Ruledger: Ensuring Execution Integrity in Trigger-Action IoT Platforms

Jingwen Fan, Yi He, Bo Tang, Qi Li, Ravi Sandhu

March 04, 2021
IoT Connections Are Growing Exponentially

• Global IoT connections will reach 25 billion by 2025.*
• IoT users need flexible and easy integrations.

Rule Executions in Trigger-Action IoT Platforms

- Provides flexible and easy integrations using user-defined rules
- Helps to achieve inter-device automation along the rule execution path.

“If user arrives home, then unlock the door and turn on the light.”

- If [This] Then [That]

- Create rules

- Provided by Device Vendor

- IoT Gateways

- IoT Edge Devices

- IoT Platforms

- OAuth

- Amazon Alexa
- Google Assistant
- SmartThings
Security Threats To The Rule Execution

- Two types: API-level attacks, platform/device compromise attacks.

Platform Compromise Attacks
Platform API Attacks
Event Spoof

Trigger-Action Platforms

"If no one is at home, then unlock the door…"

Enter the home mode
Unlock the door

Device API Attacks
Things Need To Be Done In Smart Home Systems

1. Protect rule configurations of the trigger-action platforms and IoT platforms.

2. Detect event spoofs generated from the devices.

3. Prevent malicious rule execution requests from the trigger-action platforms.

We cannot fix all the vulnerabilities.
There does not exist any verification mechanism to protect the integrity of rule execution among different IoT system components.
We Need A Novel IoT Platform

Bob's Rule = “If the user arrives home, then unlock the door and turn on the light.”

User's Rule: Bob
Triggering Event: User Arrives Home
Action Event: Unlock the door

Verify: triggering event
Generate: id of next step

Verify: action event’s id, triggering event record, record of Bob’s rule

{Verify: triggering event
Generate: id of next step}
Our Framework: Ruledger
a distributed ledger-based IoT platform for event and rule verification
# Challenges and Solutions

<table>
<thead>
<tr>
<th>Goal &amp; Challenges</th>
<th>Solutions</th>
</tr>
</thead>
</table>
| Ensure the authenticity and integrity of rule configurations. | Rule Commits Module:  
  • A set of prefab scripts that wrap device APIs of different vendors.  
  • Unified IoT events for the trigger-action platforms. |
| Ensure rule executions on correct devices. Prevent privacy breaches. | Triggering Event Verification Module:  
  • Algorithm 1: Check Trigger Condition  
  • Algorithm 2: Triggering Event Verification |
| Check the trigger conditions and verify the triggering events. Make least changes to the current IoT platforms and device platforms. | Action Verification Module:  
  • Algorithm 3: Action Verification |
| Ensure the integrity of rule executions. Develop a stateful verification framework. | |
“If the user's heart rate is abnormal, then unlock the front door and call the doctor.”

The Trigger
If the user is inactive and heart rate is above 120 BPM, submit heart rate alert event

Fake Trigger Events

Set Rule

Malicious Action Requests

The Action
If heart rate alert event happens, then unlock the door and call the doctor.

Apple Smart Watch Cloud API

Execution Agent

IFTTT
Microsoft Flow

SmartThings Cloud API

Execution Agent

Task Agent

Trigger Verification Smart Contact

Task Record Smart Contact

Action Verification Smart Contact

Distributed Ledger
Triggering Event Verification

Smart Watch

IoT Log Service

IoT Gateway

Execution Agent

Device Vendor Cloud Service

Wallet-based SDK

Smart Contract

Triggering Event Verification Service

- Write event_log

- event id, log_key

- result_checksum

- event id, log_key

- verify event_log

- submit event_log
1. verify the message, prevent reply attacks.
2. check the triggering event record, prevent event spoof attacks.
   • authenticity
   • correctness
3. check the rules in the ledger, prevent false triggering of the rules.
Evaluation

The deployment of Ruledger

- We use 7 elastic cloud servers to deploy the ledger service.
- We use real IFTTT platform and SmartThings Platform to measure the end-to-end latency.
- Due to the rate-limit mechanism of these platforms, we need to implement simulating services to measure the throughput.
Performance Evaluation

rule = “If the user's heart rate is above 120 times per second, then unlock the door.”

1. Performance of Ruledger Modules
   • Latency
     
     | Ruledger Module              | Latency Incurred by Modules |
     |-------------------------------|-----------------------------|
     | trigger event verification    | 32.45 ms                    |
     | action verification           | 32.83 ms                    |

   total latency of the smart contract modules <70ms 4.36% of the whole rule execution latency.

   • Throughput
     
     use the execution agent and the task agent to submit different number of concurrent requests to the smart contract module.
Performance Evaluation

2. Performance of The Entire System

• End-to-End Latency

<table>
<thead>
<tr>
<th></th>
<th>SmartThings</th>
<th>Ruledger</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>End-to-end execution latency</td>
<td>1.403 s</td>
<td>1.604 s</td>
<td>12.53%</td>
</tr>
<tr>
<td>average of 30 trials</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SmartThings: Baseline system that directly connects SmartThings and IFTTT.
Ruledger: Deploy Ruledger as a middleware between SmartThings and IFTTT.
Set simulated devices for a smart watch and a smart lock in SmartThings.

• Throughput

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Ruledger</th>
<th>Decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput (req/s)</td>
<td>43.37</td>
<td>40.57</td>
<td>6.5%</td>
</tr>
</tbody>
</table>

implement a skeleton device simulator and a trigger-action service.
2000 concurrent requests using Apache JMeter.
Baseline: the execution agent sends the trigger event directly to the task agent.
Summary

- Ruledger: A ledger based IoT platform to protect the integrity of rule executions in trigger-action based smart home system.
  - wallet-based agents record stateful information generated by smart home systems in the ledger during rule executions.
  - smart contracts automatically verify the authenticity of the information according to the tamer-proof ledger records.

- State generation and verification algorithm built upon ledgers and wallet-based agents to ensure the stateful information are properly submitted and verified.
  - check trigger conditions, triggering event verification
  - action verification

- Prototype Ruledger with a real trigger-action platform and a real IoT system SmartThings.
  - acceptable overhead, feasible to be deployed in large scale.
Thank you for your time!