HOMESNITCH

Behavior Transparency for Smart Home IoT Devices

TJ O'Connor, North Carolina State University
Reham Mohamed, Technische Universität Darmstadt
Markus Miettinen, Technische Universität Darmstadt
William Enck, North Carolina State University
Bradley Reaves, North Carolina State University
Ahmad-Reza Sadeghi, Technische Universität Darmstadt

Motivating Example

response.response.shouldEndSession = false
response.response.reprompt = NULL;

IoT lacks transparency of behaviors and offers limited access control

Problem

How can we provide transparency and control of behaviors of otherwise resource constrained smart home IoT devices?

Challenges

- Behavior Classification
- Network Mediation

Related Work

IoT Behavior Detection/Classification

- Bezawada et. al. 2018. Behavioral Fingerprinting of IoT Devices. In Workshop on Attacks and Solutions in Hardware Security (ASHES). ACM, Toronto, Canada, 41–50. **(IoTSense)**
- Acar et al. Peek-a-Boo: I see your smart home activities, even encrypted! **(Arxiv),** 2018.
- Acar et al. "Web-based Attacks to Discover and Control Local IoT Devices." Proceedings of the 2018 Workshop on IoT Security and Privacy. ACM, 2018 **(IoT-Inspector)**

Classification

- Reed and Kranch, “Identifying **https-protected Netflix videos** in real-time,” in Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, ser. CODASPY ’17.

Training Data Sets

- M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. R. Sadeghi, and S. Tarkoma, “Iot sentinel: Automated device-type identification for security enforcement in iot,” ICDCS 2017.
- Omar Alrawi, Chaz Lever, Manos Antonakakis, Fabian Monrose; SoK: Security Evaluation of Home-Based IoT Deployments, IEEE S&P, May 2019.

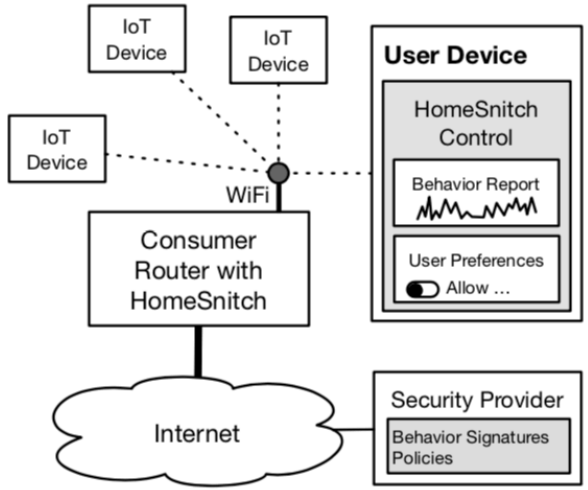
Challenges

Behavior Classification

- Encrypted communications
- Proprietary protocols
- Using only transport headers

Network Mediation

- Flat IP address space
- Cannot segment the network
- Perpetually connected devices



Threat Model

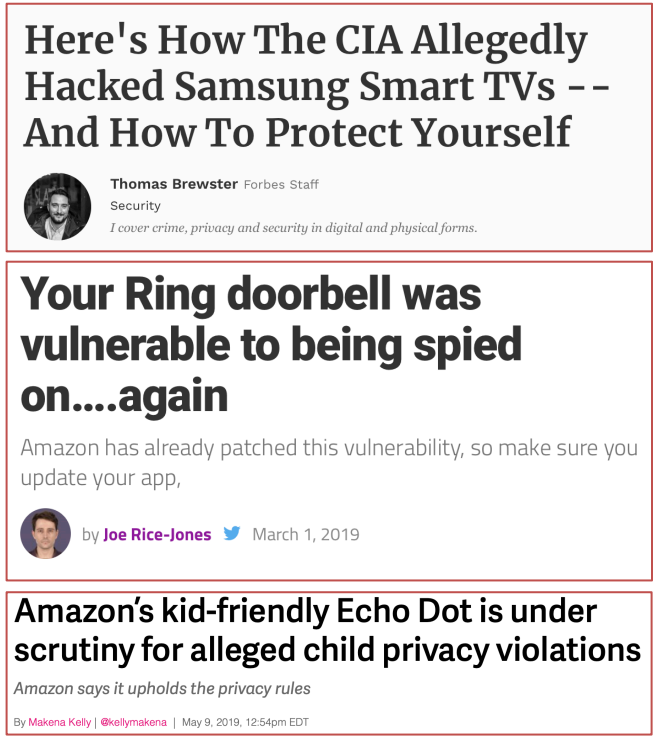
Assumptions: Devices with default credentials, lack security protocols, enable over-privilege.

Attacker Goal: Execute a behavior transparently to end user.

TCB: the SDN security application, the network data plane devices.

We do not address the case of a compromised device that can perform mimicry attacks.

Deployment Task: Extend beyond TCP/IP to ZigBee, Bluetooth, NFC protocols.



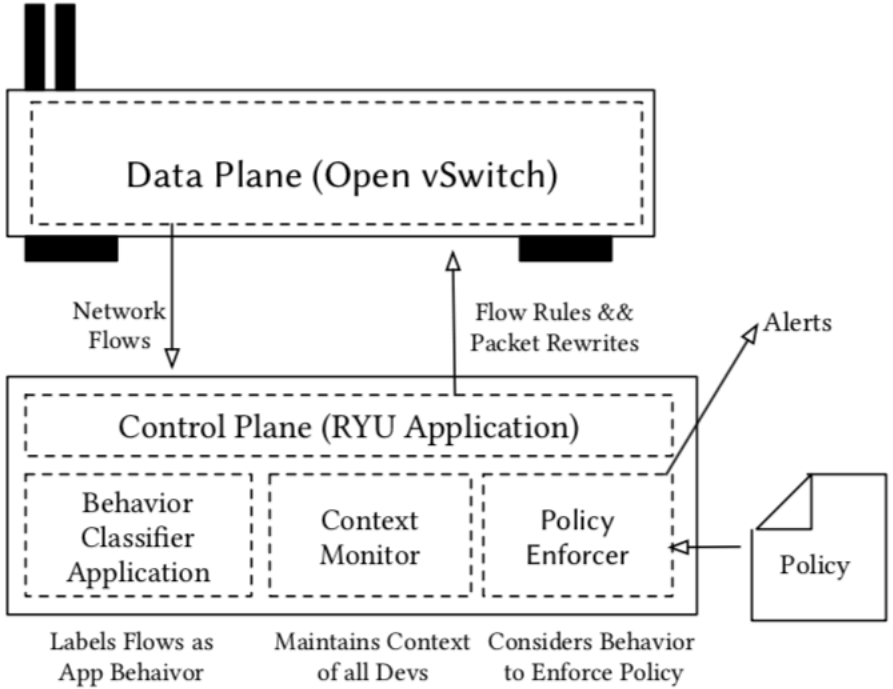
Design

Behavior Classification

- Classifies flows into known behaviors.
- Identifies when new behaviors occur.

Policy Enforcer

- Translates policy into network rules.
- Uses OpenFlow modifications for traffic.



Behaviors

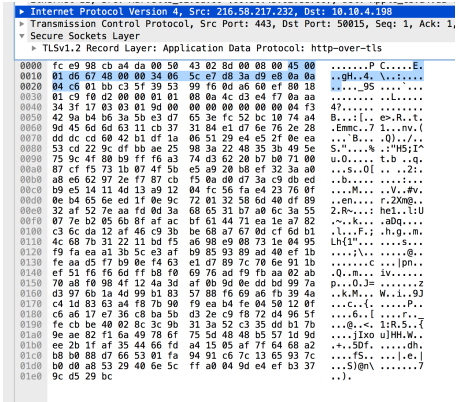
Our initial attempts tried to classify **just based on activity alone**; however we found devices implement activities differently.

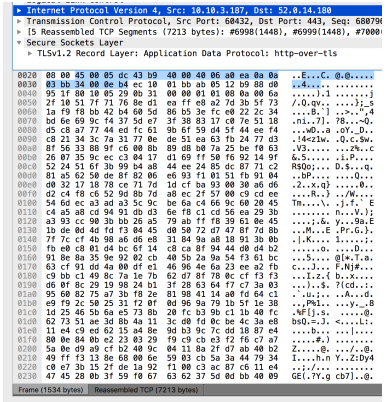
Our behaviors are a triple of **<Vendor>,<Device>,<Activity>**; examples include

- <Ring>,<Doorbell>,<Heartbeat>
- <Ring>,<Doorbell>,<Video>
- <Canary>,<Security Camera>,<Video>

We use our behavior triples to find the next nearest behavior to the vendor, type of device and activity.

Traffic Classification





Behavior: ***Nest, Protect, Alarm Test***
Protocol: HTTP-over-TLS
DPORT: 443
DST: 216.58.217.213 (Google Cloud)

Behavior: ***Canary, Camera, Video***
Protocol: HTTP-over-TLS
DPORT: 443
DST: 52.0.14.180 (Amazon AWS)

Behavior Classification Insight

Adudump constructs a structural model that ignores transport layer effects.

Provides an abstract representation model for the exchange between a client and a server.

Bridges gap between transport layer packet headers and client/server application dialogues.

```
SYN,1498821090.224348,10.10.4.197.60829,>1,130.211.93.93.80
RTT,1498821090.351369,10.10.4.197.60829,<1,130.211.93.93.80,0.126964
SEQ,1498821090.352212,10.10.4.197.60829,>1,130.211.93.93.80
ADU,1498821090.617700,10.10.4.197.60829,>1,130.211.93.93.80,726,SEQ,0.128930
ADU,1498821090.618610,10.10.4.197.60829,<1,130.211.93.93.80,457,SEQ
END,1498821090.633606,10.10.4.197.60829,x0,130.211.93.93.80
```

Behavior Classification: Feature Selection

Features derived from *ADU.*

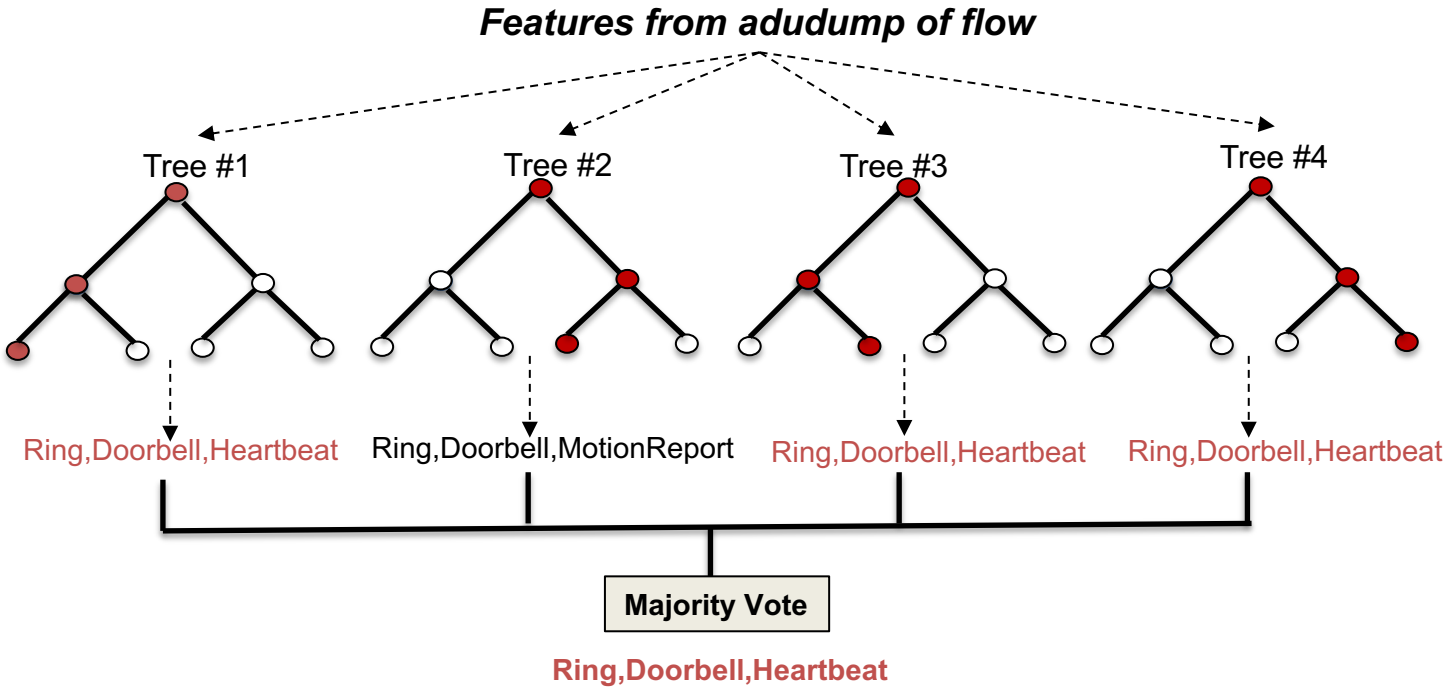
Features describe ADU:

- Throughput
- Burstiness
- Synchronicity
- Duration

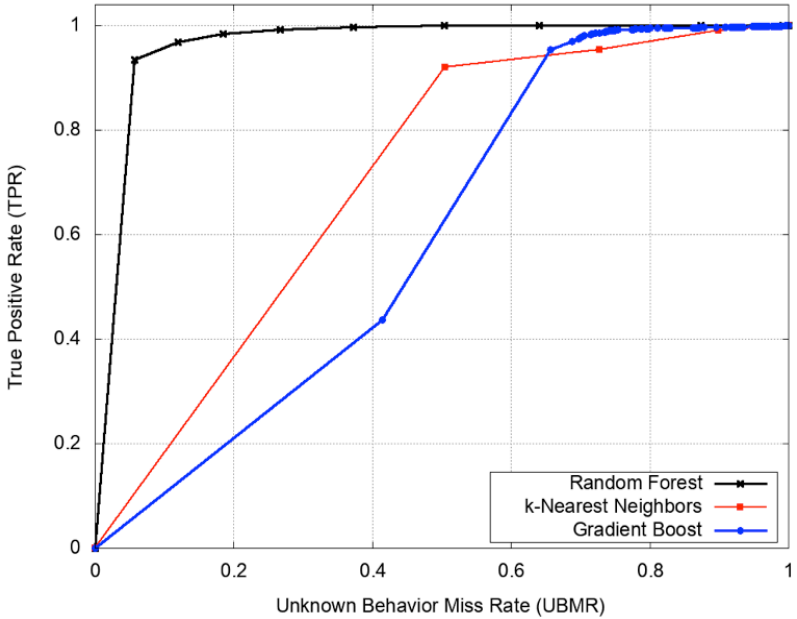
Feature	Category	Importance
Avg. bytes from client per seq.	**Throughput**	**0.213104**
Avg. bytes from server per seq.	**Throughput**	**0.072519**
Aggregate server bytes sent for ADU	Throughput	0.105775
Aggregate client bytes sent fo ADU	Throughput	0.117552
Min bytes from client for single seq.	**Burstiness**	**0.038917**
Min bytes from server for single seq.	**Burstiness**	**0.038344**
Max bytes from server for single seq.	Burstiness	0.079063
Max bytes from client for single seq.	Burstiness	0.135909
Stdev of bytes for server seq.	Burstiness	0.054491
Stdev of bytes for client seq.	Burstiness	0.050798
Server sequences per ADU	Synchronicity	0.013566
Client sequences per ADU	Synchronicity	0.016211
Total time of connection	Duration	0.063750

```
SYN,1498821090.224348,10.10.4.197.60829,>1,130.211.93.93.80
RTT,1498821090.351369,10.10.4.197.60829,<1,130.211.93.93.80,0.126964
SEQ,1498821090.352212,10.10.4.197.60829,>1,130.211.93.93.80
ADU,1498821090.617700,10.10.4.197.60829,>1,130.211.93.93.80,726,SEQ,0.128930
ADU,1498821090.618610,10.10.4.197.60829,<1,130.211.93.93.80,457,SEQ
END,1498821090.633606,10.10.4.197.60829,x0,130.211.93.93.80
```

Random Forest Classifier



Unknown Behaviors Are Important



UBMR: *percentage of data-points in a given dataset belonging to previously unseen classes that fail to be identified as a new class.*

	Acc.	Recall	F1
KNN	99.32+/-.12	87.97+/-2,16	86.35+/-2.65
Gradient Boost	64.70+/-42.10	53.55+/-33.51	51.72+/-32.72
Random Forrest	99.69+/-.06	94.66+/-1.21	93.93+/-1.40

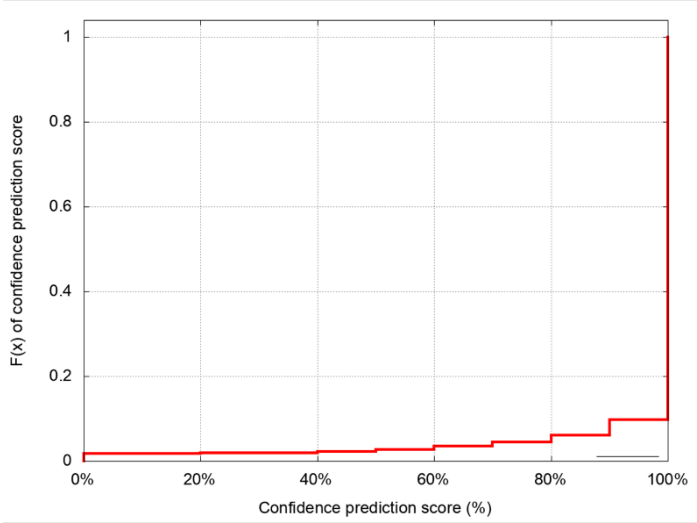
Random Forest: *perform well for numerical features without scaling and perform implicit feature selection.*

Minimal Training

We used a *test training approach* that incrementally added back individual ADUs of each behavior class.

Results indicated behaviors can be trained with relatively few samples.

58.82 samples for 90% confidence.
62.28 samples for 95% confidence.



How do we use HomeSnitch

Cluster initial dataset into distinct behavior classes

(<Ring,Doorbell,Behavior-1>, <Ring,Doorbell,Behavior-2>, <Ring,Doorbell,Behavior-3>)

Correlate classes to user and device activities

(<Ring,Doorbell,Behavior-2>, <Ring,Doorbell,Behavior-3> correlate to motion events)

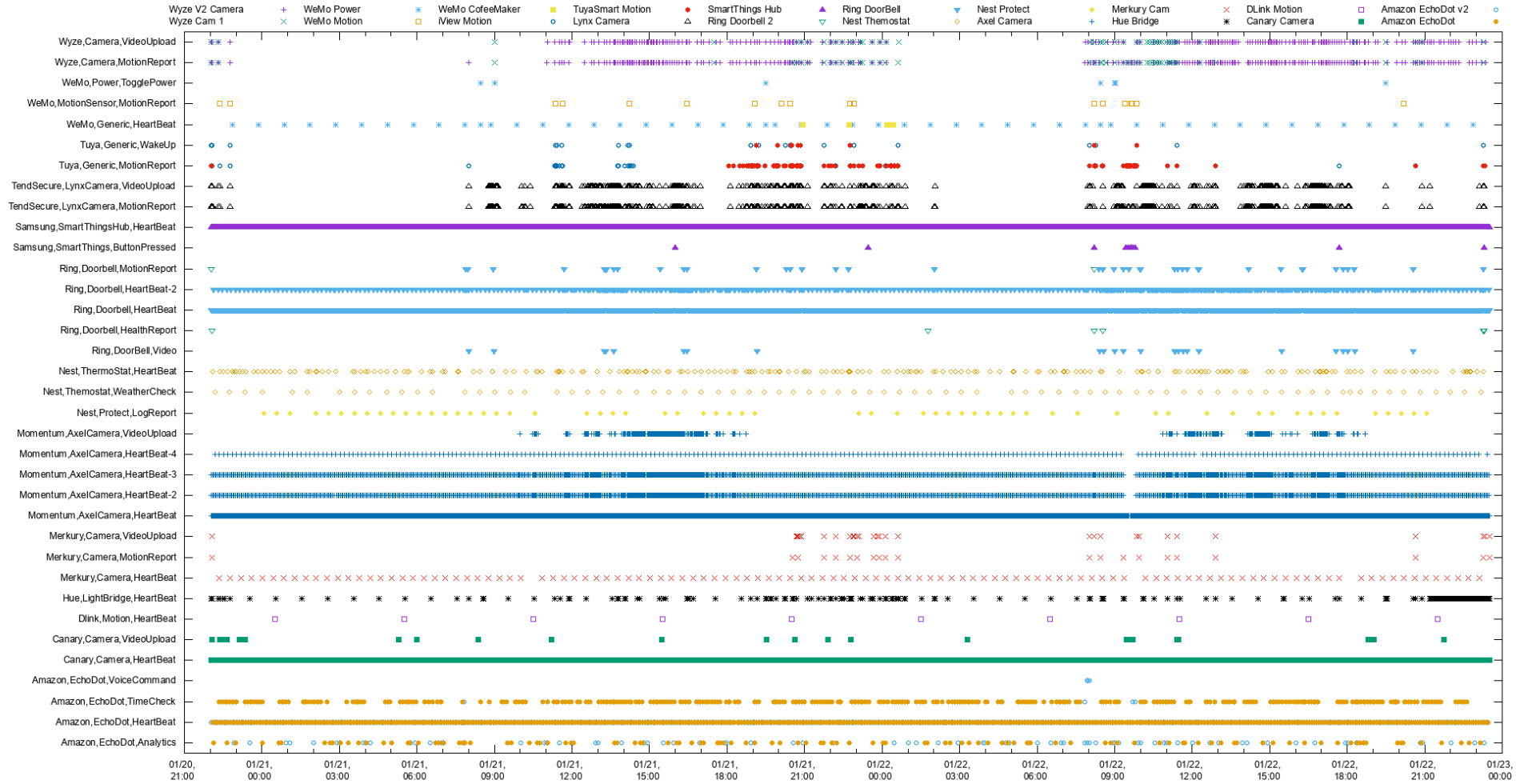
Gain further insight about each class using behavior reporting

(<Behavior-1> is periodic, <Behavior-3> looks most like another video behavior)

Apply descriptive behavior triple labels to each class

(<Ring,Doorbell,Heartbeat>, <Ring,Doorbell,MotionReport>, <Ring,Doorbell,Video>)

Use model to classify future behaviors and identify new behaviors.


Wyze V2 Camera
Wyze Cam 1
WeMo Power
WeMo Motion
WeMo CofeeMaker
iView Motion
TuyaSmart Motion
Lynx Camera
SmartThings Hub
Ring Doorbell 2
Ring DoorBell
Nest Themostat
Nest Protect
Axel Camera
Merkury Cam
Hue Bridge
DLink Motion
Canary Camera
Amazon EchoDot v2
Amazon EchoDot
Wyze,Camera,VideoUpload
Wyze,Camera,MotionReport
WeMo,Power,TogglePower
WeMo,MotionSensor,MotionReport
WeMo,Generic,HeartBeat
Tuya,Generic,WakeUp
Tuya,Generic,MotionReport
TendSecure,LynxCamera,VideoUpload
TendSecure,LynxCamera,MotionReport
Samsung,SmartThingsHub,HeartBeat
Samsung,SmartThings,ButtonPressed
Ring,Doorbell,MotionReport
Ring,Doorbell,HeartBeat-2
Ring,Doorbell,HeartBeat
Ring,Doorbell,HealthReport
Ring,DoorBell,Video
Nest,ThermoStat,HeartBeat
Nest,Themostat,WeatherCheck
Nest,Protect,LogReport
Momentum,AxelCamera,VideoUpload
Momentum,AxelCamera,HeartBeat-4
Momentum,AxelCamera,HeartBeat-3
Momentum,AxelCamera,HeartBeat-2
Momentum,AxelCamera,HeartBeat
Merkury,Camera,VideoUpload
Merkury,Camera,MotionReport
Merkury,Camera,HeartBeat
Hue,LightBridge,HeartBeat
Dlink,Motion,HeartBeat
Canary,Camera,VideoUpload
Canary,Camera,HeartBeat
Amazon,EchoDot,VoiceCommand
Amazon,EchoDot,TimeCheck
Amazon,EchoDot,HeartBeat
Amazon,EchoDot,Analytics
01/20, 21:00
01/21, 00:00
01/21, 03:00
01/21, 06:00
01/21, 09:00
01/21, 12:00
01/21, 15:00
01/21, 18:00
01/21, 21:00
01/22, 00:00
01/22, 03:00
01/22, 06:00
01/22, 09:00
01/22, 12:00
01/22, 15:00
01/22, 18:00
01/22, 21:00
01/23, 00:00

Policy Enforcement

Policy Language: Device, Behavior, Context → Action

```
⟨rule⟩ ::= ⟨action⟩ ⟨device⟩ ⟨behavior⟩ ⟨context⟩                    (1)
⟨action⟩ ::= ⟨alert⟩ | ⟨log⟩ | ⟨pass⟩ | ⟨drop⟩ | ⟨reject⟩ | ⟨redirect⟩  (2)
⟨device⟩ ::= ⟨vendor⟩"-"⟨model⟩                                   (3)
⟨vendor⟩ ::= ⟨const⟩                                              (4)
⟨model⟩ ::= ⟨const⟩                                               (5)
⟨behavior⟩ ::= ⟨vendor⟩ ⟨model⟩ ⟨activity⟩                        (6)
⟨activity⟩ ::= ⟨const⟩                                            (7)
⟨context⟩ ::= ⟨time_ctx⟩ | ⟨enc_ctx⟩ | ⟨mgt_ctx⟩                (8)
⟨time_ctx⟩ ::= "(T:" ⟨time⟩ "-" ⟨time⟩ ")"                       (9)
⟨enc_ctx⟩ ::= "(C:" ⟨encrypted⟩ | ⟨unencrypted⟩ ")"              (10)
⟨mgt_ctx⟩ ::= "(M:" ⟨present⟩ | ⟨notpresent⟩ ")"                  (11)
⟨time⟩ ::= [0-9] [0-9] ":" [0-9] [0-9]                           (12)
⟨const⟩ ::= "'"[A-Za-z0-9_.]+"'"                                 (13)
```

Policy language develops rules to control application behaviors.

Develops OpenFlow FlowMods that enforce policy by:

- matching from historical knowledge of behaviors.
- providing instructions for future matches of behaviors.

Enforcing Policies With HomeSnitch

OFPST_FLOW reply (OF1.3) (xid=0x2):

n_bytes=5488, actions=drop

n_bytes=25974, actions=drop

OFPST_FLOW reply (OF1.3) (xid=0x2):

n_bytes=1936, actions=drop

Amazon,EchoDot,Analytics	2018-10-08 04:56:21	233	Permitted	⏸
Amazon,EchoDot,FirmwareCheck	2018-05-04 18:13:00	1	Paused	▶
Amazon,EchoDot,TimeCheck	2018-10-08 07:51:07	16782	Permitted	⏸
Amazon,EchoDot,VoiceCommands	2018-10-05 16:48:30	12	Paused	▶
Amazon,EchoDot,WiFiCheck	2018-10-06 21:33:11	610	Paused	▶
Aria,Scale,WeightResults	2018-05-04 08:36:18	1	Permitted	⏸
Canary,Camera,HeartBeat	2018-10-08 04:58:23	4870	Paused	▶
Canary,Camera,Video	2018-10-05 16:05:35	38	Permitted	⏸

Limitations

- Communications Protocols
- Online Classification
- Mimicry Attacks

Thank you

- Our work provides a building block for transparency and control of smart-home devices.
- Leverages software defined networking and machine learning to classify behaviors.
- Offers insight into device semantic behaviors and fine-grained control over behaviors.

TJ O'Connor

Wolfpack Security and Privacy Research (WSPR) Lab

NC State University

tjoconno@ncsu.edu

https://www.tjoconnor.org