Secure Information and Resource Sharing in Cloud

Yun Zhang
Institute for Cyber Security
Univ of Texas at San Antonio
San Antonio, TX 78249
Amy.u.Zhang@gmail.com

Ram Krishnan
Dept. of Electrical and
Computer Engineering
Univ of Texas at San Antonio
San Antonio, TX 78249
Ram.Krishnan@utsa.edu

Ravi Sandhu
Institute for Cyber Security
Univ of Texas at San Antonio
San Antonio, TX 78249
Ravi.Sandhu@utsa.edu

ABSTRACT

The significant threats from information security breaches in cyber world is one of the most serious security problems. Organizations are facing growing number of sophisticated cyber-attacks every year. Efficient and secure sharing of attack and security information during cyber incident response plays increasingly significant role in fixing the problems as well as helping organizations recover fast. While traditional systems are slow and inefficient in sharing information and resources securely, cloud platform provides us a considerable convenience to facilitate the sharing. In this paper, we propose access control models for secure information and resource sharing (IARS) in cloud Infrastructure as a Service (IaaS).

Keywords

Formal models; IaaS; OpenStack; Resource sharing

1. INTRODUCTION

It has been a long history since the concept of information sharing came out in the field of information technology. The need to share information between organizations becomes significant for various reasons, such as cyber incident response, business collaboration, etc. The lacking of important attack and threat information sharing may lead to a big security breach. As organizations are moving to cloud, we explore information sharing in cloud platforms.

Traditional techniques for information sharing are lacking in three aspects, as follows.

1. Systems in different organizations are incompatible with each other, which make it impossible for one user in one organization to have access to the system in another organization.

2. Systems are lacking of mechanisms to accommodate temporary users, whereby a temporary user may only be assigned with a internal user account, which incurs a substantial risk of information leakage.

3. Trust issues among organizations remain a big concern.

In the environment of cloud platform, systems of different organization can talk to each other under the same access control architecture. Cloud platform make is possible for users belongs to one organization have the possibility to be assigned to another organization conveniently. Cloud platform also gives the ability to define different level of users with more specific permissions, under which risk is more likely to be controlled.

Our motivation to build a secure information and resource sharing (IARS) model in IaaS came from the case of response to a cyber incident. Consider a community cyber incident response scenario where organizations that provide critical infrastructure to a community (such as a city, county or a state) share information related to a cyber incident in a controlled manner [2]. Sharing information amongst such organizations can greatly improve the resilience of increasingly cyber-dependent communities in case of coordinated cyber attacks [3]. An effective cyber incident response mechanism needs to be built up to provide organizations technical support and services to handle the problems once the cyber incident happens. The most significant part of this mechanism is how to share incident data securely and efficiently. The incident data may not only include the logs of attacks, but also the compromised machine, even the malware. With traditional ways, sharing is constrained in a manual way by means of reporting, which is slow and insufficient. In order to response to cyber attack fast and accurately, we want to share not only the report, but all related information, even a virus or a compromised machine. Cloud platform of IaaS gives us a very good option to place the sharing space, which holds the incident data. Thus, we explored various modeling ways in cloud of IaaS and formalized one of the models.

2. BACKGROUND

OpenStack is an open source cloud platform of IaaS. OpenStack software controls large pools of compute, storage, and networking resources throughout a datacenter [1]. It provides several services, including compute (Nova), identity (Keystone), block storage (Cinder), object storage (Swift), image (Glance), networking (Neutron) and Dashboard (Horizon). Nova provides computing service, like virtual machines. Keystone provides authentication and authorization services. Swift provides object storage services. In [7], the authors present a core OpenStack Access Control (OSAC) model, as shown in figure 1. The OSAC model consists of eight entities: users, groups, projects, domains, roles, services, operations, and tokens. Users represent people who

CODASPY'15, March 2–4, 2015, San Antonio, Texas, USA.
ACM 978-1-4503-3191-3/15/03.
http://dx.doi.org/10.1145/2699026.2699126.

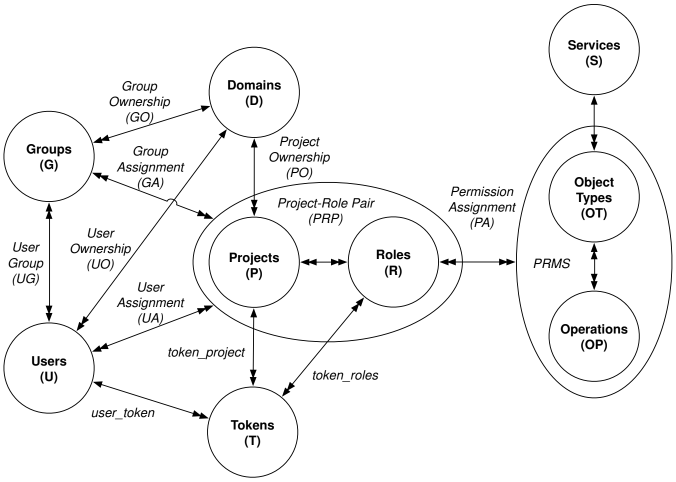

Figure 1: OpenStack Access Control (OSAC) model [7]

are authenticated to access OpenStack cloud resources while a group is a set of users. Projects define a boundary of cloud resources—a resource container in which users can get access to the services the cloud provides, such as virtual machines, storages, networks, and so on. Domain is a higher level concept that equates to a tenant (customer) of the CSP. Roles are global, which are used to specify access levels of users to services in specific projects in a given domain. An object type and operation pair defines actions which can be performed by end users on cloud services and resources. Users authenticate themselves to Keystone and obtain a token which they then use to access different services. The token contains various information for a user to get permission to a project or domain.

3. OSAC-SID MODEL

In the OSAC-SID model, we assume that a user can belong to only one organization, which is consistent with the user and home-domain concept in OpenStack. We extend the OSAC model to include SID and SIP components, as shown in figure 2. For every possible combination of organizations in the cloud, we create a SID to include all SIPs that will be set up among these organizations. For each IARS event, we create a SIP within the appropriate SID.

Similar to the concept of domains which is designed to add one more layer for administration of projects, SID is a administrative concept to manage SIPs. The SID function is transparent to users. SID and SIP components are isolated from the regular domain and projects components. Unlike the concept of domains, there are no users that belong to a SID. A SID exists only for setting up SIPs. However, since a SID is formed and associated with a collection of domains, there are users who will be associated with the SID—but only under the constraint that they are from the collection of domains which are associated with that SID.

A SIP provides a secure isolated space for IARS in the cloud. In other words, SIP is another type of resources container in OpenStack, which is restricted only for IARS among domains and projects. Users who are assigned to a SIP have similar access capability to request all cloud services like users who are assigned to a project.

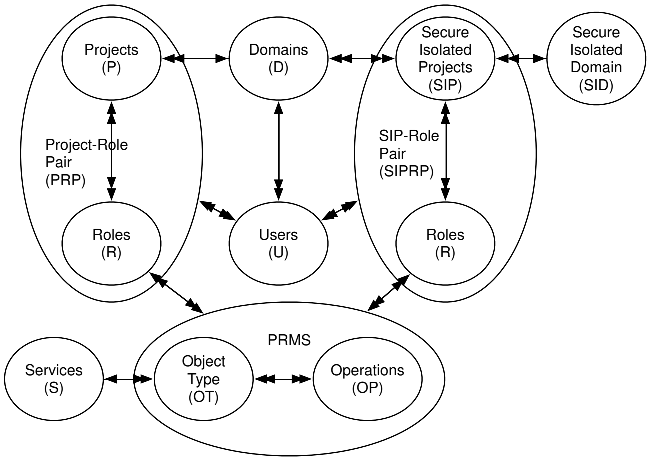

Figure 2: OpenStack Access Control (OSAC) model with SID extension(ignore group and token components)

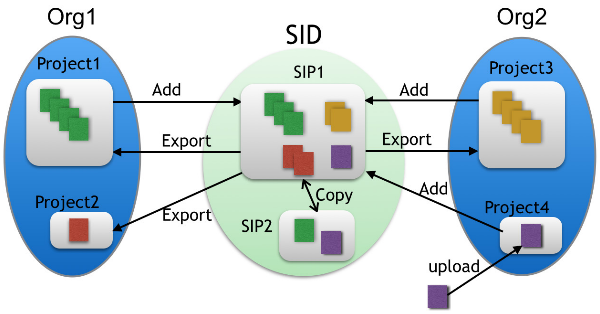

Figure 3: Establish a SID

4. IMPLEMENTATION

In this section, we discuss implementation considerations in OpenStack. Recall that Keystone is the authentication and authorization service in OpenStack. In order to deploy the model in OpenStack platform, we need to modify Keystone entity to include SID and SIP functionality in OpenStack, which facilitates features of IARS.

We assume that each domain represents an organization in OpenStack while projects inside a domain could represent a department or temporary project in the organization. In the case of collaboration, multiple organizations would form a group to create a SID. SIPs will be created inside the SID to facilitate collaboration for different reasons. Objects are exchanged among organizations and SIPs as per SID policy. Figure 3 gives a simple view of how SID is established among two organizations Org1 and Org2.

We implement the model in OpenStack Icehouse release. To establish a SID, we need to modify two parts in OpenStack: policy and Keystone. In policy file, we define that only domain admins are allowed to create a SID and only a SID admin is allowed to create a SIP inside the SID. In Keystone, we add methods to constrain the grants that a SID admin can only add users from his own home domain to the SID while a domain admin is allowed to add any users to his own home domain in OpenStack.

The SID/SIP establishment steps are as follows.

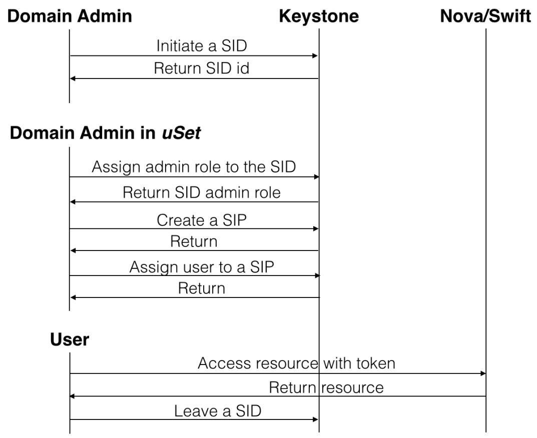

Figure 4: SID/SIP Establishment Process

1. A domain admin initiate a SID creation with parameter of *uSet* which is a set of domain admin users of domains which are willing to collaborate.

2. Domain admins who belong to uSet assign themselves the SID admin role to the SID.

3. SID admins create SIPs and assign users from their home domains to any SIPs inside the SID.

The point here is that uSet defines the set of domain admins who will be SID admins in the collaboration group. However, it doesn't mean that being in the uSet will give the domain admin the admin power over the SID. Instead, in order to complete the process of assigning domain admin to be a SID admin, the domain admin need to agree with the initiation which we implemented by allowing the domain admin to assign himself to the SID admin role. After step two, a domain admin officially becomes a SID admin. Step three allows SID admin to set up SIPs and assign their users to a specific SIP. A user does not have an access to a SIP without being assigned to it, i.e., even if a user has access to the SID, he/she can only access SIPs he/she is assigned to inside the SID. Figure 4 shows the flow to establish a SID/SIP.

After the SID/SIP is established, users can start operation inside SID/SIP as well as between SID/SIP and their home projects. We choose Swift as storage in our enforcement in OpenStack. Nova provides us the ability to create a virtual machine in a SIP. Glance helps to share virtual machine image among domains, which is very useful in case that the collaboration group need to share a virtual machine, like an attacked machine image. Currently, the access control for user accessing the resource inside a SIP is very straight forward. We use the access policy of project for a SIP. Thus, more fine-grained access control over a normal user to a SIP would be our future work.

5. RELATED WORK

There are several differences between our model and other models. First, we proposed our model in a IaaS cloud environment rather than distributed systems. Second, we don't give the collaboration group the direct access over the original data and resources in the organization s done in [6, 5]. Instead, we transfer copies to the collaboration group. Third, we don't use a separate Community Authorization Service(CAS) [5] to manage the access control policies for the collaboration group. Instead, we utilize the setting of roles, users and policies of the cloud to facilitate the access control over the collaboration group. Out approach is more like the model proposed in [4], from where we adapt the group-centric concept and relaize it in a IaaS cloud environment.

6. CONCLUSION AND FUTURE WORK

We developed and implemented a model for IARS. For the future work, we plan to investigate fine-grained access control within a SIP, such as virtual machine access control, networking access control, and so on.

7. ACKNOWLEDGMENTS

The authors would like to acknowledge the support of the LMI Research Institute's Academic Partnership Program.

8. REFERENCES

[1] http://openstack.org.

[2] K. Harrison and G. White. Information sharing requirements and framework needed for community cyber incident detection and response. In *Homeland Security (HST), 2012 IEEE Conference on Technologies for*, pages 463–469, Nov 2012.

[3] Keith Harrison and Gregory B. White. Anonymous and distributed community cyberincident detection. *IEEE Security and Privacy*, 11(5):20–27, 2013.

[4] Ram Krishnan, Ravi Sandhu, Jianwei Niu, and William Winsborough. Towards a framework for group-centric secure collaboration. In *Collaborative Computing: Networking, Applications and Worksharing, 2009. CollaborateCom 2009. 5th International Conference on*, pages 1–10. IEEE, 2009.

[5] Laura Pearlman, Von Welch, Ian Foster, Carl Kesselman, and Steven Tuecke. A community authorization service for group collaboration. In *Policies for Distributed Systems and Networks, 2002. Proceedings. Third International Workshop on*, pages 50–59. IEEE, 2002.

[6] Deborah Shands, Richard Yee, Jay Jacobs, and E John Sebes. Secure virtual enclaves: Supporting coalition use of distributed application technologies. In *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, volume 1, pages 335–350. IEEE, 2000.

[7] Bo Tang and Ravi Sandhu. Extending OpenStack access control with domain trust. In *In Proceedings 8th International Conference on Network and System Security (NSS 2014)*, October 15-17 2014.