus on passive network monitoring
- Automate as much as possible, as home users are not expert administrators
- Support exporting policies to different targets

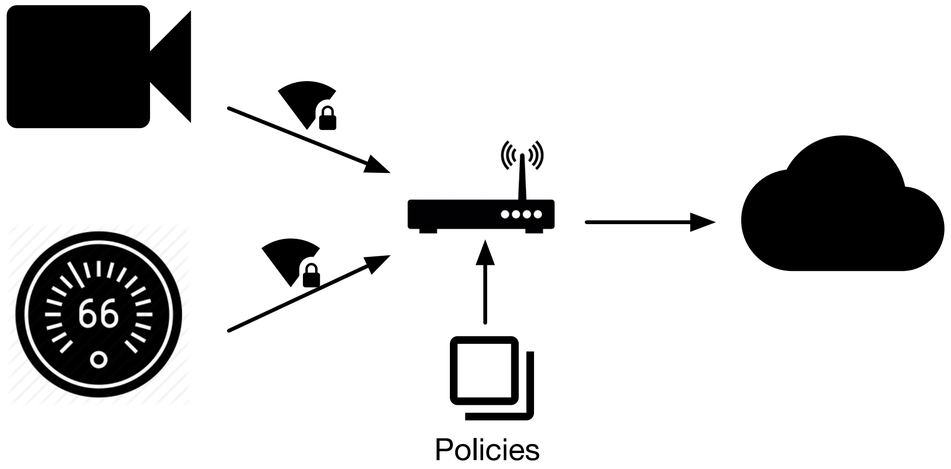


Overview

- Create a security policy enforcement mechanism that restricts the network communication of IoT devices to only what is essential
  - E.g., surveillance cameras can upload footage to a cloud storage provider, but can’t flood DNS resolvers with bogus queries
- Policy rules supporting multiple layers
  - Network layer (IP addresses, throughput, packet length, etc.)
  - Application layer (DNS, NTP, HTTP, etc.)
- Flexible enforcement
  - At the edge - better visibility control
  - In the cloud – easier setup and management

- Handle Zigbee, Bluetooth, etc. on mobiles or hubs





Quick Analysis

- Monitored network traffic for 12 minutes from cold start
- “Representative” devices from our houses and UNSW Data
- IoT devices connect to small number of services and domains
- General purpose devices more complicated network behavior
- Apps and skills complicating separation

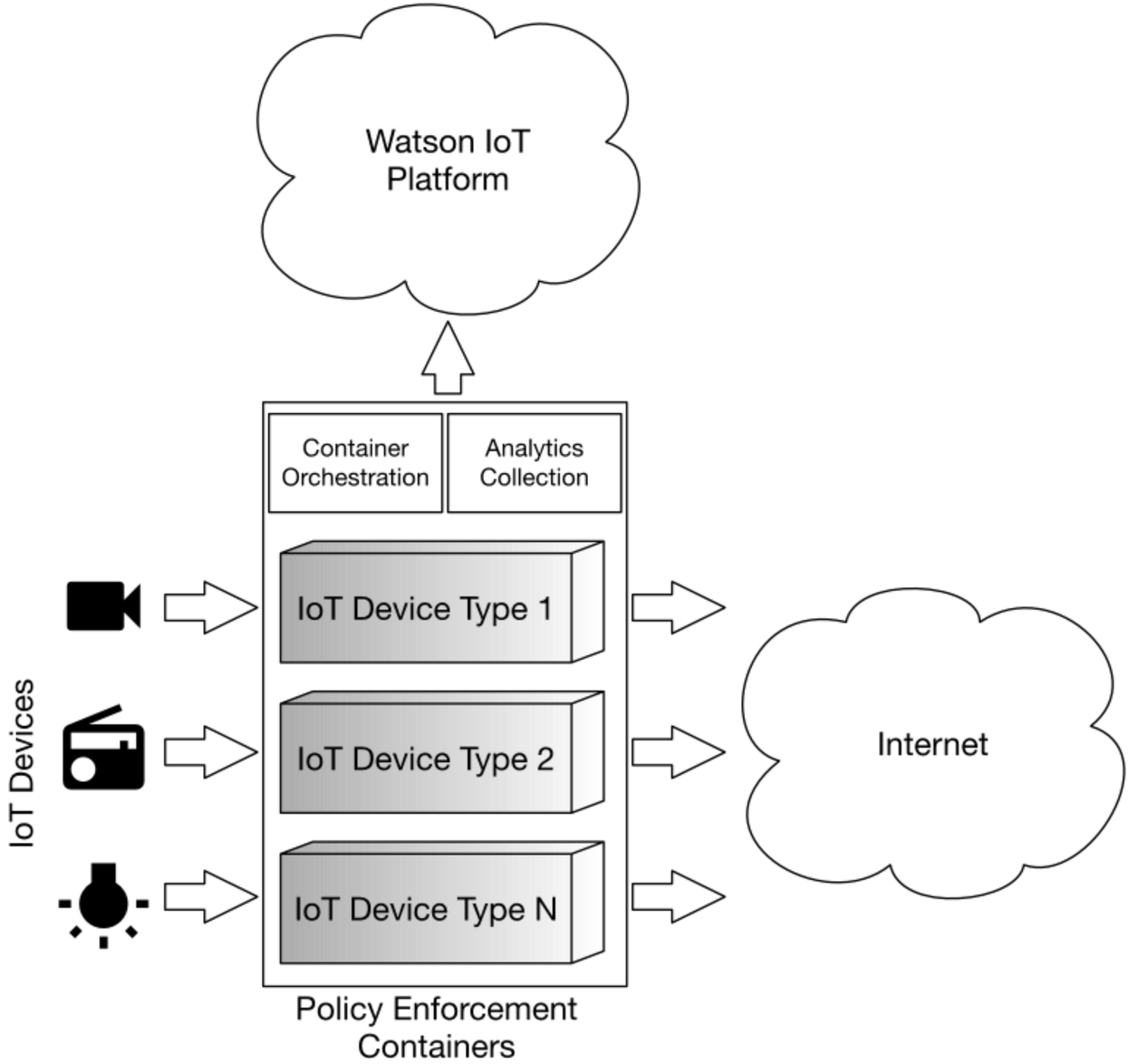
Device	Distinct Endpoints	Distinct Domains	HC IPs
AT&T Microcell	2	0	2
Fitbit Aria Digital Scale	2	1	0
Withings Smart scale†	2	1	0
Withings Baby Monitor†	2	1	0
PIX-STAR Photo-frame†	2	1	0
Belkin Wemo switch†	2	1	0
Blipcare BP meter†	2	1	0
Samsung Bluray Player	4	1	0
Netatmo Weather Station	5	1	0
LIFX Gen 1 bulb✳	5	1	0
LIFX Gen 2 bulb✳	5	2	0
Triby Speaker†	6	2	0
NEST Smoke Alarm†	6	4	0
TP-Link Smart plug†	7	2	0
Netatmo Welcome†	7	2	6
Amazon Fire TV	8	4	0
Amazon Kindle	9	8	1
TP-Link Cloud camera†	15	2	3
Amazon Echo✳	20	13	0
AppleTV 4th Gen	37	23	2
Samsung Galaxy Tab†✳	48	21	0
Android Phone†	57	48	0
Microsoft XBox One	74	57	0
Laptop†	140	101	0

Policy Enforcement Details

- Schedule (fixed: Mon-Fri, 10:00-10:30, periodic: once per week)
- Throughput/quota: packet rate (10Kb/s), Bandwidth (10 MB/month), session bytes (500 Kb out)
- Endpoints: Src/Dst (IP or hostname)
- Protocols (TCP/UDP) and port numbers
- Layer 7:
  - HTTP requests (URI http://api.lifx.co/status, parameters: POST, PUT, including wildcards for auth tokens and nonces)
  - NTP (version, mode, stratum, etc)
  - DNS (query/response type, hostnames)
  - TLS (ciphers, public key, certificate metadata)

Architecture

- Containers act as the default gateways for IoT devices
- One container per type of device. Each container can enforce policies for multiple devices of the same type (e.g., Philips light bulbs or Linksys surveillance cameras)
- Containers allow traffic specified in policies to reach the Internet
  - Traffic that violates the policy is dropped and logged



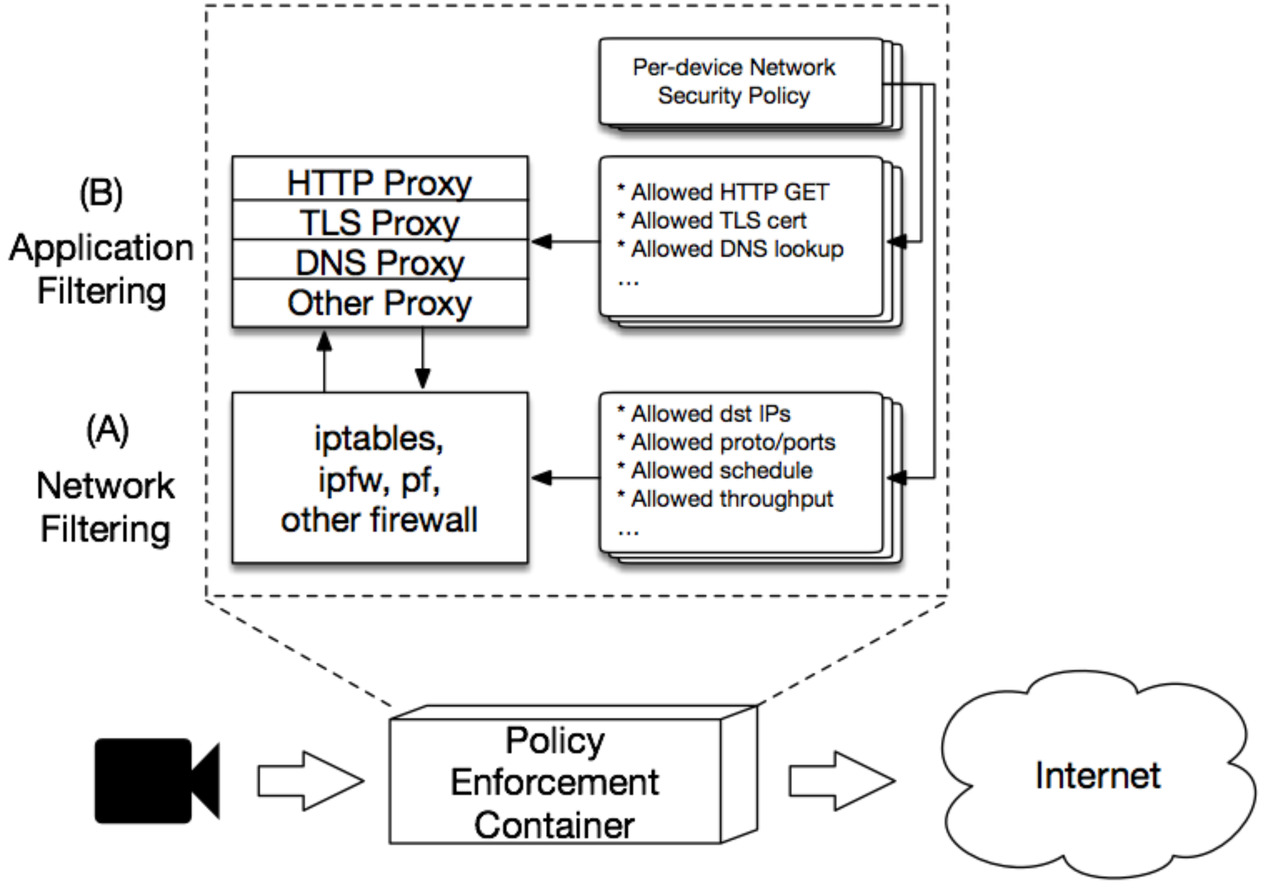
Policy Enforcement Containers and Implementation

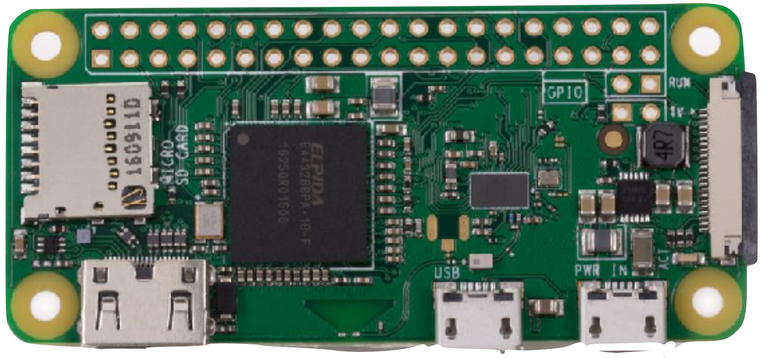
- Docker Alpine Linux base (5 MB base image)
- Pre-configured proxies and firewall rules according to policy

- `hostapd (ap_isolate=1)`
- `iptables`
- `dsnmasq (no-resolv)`

- Separated network into `172.16.1.0/24` and `192.168.1.0/24` networks

- `server=/netcom.netatmo.net/8.8.8.8`
- `address=/#/127.0.0.1`





Example

```
#iptables -t nat -A PREROUTING -i
wlan0 \\
  -s 172.16.1.2 -d 62.210.92.0/24 -p
tcp \\
  --dport 25050 -m limit --limit 6/
hour -j ACCEPT

#iptables -t nat -A PREROUTING -i
wlan0 \\
  -s 172.16.1.2 -d 192.168.1.1 -p udp
\\
  --dport 53 -j ACCEPT
```

Listing 1: "Example policy for the Netatmo weather station"

```
{"Netatmo Weather Station": {
  "MACAddr": "70:ee:50:13:ab:cd",
  "IPAddr": "172.16.1.2",
  "AllowedDNSQueries": [
    {"type": "A", "query": "netcom.netatmo.net",
        "resolver": "192.168.1.1"}
  ],
  "AllowedDNSReplies": [
    {"type": "A", "query": "netcom.netatmo.net",
        "answers": "62.210.92.0/24"}
  ],
  "AllowedConnections": [
    {"family": "IPv4", "dest": "netcom.netatmo.
        net", "proto": "TCP", "dstport": "25050",
         "freq": "6/hr"}
  ]
 }
}
```

Testing / Comments

- Need to accommodate user-initiated activity (netatmo pulls every 10m)
- Some additional latency for some devices (going to cloud)
- Device identification has obvious caveats (e.g., MAC spoofing)
- Skills and Apps require more complicated profiles — enforced on device?
- Multihoming (e.g., cellular) moves enforcement point

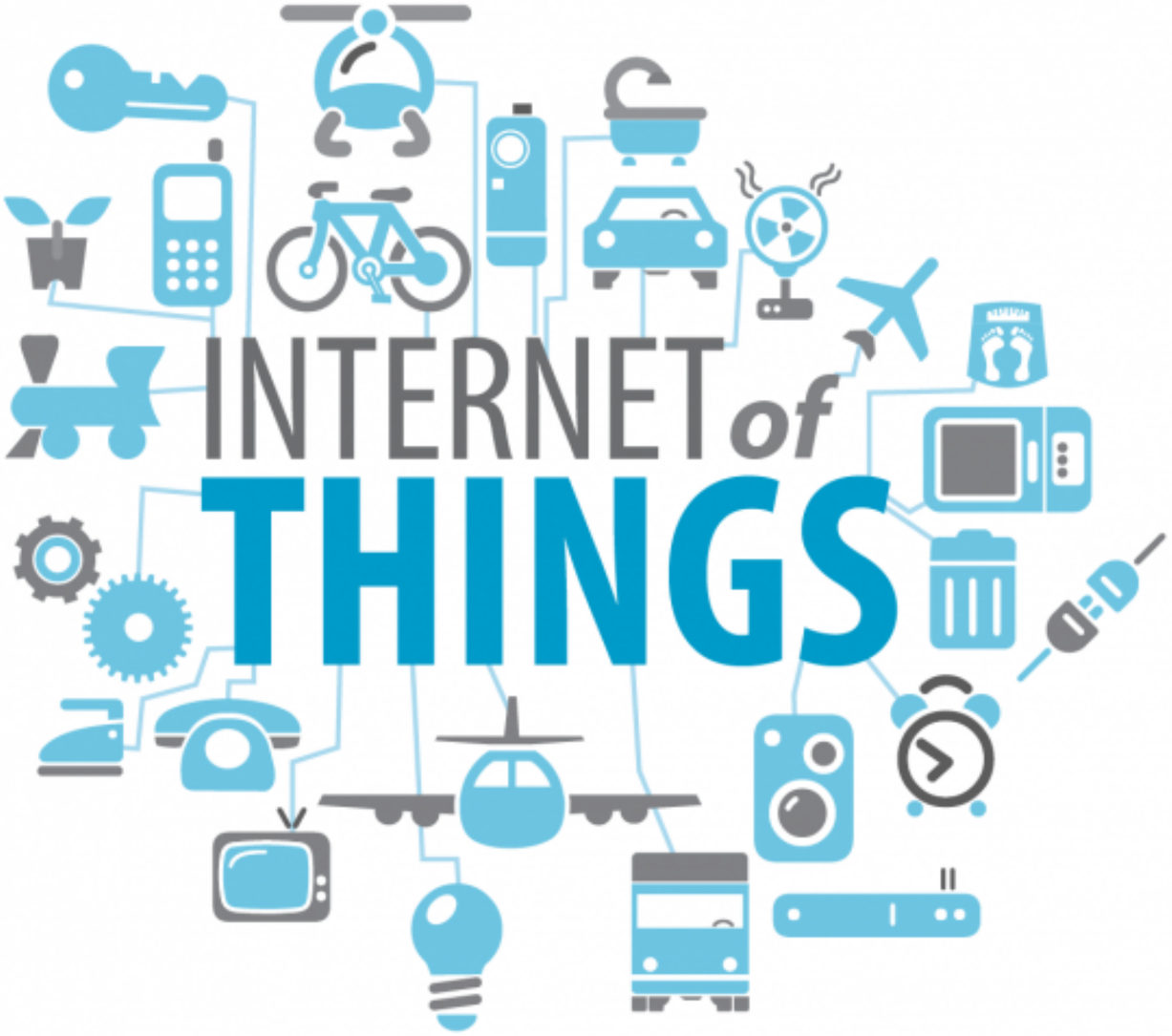
Obtaining Network Access Policies

- **Vendor provided**: delivered with device purchase (scan QR code, install from website)
- **Dynamically learned**: observe IoT device traffic for some time, generate a policy
- **Crowdsourced**: leverage blockchain to collect anonymized network profiles of devices and build policies

- Blockchain

Conclusions

- Most IoT devices serve a single purpose—keep it that way
- Profiled IoT devices using through network analysis to create security policies
- Enforce policies using per-device network stacks in Docker containers on a Raspberry Pi
- Provide human and machine understandable policies for what devices should do



Questions?

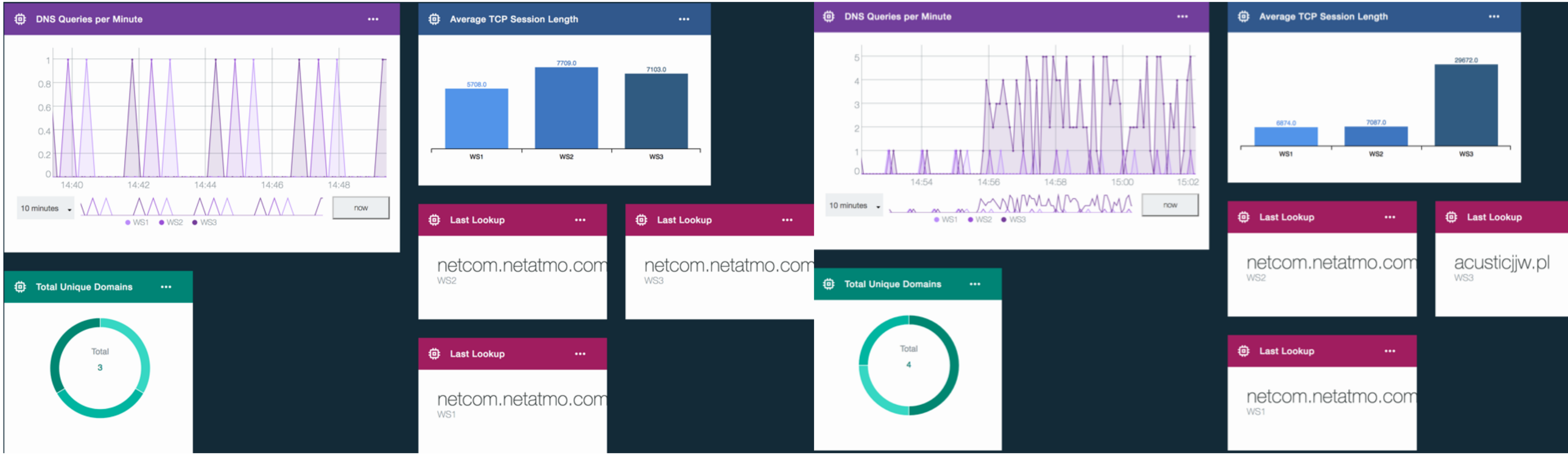
MUD

- Trust manufacturer

LB100


```
{
    "Device": "50:c7:bf:5e:
47:41",
    "AllowedLookups": [
        "A
devs.tplinkcloud.com",
        "A pool.ntp.org",
        "A time-a.nist.gov"
    ],
    "NeedsDHCP": true,
    "AllowedConnections": [
        {
            "IP": "",
            "Domain": "",
            "Protocol": "",
            "Port": 0,
            "Lookup": false,
            "Bytes": 0,
            "InPackets": 0,
            "OutPackets": 0
        },
        {
            "IP":
"52.204.41.30",
            "Domain":
"devs.tplinkcloud.com",
            "Protocol": "TLS",
            "Port": 50443,
            "Lookup": true,
            "Bytes": 7710,
            "InPackets": 12,
            "OutPackets": 20,
            "TLSHandshake": {
                "ClientFP":
"0303/2F353C3D9C9DC004C005C009
C00AC00EC00FC013C014C023C024C0
25C026C027C028C029C02AC02BC02C
C02DC02EC02FC030C031C032C09CC0
9DC0A0C0A1CC13CC14/00/000A0019
0018001700150013010000120603050304030203060105010401020101
",
"ClientVersion": "TLSv1.2"
            }
        },
        {
            "IP":
"45.76.92.117",
            "Domain":
"pool.ntp.org",
            "Protocol": "UDP",
            "Port": 123,
            "Lookup": true,
            "Bytes": 90,
            "InPackets": 1,
            "OutPackets": 1
        }
    ]
}
```

```
{
    "Device": "34:d2:70:6d:c5:2e",
    "AllowedLookups": [
        "A spectrum.s3.amazonaws.com",
        "A 2.android.pool.ntp.org",
        "A kindle-time.amazon.com",
        "AAAA pindorama.amazon.com",
        "AAAA www.example.com",
        "A ntp-g7g.amazon.com",
        "AAAA www.example.net",
        "AAAA www.example.org",
        "A dcape-na.amazon.com",
        "A device-messaging-
na.amazon.com",
        "A todo-ta-g7g.amazon.com",
        "A arcus-uswest.amazon.com",
        "A softwareupdates.amazon.com",
        "A dp-rsm-prod.amazon.com",
        "A dp-gw-na.amazon.com",
        "A api.amazon.com",
        "A device-metrics-us.amazon.com",
        "A det-ta-g7g.amazon.com"
    ],
    "NeedsDHCP": true,
    "AllowedConnections": [
        {
            "IP": "",
            "Domain": "",
            "Protocol": "",
            "Port": 0,
            "Lookup": false,
            "Bytes": 0,
            "InPackets": 0,
            "OutPackets": 0
        },
        {
            "IP": "52.216.66.32",
            "Domain":
"spectrum.s3.amazonaws.com",
            "Protocol": "TCP",
            "Port": 80,
            "Lookup": true,
            "Bytes": 3032,
            "InPackets": 7,
            "OutPackets": 12
        },
        {
            "IP": "176.32.98.203",
            "Domain": "kindle-
time.amazon.com",
            "Protocol": "TCP",
            "Port": 80,
            "Lookup": true,
            "Bytes": 721,
            "InPackets": 3,
            "OutPackets": 4
        },
        {
            "IP": "54.239.29.231",
            "Domain":
"pindorama.amazon.com",
            "Protocol": "TLS",
            "Port": 443,
            "Lookup": true,
            "Bytes": 55551,
            "InPackets": 216,
            "OutPackets": 359,
            "TLSHandshake": {
                "ClientFP":
"0303/345689A1112131415162F32333538393C3D4
0676A6B9C9D9E9FA2A3FFC002C003C004C005C007C
008C009C00AC00CC00DC00EC00FC011C012C013C01
4C023C024C025C026C027C028C029C02AC02BC02CC
02DC02EC02FC030C031C032/00/0032000E000D001
9000B000C00180009000A0016001700080006000700
0140015000400050012001300010002000300F001
0001103000102001E060106020603050105020503
0401040204030301030203030201020202203",
                "ClientVersion": "TLSv1.2"
            }
        },
        {
            "IP": "93.184.216.34",
            "Domain": "www.example.com",
            "Protocol": "TCP",
            "Port": 80,
            "Lookup": true,
            "Bytes": 264,
            "InPackets": 1,
            "OutPackets": 4
        },
        {
            "IP": "72.21.195.82",
            "Domain": "dcape-
na.amazon.com",
            "Protocol": "TLS",
            "Port": 443,
            "Lookup": true,
            "Bytes": 7653,
            "InPackets": 10,
            "OutPackets": 12,
            "TLSHandshake": {
                "ClientFP":
"0303/52F32333538399C9D9E9FFFC007C009C00AC
011C013C014C02BC02CC02FC030/00/0032000E000
D0019000B000C00180009000A00160017000800060
0070014001500040005001200130001000200030000
F001000110300010200 1E060106020603050105020
503040104020403030103020303020102020203",
                "ClientVersion": "TLSv1.2"
            }
        },
        {
            "IP": "52.94.225.171",
            "Domain": "device-messaging-
na.amazon.com",
            "Protocol": "TLS",
            "Port": 443,
            "Lookup": true,
            "Bytes": 7638,
            "InPackets": 10,
            "OutPackets": 12,
            "TLSHandshake": {
                "ClientFP":
"0303/52F32333538399C9D9E9FFFC007C009C00AC
011C013C014C02BC02CC02FC030/00/0032000E000
D0019000B000C00180009000A00160017000800060
0070014001500040005001200130001000200030000
F001000110300010200 1E060106020603050105020
503040104020403030103020303020102020203",
                "ClientVersion": "TLSv1.2"
            }
        },
        {
            "IP": "52.94.225.226",
            "Domain": "todo-ta-
g7g.amazon.com",
            "Protocol": "TLS",
            "Port": 443,
            "Lookup": true,
            "Bytes": 7817,
            "InPackets": 9,
            "OutPackets": 10,
            "TLSHandshake": {
                "ClientFP":
"0303/52F32333538399C9D9E9FFFC007C009C00AC
011C013C014C02BC02CC02FC030/00/0032000E000
D0019000B000C00180009000A00160017000800060
0070014001500040005001200130001000200030000
F001000110300010200 1E060106020603050105020
503040104020403030103020303020102020203",
                "ClientVersion": "TLSv1.2"
            }
        },
        {
            "IP": "52.94.208.165",
            "Domain": "arcus-
uswest.amazon.com",
            "Protocol": "TLS",
            "Port": 443,
```


DNS Queries per Minute
1
0.8
0.6
0.4
0.2
0
14:40
14:42
14:44
14:46
14:48
10 minutes
now
WS1
WS2
WS3
Average TCP Session Length
5708.0
7709.0
7103.0
WS1
WS2
WS3
Last Lookup
netcom.netatmo.com
WS2
Last Lookup
netcom.netatmo.com
WS3
Last Lookup
netcom.netatmo.com
WS1
Total Unique Domains
Total
3
DNS Queries per Minute
5
4
3
2
1
0
14:54
14:56
14:58
15:00
15:02
10 minutes
now
WS1
WS2
WS3
Total Unique Domains
Total
4
Average TCP Session Length
6874.0
7087.0
29672.0
WS1
WS2
WS3
Last Lookup
netcom.netatmo.com
WS2
Last Lookup
acusticjjw.pl
WS3
Last Lookup
netcom.netatmo.com
WS1

Policy Enforcement at Multiple Layers

- **Network layer (firewall rules)**
  - Allowed endpoints
  - Allowed ports, protocols
  - Allowed bandwidth
- **Application layer (proxies)**
  - Allowed DNS lookups, answers
  - Allowed TLS certificates
  - Allowed GET/POST requests