SITUATION-AWARE ACCESS CONTROL FOR INTELLIGENT TRANSPORTATION SYSTEMS

Adams Ubini
Mentor: Dr. Zhiyuan Chen
Summer 2023
Background

- Intelligent Transportation System: ground transportation technologies
Existing method

- The most used access control solution in ITS is role-based access control, however,
- Situation-aware access control is more appropriate for ITS because the access control decisions often depend on dynamically changing situations.
- For instance, a driver can only access availability of a parking spot when the driver's vehicle is on the associated parking lot.
Statement of the Research Problem

• To propose situation-aware access control framework for ITS

• Show that the overhead of enforcing situation-aware access control rules is acceptable.
Methods/Approach

• We created an ontology for 4 ITS use cases, including major classes such as users, vehicles, infrastructure, roles, data, events, topics.

• We proposed a query rewriting method that modifies a SPARQL query over ITS data to enforce access control rules.
We created a small TTL dataset of 157 triples then we generated more data using a python program to 1305 triples (medium dataset) and 11714 triples (large dataset).

Program to scale up data

```python
def increase_numbers_in_string(input_string):
    def increase_number(match):
        number = int(match.group(0))
        return str(number + 1)

    pattern = r'\d+'
    result_string = re.sub(pattern, increase_number, input_string)
    return result_string

# Test the function
input_string = "its:vehicle\d+ its:has its:licence_plate\d+ ."
x = 0
while x < 105:
    input_string = increase_numbers_in_string(input_string)
    print(input_string)
x+=1
```
Results/Evaluation

- **Rule 2-Smart Parking System:**

  **Original Query**
  Select ?status
  Where {
  its:parkingLot1 its:contains ?Parking_Space .
  }

  **Rewritten query**
  Select ?status ?Parking_Space ?locationLot
  Where {
  its:user1 its:owns its:vehicle1 .
  its:parkingLot1 its:contains ?Parking_Space .
  its:parkingLot1 its:is_at ?locationLot .
  its:vehicle1 its:is_at ?locationLot .
  its:user1 its:subscribes_to its:SmartParkingTopic .
  }
Results/Evaluation

- Rule 2 - Smart Parking System:

<table>
<thead>
<tr>
<th>Smart Parking Topic</th>
<th>Original time (ms)</th>
<th>Rewritten time (ms)</th>
<th>Original size (Triples)</th>
<th>Rewritten size</th>
</tr>
</thead>
<tbody>
<tr>
<td>small dataset (Triples=157)</td>
<td>2.6</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>medium dataset (Triples=1305)</td>
<td>3</td>
<td>3.2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>large dataset (Triples=11714)</td>
<td>3.4</td>
<td>6.2</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

0.35% percentage difference
Results/Evaluation

• Rule 4-Accident Report Event:

Original query
Select ?message
where {
  its:user4 its:owns ?vehicle .
  ?vehicle its:drives_on ?segment .
  ?location its:is_at ?segment .
}

Rewritten Query
Select ?message ?location
where {
  its:user4 its:owns ?vehicle .
  ?vehicle its:drives_on ?segment .
  its:user4 its:subscribes_to its:AccidentReport .
  ?location its:is_at ?segment .
}
Results/Evaluation

- Rule 4 - Accident Report Event:

<table>
<thead>
<tr>
<th>Accident Report Event</th>
<th>Original time (ms)</th>
<th>Rewritten time (ms)</th>
<th>Original size (Triples)</th>
<th>Rewritten size</th>
</tr>
</thead>
<tbody>
<tr>
<td>small dataset (Triples=157)</td>
<td>2.4</td>
<td>2.8</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>medium dataset (Triples=1305)</td>
<td>3</td>
<td>3.2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>large dataset (Triples=11714)</td>
<td>3</td>
<td>3.2</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

0.1% percentage difference
Conclusion

• The overhead of enforcing access control rules ranges from 0-0.65% over 4 test queries

• Overhead varies by queries, data sizes, result sizes.
Future Work

Use reinforcement learning to dynamically adjust access control, e.g., to block a user who keeps posting inaccurate messages.
Reference


