Table 7: HAWatcher's detection performance on Tesbed 1. "#inst." indicates the number of instances for one testing case. As

switch is a common attribute for all actuators, we point out the specific appliance controlled by each switch after the colon.

Case

Type

Anomaly Description

Anomaly Creation Method

#inst.

Precision

Recall

Correlations Violated

1

false motion(MS1) active

50

97,77%

86.00%

C26

2

false contact(C1) open

50

100.00%

100.00%

C9

Faulty/Fake

3

false acceleration(C1) active

insert events into the dataset

50

97,87%

92.00%

C27

Events

4

false presence(PS1,PS2) present

50

96.15%

100.00%

C3,C5,C25,C7

5

false button(B) pushed

50

100.00%

100.00%

C8

6

missing motion(MS2) active

57

100.00%

92,98%

C28, C35, C36, C14

7

missing motion(MS3) active

38

100.00%

100.00%

C17

Event Losses/

8

missing contact(C1) open

remove events from the dataset

11

78,57%

100.00%

C3, C5, C27

Interceptions

9

missing presence(PS1,PS2) present

9

77,78%

77,78%

C37

10

missing illuminance(MS3) events

46

100.00%

43.47%

C12,C13

11

Ghost/Fake

turn on switch(P2):fan

50

100.00%

100.00%

C30

12

turn on switch(P3):lamp

toggle from the ghost smart app

50

100.00%

100.00%

C31

Commands

13

turn on switch(L4):light

50

100.00%

100.00%

C19

14

stealthily turn on switch(P2):fan

Stealthy

toggle from the ghost smart app

50

100.00%

100.00%

C6

15

Commands

stealthily turn on switch(P3):lamp

and

50

100.00%

100.00%

C32

16

stealthily turn on switch(L4):light

remove feedback events

50

100.00%

100.00%

C23

17

Command

fail to turn on switch(L1):light

9

100,00%

100.00%

C2

18

Failures (cyber)/

fail to turn on switch(L4):light

12

100.00%

100.00%

C22

19

Command

fail to turn on switch(P2):fan

cut off devices' power supply

10

100.00%

100.00%

C34

20

Interceptions

fail to turn on switch(P4):lamp

53

100.00%

100.00%

C38

21

Command

fail to turn on switch(L1):light

9

100.00%

66.67%

C24

22

Failures (physical)/

fail to turn on switch(L4):light

cover bulbs with paper

12

100.00%

100.00%

C12, C1

23

Denial of

fail to turn on switch(P2):fan

10

100,00%

100.00%

C16

24

Executions

fail to turn on switch(P4):lamp

unplug connected appliances

53

100.00%

100.00%

C10

Avg

-

-

97.83%

94.12%

-

Event Losses/Interceptions. To simulate them, we ran-

6.4

Performance of Anomaly Detection

domly remove events of some devices from the testing event

We first evaluate HAWatcher's precision and recall in detect-

logs. We select various types of devices that users complain

about event losses, such as presence sensors [20], contact

ing anomalies, and compare them with two baseline detectors.

We then measure the false alarm rate of HAWatcher.

sensors [23], and motion sensors [10].

Evaluation Metrics. Given an anomaly case (see Table 7),

precision is the number of correctly detected instances of

Ghost/Fake Commands Both smart lights and plugs have

that case divided by the number of alarms reporting that

been frequently reported by users for turning on/off unex-

anomaly case (i.e., ratio of true anomalies to alarms), recall

pectedly [5,6,12]. We write a ghost smart app, which is not

is the number of correctly detected instances of that case

known by HAWatcher, and use the app randomly issue com-

divided by the number of injected instances of that case (i.e.,

mands to turn on smart lights and plugs.

percentage of anomalies that can be detected), and the false

alarm rate is the number of false positives divided by the

Stealthy Commands With compromised smart lights [65]

number of IoT events.

and plugs [58], attackers can control them to make stealthy

but hazardous actions. We simulate this type of attacks using

True Positive

Precision =

the same method as ghost/fake commands but remove the

True Positive + False Positive

feedback event of each fake command.

True Positive

Recall =

(1)

True Positive + False Negative

Command Failures (cyber)/Command Interceptions

False Positive

False Alarm Rate =

We simulate Command Failures (cyber-part malfunctions)

All Events

and Command Interceptions on smart plugs [11] and smart

lights [7]. We cut the power of target devices to make them

Detectors for Comparison. We compare the performance

irresponsive. For each target device, we conduct the experi-

of HAWatcher with that of two baseline approaches described

ment multiple times during one day.

in Section 6.2, ARM and OCSVM. For the ARM-based detector,

we segment the testing dataset as during the training phase,

and check each segment against all mined rules to detect

Command Failures (physical)/Denial of Executions

anomalies. For the OCSVM-based detector, as in [48], we

Command Failures (physical part malfunctions) and Denial of

take a snapshot of all devices' states as a frame each time a

Executions are simulated on lights [65] and smart plugs [18].

new event arises and concatenate four consecutive frames

We cover smart lights with a lightproof paper, and unplug

as one data vector, which is fed into the trained OCSVM for

appliances from smart plugs. The smart lights and plugs still

detecting anomalies.

respond to commands with feedback events, but those com-

In addition, to evaluate the effect of semantic analysis of

mands would not have any physical effect. For each case, we

smart apps and correlation mining each and also to measure

conduct the experiment multiple times during one day.