Semantic Analysis

Correlation Mining

Smart

Configuration

Adjacency

Device

Training

Command

Apps

Table

Installation

Event Logs

Interception

Location

Smart app channel correlation

Query

(a)

Water detected

Preprocessing

water

detected

Evidence

without

Mismatch

Valve closed

Semantic

Hypothetical

Hypothesis

Analysis

Correlations

Testing

Command Failure

Physical channel correlation

Correlation Refining

Query

Power high

(b)

< 'high power Sswitch on

with

Correlation Refining

Mismatch

Switch off

Event

Contextual

Consequential

Device States

Fake Event

Stream

Checking

Checking

User activity channel correlation

Query

(c)

Presence present

(E)

presence

Econtact

(present

without

Alarms

Mismatch

Contact open

Anomaly Detection

Figure 4: Architecture of HAWatcher.

Figure 5: Detecting anomalies depicted in Figure 1.

5.1 Workflow of Anomaly Detection

```
def installed() {
subscribe(lightSensor, "illuminance", illuminanceHandler)
}
```

The Anomaly Detection module runs parallel with the appi-

fied home automation, and checks the events received from

```
def updated() {
```

IoT devices against the learned correlations to detect anoma-

```
unsubscribe()
```

lies. Figure 5 illustrates how this module detects anomalies,

```
subscribe(lightSensor, "illuminance", illuminanceHandler)
}
```

using anomalies depicted in Figure 1 as examples.

In case (a), the smart app automatically shuts the valve

def illuminanceHandler(evt) {

when water is detected. By applying semantic analysis to

if (evt. .integerValue < 30)

lights.on()

the app, HAWatcher extracts an e2e correlation (Ewater detected

else if integerValue 50)

Evalve ) closed Since attackers intentionally intercept the command

lights.off()

"close the valve" towards the valve, there is no feedback event

}

Evalve 'closed' which contradicts the correlation. Furthermore, if

Figure 6: Code snippet of the app LightUpTheNight.

it is a Command Failure caused by the valve's cyber-part

malfunction, HAWatcher can detect it the same way.

(1). It applies symbolic execution to the Intermediate Repre-

In case (b), the hypothetical e2s correlation 'high power

sentation of apps and captures the configuration information,

sswitch is first proposed based on the physical channel and

achieving precise semantics extraction. The extracted seman-

then gets confirmed using the training event logs. After a

tics of each app is represented as one or more rules, each in

turning-off command is sent to the plug and executed by

the form of a tuple trigger(T)-condition(C)-action(A) which

its cyber part (hence, its Switch=off), however, due to its

means that "if T occurs, when C is true, execute A."

broken relay, the plug still supplies power and thus the power

Step (2), which converts rules to correlations, is straight-

meter reports events of high power usage, which violates the

forward. Assuming T is reflected by the event E1, and E2

aforementioned correlation and triggers an alarm.

is the feedback event due to executing A, the rule above is

In case (c), as the resident does not actually return home,

converted to a correlation (E1 C - E2>.

there is no event Lopen contact that follows the fake event present presence

Taking a SmartThings official app LightUpTheNight [16]

This deviates from the user activity channel correlation

shown in Figure 6 as an example, the Semantic Analysis

(Epresent

presence

Econtact and is thus reported as an anomaly.

module converts it into two e2e correlations: (Elluminance <30

5.2 Semantic Analysis

Eon Light > and < Illuminance 50 Elight) Here, note that the condi-

tion ("Illuminance < 30" or "Illuminance >50") and the trigger

The Semantic Analysis module executes two steps: (1) extract

event in each rule refer to the same attribute of the same

semantics from smart apps and their configuration, such as

device; we thus merge the trigger and the condition to derive

the temperature threshold for turning on AC and which IoT

a concise representation of the trigger events.

devices are bound to which app, and (2) convert the semantics

Moreover, as described in Section 4.1, given an e2e correla-

to correlations.

tion (E.(A) EB(B), extracted from the smart app, we fur-

Semantic analysis has been used to detect malicious or

risky smart apps as in [41, 79]. We use the method de-

ther propose a hypothetical e2s correlation

scribed in our prior work [33,34] to extract semantics in Step

which means that the event EB(B) only arises when so(A) is