OPNFV MOON platform overview

Release draft (51a65a7)

OPNFV

August 12, 2016

CONTENTS

Moon: Toward a Policy-based User-centric Security Management System for Cloud Infrastructure

CHAPTER

ONE

INTRODUCTION

Cloud infrastructure is able to provision a set of different cloud resources/services for cloud service consumers. Trust over the provided cloud resources/services becomes a new challenge. In order to avoid losing control over IT assets that consumers put in the cloud, we design and develop Moon, a policy-based user-centric security management system for cloud infrastructure. Administrator can define policies in Moon, which will be enforced through different mechanisms in the cloud infrastructure.

### Why security management system? A security management system is a combine system that integrates mechanisms of different security aspects. It first has a security policy that specifies users' security requirements. It enforces the security policy through various mechanisms like authorization for access control, firewall for networking, isolation for storage, logging for traceability, etc. ###Why policy-based approach? Cloud computing embeds a various set of heterogeneous resources into resource pools. Such a mechanism makes a management system hard through one standard interface. Alternatively, the policy-based approach bypasses the heterogeneity, the management is achieved through a standard policy instead of a standard interface. Each module of a cloud infrastructure only needs to accept the policy. ###Why user-centric? The flexibility of resource pool for cloud computing makes users custom cloud resources/services for their own purpose. However, the current management system is not able to support this flexibility. A user-centric management system enables users to define, configure and manipulate on the management layer, in order to adapt to their usage and requirements.

CHAPTER

TWO

DRAWBACKS OF OPENSTACK

The first version of Moon is implemented in OpenStack. However, we also plan to make Moon protect other cloud infrastructure like VMware vCloud Director, etc in the future. Considering the current state of OpenStack (release Juno), several drawbacks related to the security management are identified: ###No centralized control The OpenStack platform is divided into different services, like Nova for computing, Swift for storage, Neutron for networking, etc. Each service uses a configuration file with the corresponding security policy. But it lacks a synchronization mechanism between there configuration file. This may bring conflict. ###No dynamic control Currently, each service of OpenStack is managed by a "Policy.JSON" file, all the modification should be done manually and reboot is necessary after the modification. On the other side, the authentication and authorization are achieved through the token mechanism, but there isn't any token revocation mechanism. Once a user gets an authorization token, we will not have any control over the user. It lacks dynamic control at runtime in OpenStack. ###No customization and flexibility Each user of OpenStack consumes their resource pool in their own manner, but it lacks customization for the management system. In OpenStack, user cannot configure their resources and define their own policy for each resource pool (called project in OpenStack). Users may also be a software application, it also lacks an automated policy enforcement mechanism in OpenStack. ###No fine-granularity Finally, the granularity of authorization in OpenStack is not enough fine. Currently, it's at the API-level. This means that we can authorize or deny a user from using an API like launch any VM. But we need the granularity to be pushed to the resource-level, authorize or deny a user from using a specific resource through the API, e.g. allowing a user to launch a dedicated VM.

CHAPTER

THREE

MOON DESCRIPTION

For all the listed reason, we decided to build a security management system to monitor, control and project the OpenStack infrastructure.

###Functional architecture Moon can be considered as a management layer over OpenStack. We can dynamically create security modules in Moon and assign these modules to protect different tenants in OpenStack. ![](../img/moon_infra.png)

###Policy engine The core part of the security management layer is its policy engine. The policy engine should be at same time generic to support a large set of security models used by consumers and robust so that all the manipulations on the policy engine need to be proved correct. For all these purposes, we designed EMTAC (Extensible Multi-tenancy Access Control) meta-model, which defines policy specification, policy administration, inter-policy collaboration and administration over this collaboration. ![](../img/policy_engine.png)

###User-centric At the same time, Moon enables administrators or a third-party application to define, configure and manage its policies. Such a user-centric aspect helps users to define their own manner in using OpenStack's resources.

###Authorization enforcement in OpenStack As the first step, the security policy in Moon is enforced by authorization mechanism in Keystone and Nova and Swift. All the operations in Keystone and Nova and Swift are controlled and validated by Moon. In OpenStack, we implemented 3 hooks respectively for Keystone and Nova and Swift, the hooks will redirect all authorization requests to Moon and return decision from Moon.

###Log System Traceability and accountability are also handled in Moon, all the operations and interactions are logged and can be consulted for any purpose.

###Separation of management layer from OpenStack The separation of management layer from OpenStack makes the management system totally independent from OpenStack. We can install Moon in client's local so that Moon can be locally administrated by clients and remotely project their data which are hosted in Cloud Service Provider's datacenter.

CHAPTER

FOUR

ROADMAP

Even if Moon can now work with OpenStack as a security management system, several blueprints are planned for its improvement.

###Technical improvements

- Update Moon's policy engine with Prolog/Datalog: in the last version of Moon, we use a hard-coded Python policy engine. We will collaborate with VMware and their "Congress" policy engine.
- Networking enforcement: other important improvement is to enable network management by Moon. Based on the defined policy, Moon will configure Neutron, FWaaS, etc, in OpenStack.
- Storage enforcement: storage protect is another important aspect, access to storage blocks or files will be control by Moon.
- IDS/IPS and self-protection: as Moon's security module will protect each tenant,

###Contribution to OpenStack We have worked in the OpenStack community since 1 year and half, our next step is to integrate with the community and contribute Moon to OpenStack/Keystone. Once Moon is integrated in OpenStack, the community, together with its developers will maintain Moon's evolution.

###Contribution to European project "SuperCloud" The H2020 European project "SuperCloud" will start this year, its objective is to provide 360 degree protection on cloud infrastructure. Moon and its policy meta-model will be contributed as core security management system in this project.

###Contribution to OPNFV OPNFV ([OPNFV]) wants to build a reference cloud-based architecture for NFV (Network Function Virtualization) based on OpenStack. Orange will propose a "security management" project in OPNFV. Moon is supposed to be the base module to develop and manage dedicated security mechanisms for vNF (virtualized Network Functions).

See: [OPNFV-MOON]

[OPNFV]:http://www.opnfv.org [OPNFV-MOON]:https://wiki.opnfv.org/moon

Revision:

Build date: August 12, 2016