1.0

HAWatcher

0.8

HAWatcher(Apps Only)

HAWatcher(Mining Only)

0.6

ARM

0.4

OCSVM

0.2

0.0

1

2

3

4

5

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Case ID

1.0

HAWatcher

0.8

HAWatcher(Apps Only)

HAWatcher(Mining Only)

0.6

ARM

0.4

OCSVM

0.2

0.0

1

2

4

5

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Case ID

Figure 8: Recall and precision of HAWatcher and four other detectors for comparison purposes.

events collected during a week, which makes 0.57 false alarms

malware, rather than IoT malfunctions. For example, Home-

per day and a false alarm rate of 0.04%. In comparison, ARM

Guard [33, 34] presents the first systematic categorization

and OCSVM cause 722 and 1,116 false alarms, respectively;

of threats due to interference between different automation

that is, 103 and 159 per day and false alarm rates 7.40% and

apps, dubbed cross-app interference (CAI) threats, such as

11.44%, respectively.

automation conflicts, chained execution, and loop triggering;

it is also the first that uses SMT solvers to systematically de-

6.5

Performance upon Smart App Changes

tect such threats. It conducts symbolic execution to extract

In an appified home, it is common that users change the

automation rules from apps, which is used in this work.

smart apps, such as installing new apps and changing the

PFirewall [32] is a unique work that notices excessive IoT

configuration. However, traditional mining based anomaly

device data continuously flows to IoT automation platforms.

detection needs a long time to adapt to the changes and,

It enforces data minimization, without changing IoT devices

during the adaptation time, may trigger many false alarms.

or platforms, to protect user privacy from platforms.

Handling such changes for anomaly detection in appified

IoTSan [61] statically analyzes smart apps to predict

homes has been challenging. We conduct smart app change

whether the resulting automation may violate any safety

experiments to evaluate HAWatcher's performance and com-

properties. IoTGuard [29] instruments smart apps. Before

pare it with other systems, OCSVM and ARM.

an app issues a sensitive command, the action has to pass

As listed in Table 8, we create five cases of smart app

the policies defined by users. Both rely on pre-defined poli-

changes, which cover changes of trigger, condition, action,

cies, while HAWatcher does not. Unlike our work, which

and the whole rule. For each case, we use one day to collect

detects IoT device anomalies, HoMonit [79] is focused on the data, and then apply HAWatcher, OCSVM, and ARM to

detecting misbehaving smart apps. Given a physical event,

the collected data. The results show that HAWatcher does

Orpheus [31] checks the system call trace due to the event

not trigger any alarms, while OCSVM triggers many alarms

against an automaton to detect attacks; it cannot detect

for all the five cases and ARM for the changes of R8 and

anomalies such as fake events, event interceptions, etc.

R10. We manually inspect the alarms and confirm that they

Many anomaly detection detectors learn normal behaviors

are all false alarms caused by app changes.

of a smart home from its historical data [26, 35, 51, 54, 60, 69,

ARM does not trigger false alarms for the changes of R3,

75,76]. For example, SMART [51] trains multiple user activ-

R5, and R14 because it does not include any association

ity classifiers based on different subsets of sensor readings,

rules covering the devices, such as L1 and L3, involved in the

and further trains another classifier that takes the vector

updated rules. For the OCSVM-based detector, each vector

of activity-classification results as its input to detect sensor

contains four consecutive snapshots of device states. In the

failures. DICE [35] detects anomalies during state transitions

switch(L)

case of R3, for example, the missing Eon

by checking the context. Peeves [26] makes use of data from

causes unseen

vectors and thus triggers false alarms. For HAWatcher, upon

an ensemble of sensors to detect spoofed events.

app changes, the semantics of the updated apps are extracted

The main difference of these existing anomaly detectors

and an updated set of correlations obtained. Thus, it is able

and our work is that HAWatcher extracts various semantics

to handle the changes without triggering false alarms.

(device types, device relations, smart apps and their configu-

ration), and infuses the semantics into the mining process.

7

Related Work

Not only is the detection more accurate, but each detected

anomaly can be interpreted as a violation of a correlation,

With the emerging development of IoT devices and appified

which itself is explainable. Prior to our work, it is unclear

home automation, their security and privacy issues have

how a mining based approach is able to accurately learn

drawn great attention [28, 29, 34, 50, 57, 61, 73, 74, 78, 79].

complex behaviors in an appified home (e.g., Testbed 1 with

Most of them are focused on detecting threats, attacks and

17 apps). HAWatcher provides an effective solution.