Secure Information and Resource Sharing in Cloud Infrastructure as a Service

Cyber Incident Response
Models for Information and Resource Sharing

Amy(Yun) Zhang, Ram Krishnan, Ravi Sandhu
Institute for Cyber Security
University of Texas at San Antonio
San Antonio, TX 78249
Nov 03, 2014

Presented by: Amy(Yun) Zhang
Information Sharing and Coordination Initiatives

• collaboration and coordination to enhance situational awareness
  – Share malicious activities on federal systems
  – Technologies, tools, procedures, analytics
Electric Grid Scenario

- Cyber incidents in electricity providers
  - Local utilities, regional, state, national operators
- Need a standing platform that facilitates sharing
  - Controlled access

UTSA
Scope

• Focus on technical challenges
• Sharing amongst a set of organizations
  – Information, infrastructure, tools, analytics, etc.
  – May want to share malicious or infected code/systems
    (e.g. virus, worms, etc.)
  – Sensitive
  – Often ad hoc
• What are the effective ways to facilitate sharing in such circumstances?
  – Information sharing models
  – Infrastructure, technologies, platforms
Cyber Infrastructure for Sharing

• Traditional platforms
  – Shared storage
    • SharePoint, Dropbox, Google Drive, etc.
  – Shared infrastructure
    • Grid computing

• Modern platform
  – Cloud
Cloud IaaS Advantages for Cyber Incident Sharing

• Virtualized resources
  – Theoretically, one can take a snapshot and mobilize

• Operational efficiency
  – Light-weight and agile
  – Rapid deployment and configuration
  – Dynamic scaling
  – Self-service
Cloud IaaS Challenges for Cyber Incident Sharing

• IaaS clouds lack secure sharing models
  – Storage
  – Compute
  – Networks

• Need ability to snapshot tenant infrastructure, share, and control who can access
  – Share by copy
Sharing Model in Cloud IaaS

Can create multiple secure isolated projects (SIPs) within SID with different controls
OpenStack

- OpenStack
  - Dominant open-source cloud IaaS software

Ref: http://www.openstack.org
OpenStack Access Control (OSAC)
OSAC-SID

- Projects (P)
- Domains (D)
- Secure Isolated Projects (SIP)
- Secure Isolated Domain (SID)
- Roles (R)
- Users (U)
- Roles (R)
- SIP-Role Pair (SIPRP)
- Project-Role Pair (PRP)
- Services (S)
- PRMS
- Object Type (OT)
- Operations (OP)
Conceptual Model

- Administrative Model
- Operational Model

Collaboration Group

Establish/Disband

Create RO/RW Subject
Kill Subject
Create Object
Read/Update Version
Suspend/Resume Version

Import Version

ORG A
- Join User
- Leave User
- Add Version
- Remove Version
- Merge Version
- Substitute User

ORG B
- Join User
- Leave User
- Add Version
- Remove Version
- Merge Version
- Substitute User
## OSAC-SID Administrative Model

<table>
<thead>
<tr>
<th>Operation</th>
<th>Authorization Requirement</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>SipCreate(uSet, sip) /* a set of domain admin users together create a sip */</td>
<td>$\forall u_1, u_2 \in uSet.( (DA(u_1)=True \land DA(u_2)=True \land u_1 \neq u_2 \land UO(u_1) \neq UO(u_2)))$ &lt;br&gt; sip $\in$ (UNIV SIP - SIP)</td>
<td>SIPO(sip) = $\bigcup_{u \in uSet} UO(u)$ &lt;br&gt; SIPU(sip) = uSet &lt;br&gt; $\forall u \in uSet.\text{SIPA}(u) = \text{SIPA}(u) \cup {\text{sip}}$ &lt;br&gt; SIP' = SIP $\cup {\text{sip}}$</td>
</tr>
<tr>
<td>SipDelete(uSet, sip) /* delete the sip*/</td>
<td>$\forall u \in uSet.( (DA(u)=True \land sip \in \text{SIPA}(u)) \land$ &lt;br&gt; sip $\in$ SIP</td>
<td>SIPO(sip) = NULL &lt;br&gt; SIPU(sip) = NULL &lt;br&gt; $\forall u \in uSet.\text{SIPA}(u) = \text{SIPA}(u) - {\text{sip}}$ &lt;br&gt; SIP' = SIP $- {\text{sip}}$</td>
</tr>
<tr>
<td>SidCreate(uSet, sid) /* a set of domain admin users together create a sid */</td>
<td>$\forall u_1, u_2 \in uSet.( (DA(u_1)=True \land DA(u_2)=True \land u_1 \neq u_2 \land UO(u_1) \neq UO(u_2)))$ &lt;br&gt; sid $\in$ (UNIV_SID - SID)</td>
<td>SIDO(sid) = $\bigcup_{u \in uSet} UO(u)$ &lt;br&gt; SIDS' = SID $\cup {\text{sid}}$</td>
</tr>
<tr>
<td>SidDelete(uSet, sid) /* delete the sid*/</td>
<td>$\forall u \in uSet.( (DA(u)=True \land sid \in \text{SIDA}(u)) \land$ &lt;br&gt; SIDO(sid) = $\bigcup_{u \in uSet} UO(u)$ &lt;br&gt; sid $\in$ SID</td>
<td>SIDO(sid) = NULL &lt;br&gt; SIDS' = SID $- {\text{sid}}$</td>
</tr>
<tr>
<td>UserAdd(admin, r, u, sip) /* sip admin add a normal user to a sip*/</td>
<td>sip $\in$ SIPA(admin) $\land$ DA(admin) = True $\land$ &lt;br&gt; UO(admin) $\in$ SIDO(sid) $\land$ sip $\in$ sid $\land$ UO(u) = UO(admin) $\land$ r $\in$ R $\land$ sip $\in$ SIP $\land$ u $\in$ U</td>
<td>(u, (sip, r)) $\in$ SIPUA $\land$ &lt;br&gt; SIPU'(sip) = SIPU(u) $\cup {u}$</td>
</tr>
<tr>
<td>UserRemove(admin, r, u, sip) /* sip admin remove a normal user from a sip*/</td>
<td>sip $\in$ SIPA(admin) $\land$ DA(admin) = True $\land$ &lt;br&gt; UO(admin) $\in$ SIDO(sid) $\land$ sip $\in$ sid $\land$ UO(u) = UO(admin) $\land$ r $\in$ R $\land$ sip $\in$ SIP $\land$ u $\in$ U $\land$ (u, (sip, r)) $\in$ SIPUA</td>
<td>(u, (sip, r)) = NULL $\land$ &lt;br&gt; SIPU'(sip) = SIPU(u) $-$ {u}</td>
</tr>
<tr>
<td>CopyObject(u, so1, c1, p, d, so2, c2, sip, sid)</td>
<td>so1 $\in$ SO $\land$ c1 $\in$ C $\land$ p $\in$ P $\land$ SIP $\land$ d $\in$ D $\land$ SID $\land$ so2 $\in$ (UNIV SO - SO) $\land$ c2 $\in$ C $\land$ sip $\in$ P $\land$ SIP $\land$ sid $\in$ D $\land$ SID $\land$ (so1, c1) $\in$ SOO $\land$ (c1, p) $\in$ CO $\land$ (p, d) $\in$ PO $\land$ SIPO $\land$ (c2, sip) $\in$ CO $\land$ (sip, sid) $\in$ PO $\land$ SIPO $\land$ u $\in$ U $\land$ (u, (pip, r)) $\in$ SPUA</td>
<td>SO' = SO $\cup {so2}$ &lt;br&gt; SOO' = SOO $\cup {(so2, c2)}$</td>
</tr>
</tbody>
</table>

† uSet: a set of domain admin users.
## OSAC-SID Operational Model

<table>
<thead>
<tr>
<th>Operation</th>
<th>Authorization Requirement</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nova:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CreateVM(vm, sip, u)</td>
<td>(vm \in (\text{UNIV}_{\text{VM}} - \text{VM}) \land \text{sip} \in \text{SIP} \land ) (u \in U \land \exists (\text{perms}, r) \in \text{PA}.( \text{perms} = (\text{vm, create}) \land (u, (\text{sip}, r)) \in \text{SIPUA}))</td>
<td>(\text{VM}' = \text{VM} \cup {\text{vm}}) (\text{VMO}' = \text{VMO} \cup {(\text{vm}, p)})</td>
</tr>
<tr>
<td>DeleteVM(vm, sip, u)</td>
<td>(vm \in \text{VM} \land \text{sip} \in \text{SIP} \land ) (u \in U \land \exists (\text{perms}, r) \in \text{PA}.( \text{perms} = (\text{vm, delete}) \land (u, (\text{sip}, r)) \in \text{SIPUA}))</td>
<td>(\text{VM}' = \text{VM} - {\text{vm}}) (\text{VMO}' = \text{VMO} - {(\text{vm}, p)}) (\text{vm} = \text{NULL})</td>
</tr>
<tr>
<td>Swift:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CreateContainer(c, sip, u)</td>
<td>(c \in (\text{UNIV}_{\text{C}} - \text{C}) \land \text{sip} \in \text{SIP} \land ) (u \in U \land (u, (\text{sip}, r)) \in \text{SIPUA}))</td>
<td>(\text{C}' = \text{C} \cup {c}) (\text{CO}' = \text{CO} \cup {(c, p)})</td>
</tr>
<tr>
<td>DeleteContainer(c, sip, u)</td>
<td>(c \in \text{C} \land \text{sip} \in \text{SIP} \land ) (u \in U \land (u, (\text{sip}, r)) \in \text{SIPUA}))</td>
<td>(\text{C}' = \text{C} - {c}) (\text{CO}' = \text{CO} - {(c, p)}) (\text{c} = \text{NULL})</td>
</tr>
<tr>
<td>UploadObject(so, c, sip, u)</td>
<td>(\text{so} \in \text{UNIV}_{\text{SO}} \land c \in \text{C} \land \text{sip} \in \text{SIP} \land ) (u \in U \land (u, (\text{sip}, r)) \in \text{SIPUA})) (\text{if } \exists \text{so}' \in \text{SO}.( \text{so} = \text{so}') \text{, then } \text{so}' = \text{so})</td>
<td>(\text{SO}' = \text{SO} \cup {\text{so}}) (\text{SOO}' = \text{SOO} \cup {(\text{so}, c)})</td>
</tr>
<tr>
<td>DownloadObject(so, c, u, p)</td>
<td>(\text{so} \in \text{SO} \land c \in \text{C} \land \text{sip} \in \text{SIP} \land ) (u \in U \land (u, (\text{sip}, r)) \in \text{SIPUA}))</td>
<td>(\text{SO}' = \text{SO} - {\text{so}}) (\text{SOO}' = \text{SOO} - {(\text{so}, c)}) (\text{so} = \text{NULL})</td>
</tr>
<tr>
<td>DeleteObject(so, c, sip, u)</td>
<td>(\text{so} \in \text{SO} \land c \in \text{C} \land \text{sip} \in \text{SIP} \land ) (u \in U \land (u, (\text{sip}, r)) \in \text{SIPUA}))</td>
<td></td>
</tr>
</tbody>
</table>
SID and SIP in OpenStack

CPS
Admin: CPSadmin
Users: Alice@CPS, Bob@CPS

IT-CPS

Create

SID-Critical-Infrastructure

Admins: CPSadmin, SAWSadmin
Users: Alice@CPS, Harry@SAWS

SIP-PortScanning
SIP-DOS

Join

SAWS
Admin: SAWSadmin
Users: Harry@SAWS

T-SAWS

member

SAPD
Admin: SAPDadmin
Users: Martin@SAPD

IT-SAPD

member

Share objects, VMs, etc.

UTSA

The Institute for Cyber Security
Conclusion and future work

• Developed sharing models
  – Formal specification

• Enhanced OpenStack with SID/SIP capabilities
  – Cyber incident response capabilities
    • Self-service
    • SID/SIP specific security
    • Share data, tools, etc. in an isolated environment
    • Ability to execute and analyze malicious code in an isolated environment
  – Practitioners can deploy a “cyber incident response” cloud
  – Potential blueprint for official OpenStack adoption

• Future work
  – more fine grained access control within a SIP
  – harden the implementation to prevent overt information flow
Thanks

• Q&A