INFORMATION SHARING IN CLOUD

Amy(Yun) Zhang

Presented by Amy
May 1, 2014
ICS-UTSA
Cyber security incident response

- share information securely amongst a set of entities/organizations
  - often ad hoc
  - sensitive information involved
  - need an isolated domain to do the collaboration
➢ Agile Incident Response

cited from presentation of Dr. RAM
• Systematically study authorization aspects in a simple inter-organizational collaboration scenario

- cited from presentation of paper <Towards a Framework for Group-Centric Secure Collaboration>
SCENARIOS

➢ secure information sharing for cyber incident response

❖ high assurance (closed network)
  » unusual activity in military (Air Force, Navy & Army) networks
  » a physically secure and air-gapped meeting room with members from corresponding military organizations
  » member bring sensitive data for analysis and collaboration
  » strict control on data import/export

❖ medium assurance (community)
  » Cyber incidents across critical infrastructure providers in a community
    ▪ emergency response, healthcare, banks, utility
  » Need a community information sharing platform
    ▪ controlled access
High-assurance scenario

Cloud in a box (air-gapped meeting room)
Secure Isolated Domain

u1, u2, u3 ..., uN

Org1

Org2

OrgX

Org3

OrgN

OrgX

Community Cloud

Medium-assurance scenario
Proposed Models of High-Assurance Scenario
(Cloud-in-a-box)
domain (admin)

shared project1 (U1, U2, U3)

shared project2 (U1, U2)

shared project3 (U2, U3)

shared swift project (admin) — data sharing

shared project1 container: rw(U1, U2, U3)

shared project2 container: rw(U1, U2)

shared project3 container: rw(U2, U3)
HIGH-ASSURANCE MODEL 2

domain1(U1)

- project1(U1)
- shared project(U1,U2)
- swift account

domain2(U2)

- project2 (U2)
- project3 (U3)

get result

copy data

copy data

copy data
 ➢ shared container in swift
 ➢ utilize Zero-VM

swift project1(U1)

container: rw(U1)

copy data

get result

swift project2(U2)

container: rw(U2)

copy data

get result

shared swift project(admin)

shared container: rw(U1, U2)

ZeroVM
Proposed Models of Medium-Assurance Scenario
MODEL1

shared domain (domain admin)

domain1

P1(U1)

domain2

P2(U2)

domain3

P3(U3)

shared project (U1,U2,U3)

swift account

SID
MODEL2

domain (admin)

P1(U1)

P2(U2)

P3(U3)

SID

shared project (U1,U2,U3)

swift account
MODEL 1-1

shared domain (admin)

SID

shared project (U1,U2,U3)

shared swift project (admin)

U1 cont: r(U2,U3); rw(U1)
U2 cont: r(U1,U3); rw(U2)
U3 cont: r(U1,U2); rw(U3)
shared cont: rw(U1,U2,U3)

domain1
P1(U1)
P2(U2)
P3(U3)
domain2
domain3
MODEL 2-1

domain (admin)

- P1(U1)
- P2(U2)
- P3(U3)

SID

- shared project (U1,U2,U3)
  - U1 cont: r(U2,U3); rw(U1)
  - U2 cont: r(U1,U3); rw(U2)
  - U3 cont: r(U1,U2); rw(U3)
  - shared cont: rw(U1,U2,U3)

- shared swift project (admin)
Some Implementation details
**SID TABLE**

**Tips:**

- Domain admin create the table for every unique request and map the table according to the request.
- All the members in the group have to send the request to create the group.
- Each time a user requests to leave the group, update the table.

<table>
<thead>
<tr>
<th>SID</th>
<th>U1</th>
<th>U2</th>
<th>U3</th>
<th>U4</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID1</td>
<td>c</td>
<td>m</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>SID2</td>
<td>m</td>
<td>c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SID3</td>
<td>c</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
</tbody>
</table>
REQUESTS SEQUENCE

U1

join SID

return SID id

request SID info

return SID info

access resource with token

return resource

leave SID

Keystone

Nova
SID CREDENTIAL

Steps:

❖ U1 send join_SID request
❖ Keystone return SID ID
❖ U1 request SID info
  ○ waiting for U2, U3
  ○ ready
❖ Keystone return SID info
  ○ shared project name
  ○ shared swift url
    ▪ user’s container url(name)
    ▪ SID member’s container url(name)
    ▪ shared container url(name)
thank you!

Cited from google search.