Midterm Report: Access Control For a Database-Defined Network

Noemi Glaeser
Software-Defined Networking (SDN)
Software-Defined Networking (SDN)
Ravel: A Database Controller

- Using database as the controller

Diagram:
- Software
- Ravel
- View
- SQL queries
- Table
- Network
- PostgreSQL
Ravel in Action

routing app: check broken path, re-route

SQL rule: upon broken path, re-route

Network table

shortest path view

topology table

configuration table

Network
Ravel in Action

Routing app: check broken path, re-route

SQL rule: upon broken path, re-route

Network table

Topology table

Configuration table

Link down

Network

3
Routing app: check broken path, re-route

SQL rule: upon broken path, re-route

Topology table:

<table>
<thead>
<tr>
<th>sid</th>
<th>nid</th>
<th>active</th>
</tr>
</thead>
<tbody>
<tr>
<td>172</td>
<td>39</td>
<td>1</td>
</tr>
<tr>
<td>172</td>
<td>39</td>
<td>0</td>
</tr>
</tbody>
</table>

Network link (172,39) down

Network link (172,39) down

Configuration table:

<table>
<thead>
<tr>
<th>sid</th>
<th>nid</th>
<th>active</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Ravel in Action

Routing app: check broken path, re-route

SQL rule: upon broken path, re-route

Network link (172,39) down

Network link (172,39) down

Topology table

Configuration table

App view

Topology view

Shortest path view
Ravel in Action

Routing app: check broken path, re-route

Network table -> shortest path view

Configuration table -> shortest path view

SQL rule: upon broken path, re-route

Network link (172,39) down

Topology table:
- ... {...172,39,156,...}
+ ... {...172,38,148,...}

Network link (172,39) down

Configuration table:
- ... 172 39
+ ... 172 39
- ... 172 38
+ ... 172 148
Ravel in Action

Routing app: check broken path, re-route

Network table

Topology table

Configuration table

SQL rule: upon broken path, re-route

Topological table: shortest path

Network link (172,39) down

App view

Configuration

Topology

Network link (172,39) down

- 172 39 1
+ 172 39 0

Configuration table

Network table

Topology

App view

Routing app: check broken path, re-route

Network table

Topology table

Configuration table

SQL rule: upon broken path, re-route

Topological table: shortest path

Network link (172,39) down

- 172 39 1
+ 172 39 0

Configuration table

Network table

Topology
Toward a Secure Database-Defined Network

• Critical but less-studied aspect in SDN today

• Most SDN controllers do not implement security

• Security requirements still under development
  – Direction of information flow
  – Access control

• Currently, the Ravel controller exposes all network states to users

• This project: enhance Ravel with access control support
Toward a Secure Database-Defined Network

- Critical but less-studied aspect in SDN today
- Most SDN controllers do not implement security
Toward a Secure Database-Defined Network

- Critical but less-studied aspect in SDN today
- Most SDN controllers do not implement security
- Security requirements still under development
Toward a Secure Database-Defined Network

- Critical but less-studied aspect in SDN today
- Most SDN controllers do not implement security
- Security requirements still under development
  - Direction of information flow
Toward a Secure Database-Defined Network

• Critical but less-studied aspect in SDN today
• Most SDN controllers do not implement security
• Security requirements still under development
  – Direction of information flow
  – Access control
Toward a Secure Database-Defined Network

• Critical but less-studied aspect in SDN today
• Most SDN controllers do not implement security
• Security requirements still under development
  – Direction of information flow
  – Access control
• Currently, the Ravel controller exposes all network states to users
Toward a Secure Database-Defined Network

- Critical but less-studied aspect in SDN today
- Most SDN controllers do not implement security
- Security requirements still under development
  - Direction of information flow
  - Access control
- Currently, the Ravel controller exposes all network states to users
- This project: enhance Ravel with access control support
Strawman Solution

• Explicit specification (principal, object, privilege)
Strawman Solution

- Explicit specification (principal, object, privilege)

  (‘alice’, topology, insert)
Strawman Solution

• Explicit specification (principal, object, privilege)
  (`alice`, topology, insert)

• Manual, tedious
Strawman Solution

• Explicit specification (principal, object, privilege)
  (‘alice’, topology, insert)

• Manual, tedious
• Does not scale up
Our approach: ACL in Ravel

• Access control

Authenticate at database login

Higher-level, finer-grained access control via SQL

Network

SQL queries

Table

Authorization views

Alice, Bob, Charlie
Higher-level, finer-grained

• Advantages
Higher-level, finer-grained

• Advantages
  – dynamic
Higher-level, finer-grained

- Advantages
  - dynamic
  - content-based
Higher-level, finer-grained

- Advantages
  - dynamic
  - content-based

}\ intent rather than extent
Higher-level, finer-grained

- Advantages
  - dynamic
  - content-based

\{ \text{intent} \} \text{ rather than extent}

A network table with n columns:

| object(_,_,…,_ | … |
|----------------|

SQL query over data in object and other parts of the network database:

| object | … |
|--------|

Access control view with n+1 columns:

| object_acl | principal | … |
|------------|-----------|
| …          |           |

…
Enforcing Access Control

access control view
topology_acl(principal, __, __, __, __)

reachability matrix

<table>
<thead>
<tr>
<th>fid</th>
<th>src</th>
<th>dst</th>
<th>vol</th>
<th>fw</th>
<th>lb</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>14</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>15</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>14</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>15</td>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

...
### Enforcing Access Control

The access control view is defined as:

\[
\text{topology acl} (\text{principal}, \_\_\_\_, \_\_\_, \ldots, \_)
\]

The access control is enforced by querying the `topology_acl` table to find the principal associated with the current user:

\[
\text{select * from}\ 
\text{topology acl where principal = current_user}
\]

#### Access Control Table

<table>
<thead>
<tr>
<th>topology_acl</th>
</tr>
</thead>
<tbody>
<tr>
<td>charlie</td>
</tr>
<tr>
<td>alice</td>
</tr>
<tr>
<td>bob</td>
</tr>
<tr>
<td>alice</td>
</tr>
</tbody>
</table>

#### Tenant Table

<table>
<thead>
<tr>
<th>topology_tenant</th>
</tr>
</thead>
<tbody>
<tr>
<td>alice</td>
</tr>
</tbody>
</table>

The principal `alice` is selected from the `topology_acl` table and then inserted into the `topology_tenant` table.
CREATE OR REPLACE VIEW topology_acl AS (  
    ( SELECT 'admin' AS principal, sid, nid FROM tp )  
  UNION  
    ( SELECT s.name AS principal, sid, nid FROM tp, sla s  
      WHERE tp.sid IN (SELECT nodeid FROM sla WHERE name=s.name)  
          AND tp.nid IN (SELECT nodeid FROM sla WHERE name=s.name)  
    ) );

CREATE OR REPLACE VIEW topology_public AS (  
    SELECT sid, nid FROM topology_acl  
    WHERE principal = current_user);  

GRANT SELECT ON topology_public TO PUBLIC;
Conclusion

- SDN: Programming networking with software via a centralized controller
- Security: an important but less visited aspect
- This project: Adding security to a database controller for SDN
Conclusion

• SDN: Programming networking with software via a centralized controller

• Ravel: a database controller
Conclusion

- SDN: Programming networking with software via a centralized controller
- Ravel: a database controller
- Security: an important, but less-visited aspect
Conclusion

• SDN: Programming networking with software via a centralized controller

• Ravel: a database controller

• Security: an important, but less-visited aspect

• This project: Adding access control to a database controller for SDN
References


