

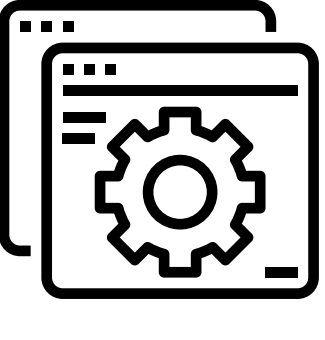
VetIoT: On Vetting IoT Defenses Enforcing Policies at Runtime

Akib Jawad Nafis*, S Mahmudul Hassan*, Omar Haider Chowdhury+, Endadul Hoque*







*Syracuse University, +Stony Brook University

IoT platforms are insecure

Programmable IoT Platform



Smart Apps



A Smart App

Trigger: **Motion Detected**

Action: **Unlock Front Door**

Malicious Interference

High Risks Involved

Several policy enforcing defense mechanisms exist

Problem: Defense mechanisms can fail

Why?

- Evaluated using a few hand-crafted scenarios
- Leaving Hundreds of untested scenarios
- Break security and safety even with a defense in place

Why under-vetted?

Testing requires manual interaction with the platform UI

No automated test-bed exists

Challenges

C1

Test inputs \equiv Long chain of triggering events

➤ How to generate numerous test inputs?

C2

No test oracle exists

➤ How to find expected test results?

Our approach: VetIoT

An automated virtual testbed

- Testing with large-chain of inputs
- Empirically vetting defense solutions
- Easy setups
- Reproducible results

```
graph TD
    subgraph Platform [Programmable IoT Platform]
        Device
        Rule
        Policy
        TargetDefense
    end
    subgraph VetIoT
        TG[Testbed Generator]
        ESG[Event Seq. Generator]
        ES[Event Simulator]
        C[Comparator]
    end
    Config --> TG
    TG -- "1 Testbed Parameters" --> ESG
    ESG -- "2 seq request" --> ES
    ESG -- "3 Ei = (e1, ..., en)" --> ES
    ES -- "4 events (in-order) e1, e2, ..., en" --> C
    C -- "5 system states S0 and Sn" --> ES
    C -- "6 Reports" --> Reports
```

VetIoT: Under the hood

C1

Event Sequence Generator

- Randomly stitch multiple devices' capabilities
- Create long chain to test inputs

C2

Differential testing

- Use same testbed and test input
- Compare baseline (i.e., oracle) vs target-defense for policy violation

```
graph LR
    S0((Initial System State S0)) -- "1 Rule" --> S1((Final System State S1 = Oracle))
    S0 -- "2 Rule Policy Target Defense" --> S2((Final System State S2 = S'))
    S1 -- Oracle --> C[Comparator]
    S2 -- S' --> C
    C --> Reports[Reports]
```

```
graph TD
    Oracle[Oracle baseline] --> D1{Oracle == S'}
    S_prime[S' final state] --> D1
    D1 -- Yes --> NoViolation[No violation]
    D1 -- No --> D2{S == S'}
    S0[S0 initial state] --> D2
    D2 -- Yes --> PolicyViolation[Policy violation]
    D2 -- No --> Indeterminate[Indeterminate]
    PolicyViolation --> DefenseWorked[Defense worked!]
```

Vet defenses against each other

```
graph TD
    Config[config] --> VetIoT1[VetIoT]
    DefenseA[Defense A] --> VetIoT1
    VetIoT1 --> Reports1[Reports]
    Reports1 --> Analyzer[Analyzer]
    Analyzer --> Diff[diff]
    DefenseB[Defense B] --> VetIoT2[VetIoT]
    VetIoT2 --> Reports2[Reports]
    Reports2 --> Analyzer
    Analyzer --> Diff
    Diff --> DefenseDiff[Defense A and B differ!]
```

Evaluation of VetIoT

RQ1. Can VetIoT create IoT testbed and execute IoT experiments automatically?

Results from VetIoT == Results from manual testing.

Defense Mechanism	Number of Experiments	Reproduced Original Eval Results?
ExPAT [1]	8	Yes
PatIoT [2]	4	Yes

RQ2. Can VetIoT empirically vet IoT Defenses?

Individually ExPAT and PatIoT can successfully defend many vulnerable situation.

Test Suite Number	Test cases in Test-suite	Test cases W/ Violation (EXPAT)	Test cases W/ Indeterminate (EXPAT)	Test cases W/ Violation (PATRIOT)	Test cases W/ Indeterminate (PATRIOT)
1	5	1	0	1	0
2	10	3	3	3	0
3	15	3	3	3	0
4	25	8	8	8	0
5	35	9	12	1	1
6	50	3	16	1	1

Vetting [1] and [2] against each other

```
graph LR
    EP[Event, Policy] --> ExPAT[ExPAT]
    EP --> PatIoT[PatIoT]
    ExPAT --> Fail[Fail]
    PatIoT --> Pass[Pass]
```

A sample test-case from test suite 5

```
..
TV;command:ON
LivingRoomTemperature:command:240
..
```

Policy in ExPAT:

Situation: state(LivingRoomWindow) = OPEN
Desire: Expect
Expectation: state(AC) = OFF and state(Heating) = OFF

Equivalent Policy in PatIoT:

POLICY P: ALLOW
action_command = OPEN and action_device =LivingRoomWindow
ONLY IF state(AC)=OFF and state(Heating) = OFF

Execution in ExPAT

Rule: R4 $\xrightarrow{TV = ON}$ LivingRoomWindow.sendCommand(OPEN)
Rule: R10 $\xrightarrow{LivingRoomTemperature > 75}$ AC.sendCommand(ON) ❌

Execution in PatIoT

Rule: R4 $\xrightarrow{TV = ON}$ LivingRoomWindow.sendCommand(OPEN)
Rule: R10 $\xrightarrow{LivingRoomTemperature > 75}$ AC.sendCommand(ON) ✅

[1] Expat - <https://github.com/expat-paper/expat>
[2] Patriot - <https://github.com/yahyazadeh/patriot>

ECS Research Day