MAY 2017

M.Sc. in Electrical and Electronics Engineering

NASSER NAJIM ABDULLAH

UNIVERSITY OF GAZİANTEP

ELECTRICAL AND ELECTRONIC ENGINEERING

IDENTIFY CLOUD SECURITY WEAKNESS RELATED TO AUTHENTICATION AND IDENTITY MANAGEMENT (IAM) USING OPENSTACK KEYSTONE MODEL

M.Sc. THESIS
IN
ELECTRICAL AND ELECTRONIC ENGINEERING

BY
NASSER NAJIM ABDULLAH
MAY 2017

Identify Cloud Security Weakness related to Authentication and Identity Management (IAM) using OpenStack keystone Model

M.Sc. Thesis

in

Electrical and Electronic Engineering

University of Gaziantep

Supervisor
Prof. Dr. Ergun ERÇELEBİ

by
NASSER NAJIM ABDULLAH
MAY 2017

REPUBLIC OF TURKEY
UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES
ELECTRICAL AND ELECTRONIC ENGEERING DEPARTMENT

Name of the thesis: Identify Cloud Security Weakness Related to Authentication and Identity Management (Iam) Using Openstack Keystone model

Name of the student: Nasr Najm ABDALLA

Exam date: May 18, 2017

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. A. Necmeddin YAZICI
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ergun ERÇELEBİ
Head of Department

This is to certify that we have read this thesis and that in our opinion; it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Ergun ERÇELEBİ
Supervisor

Examining Committee Members	Signature
Prof. Dr. Ergun ERÇELEBİ	.....
Prof. Dr. Ulus ÇEVİK	.....
Assist. Prof. Dr. Ahmet Mete VURAL	.....

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Nasr Najm ABDALLA

ABSTRACT

IDENTIFY CLOUD SECURITY WEAKNESS RELATED TO AUTHENTICATION AND IDENTITY MANAGEMENT (IAM) USING OPENSTACK KEYSTONE MODEL

ABDULLAH, NASSER NAJIM

M.Sc. in Electrical and Electronics Engineering

Supervisor: Prof. Dr. Ergun ERÇELEBİ

May 2017
78 Pages

The cloud service is one of the leading and modern technologies in the world and it has been used widely in many governments, organizations and companies, by delivering computer services from multiple data centers in different regions to the users around the world. It uses the internet as a transfer media to protect the client by increasing the security techniques which identifies and authenticate users, and it encrypts information during data transformation. The issue of putting data away from users' hands has been well addressed in this study, developing a framework to improve security of OpenStack environment which increases the recognition of identify to ensure data protection. This study has focused on identifying security vulnerabilities related to Identity and Authentication Management (IAM), a different technique has been implemented to correct the problem of security and identity. The popularity of cloud computing system may increase if we apply these corrections, which will the effect the number of cloud clients. We have demonstrated the security of clouding and clarified security problems and threats of cloud. Research depends on present literature and personal experience which is gathered by analyzing different platforms of the cloud and its applications.

Key Words: OpenStack, security, authentications.

ÖZET

OPENSTACK KİLİT TAŞI MODELİ KULLANARAK KİMLİK DOĞRULAMA YÖNETİMİ İLE İLGİLİ BULUT SİSTEMİ GÜVENLİK ZAYIFLIKLARININ BELİRLENMESİ

ABDULLAH, NASSER NAJIM

Yüksek Lisans Tezi, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Ergun ERÇELEBİ

Mayıs 2017

78 Sayfa

Bulut hizmeti, dünyanın önde gelen ve modern teknolojilerinden biridir ve çoklu merkezlerden dünya üzerinde farklı bölgelerdeki kullanıcılara bilgisayar hizmetleri sunarak birçok ülkede, kuruluşta ve şirkette yaygın bir şekilde kullanılmaktadır. Bu hizmet; interneti güvenlik tekniklerini arttırma yoluyla müşterileri korumak için bir aktarım ortamı olarak kullanmaktadır ve bu teknikler, kullanıcı tanımlama, kimlik doğrulaması yapma ve veri dönüşümü sırasında bilgileri şifreleme gibi işlemlerdir. Bu çalışmada, verilerin kullanıcılardan uzakta depolanması konusu, OpenStack ortamının güvenliğini geliştirmek için bir çerçeve geliştirme yoluyla iyi bir şekilde ele alınmıştır ve bu çerçeve verinin korumasını sağlamak için tanımlamayı arttırmaktadır. Bu çalışma, Kimlik ve Kimlik Doğrulama Yönetimi (IAM) ile ilgili güvenlik açıklarını tanımlamaya odaklanmıştır ve bu çalışmada güvenlik ve kimlik sorununu düzeltmek için farklı bir teknik uygulanmıştır. Bulut istemcilerinin sayısını etkileyecek bu düzeltmeleri uygularsak, bulut sisteminin popülaritesi artabilir. Biz bu çalışmada bulutun güvenliğini gösterdik, ve güvenlik sorunları ve bulut tehditlerini açıklığa kavuşturduk. Araştırma, bulutun farklı platformlarını ve uygulamalarını analiz ederek toplanan kişisel deneyime ve güncel literatüre dayanmaktadır.

Anahtar Kelimeler: OpenStack, güvenlik, kimlik doğrulama.

DEDICATION

To a father who did a lot thing to me, and passed away before I could return any of his favors

To a mother who does her best for her sons

To a wife who supports me all the time, even when she needs support

To my children whom I wish to be better than me

ACKNOWLEDGMENTS

The study was not possible without the guidance of a lot of people. Many gratitude to **Prof. Dr. Ergun ERÇELEBİ**, who help me and tolerated me well. Also many thanks to all of my professors at Gaziantep University Graduate College. Many thanks to my fellow workers and friends who always give support and love to help me reach my goal.

TABLE OF CONTENT

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

AIK	Attestation Identity Key
API	Application program interface
CBE	Cloud based E-Learning
CBET	Cloud based E-Learning technology
CCM	Constant comparative method
CRUD	Create, read, update and delete
CTRM	Core Root of Trust Management
DDoS Attack	Distributed Denial of service attack Endorsement Key
GRUB	Grand United Bootloader
GVMI	Generic virtual machine image
IaaS	Infrastructure as a Service
IAM	Identity and access management
IEEE	Institute of Electrical and Electronics Engineers
IMA	Integrity measurement architecture
ISO	International standard organization
ITIL	Information technology infrastructure library
LCA	Local certificate authority
LMS	Learning management system
PaaS	Platform as a service
Park	Private key
PCR	Platform Configuration Register
PDA	Personal digital assistant
PK	Public key
REST	Representational state transfer
RNG	Random number generator
SaaS	Software as a service

SCORM	Sharable content object reference model
SLA	Service level agreement
SRK	Storage root key
SSO	Single sign on
TCB	Rusted computing base
TCG	Trusted computing group
TPM	Trusted platform module
TSS	TCG software stack
TTP	Trusted third party
VM	Virtual machine
VMI	Virtual machine image

CHAPTER 1

INTRODUCTION

1.1. BACKGROUND

Cloud system is one of most technology that gives computers a service and application remotely and can reduce cost to make the live easier. Many researches start going to contributed to facilitate and support the new technology. Our research here start to take the security part of one of the biggest open source cloud system that called OpenStack. Trying to find a solution to the part of security is one of important factor in cloud distribution so we take keystone component and add many new technologies to harden the system and prevent many security gaps.

1.2. Motivations

The aim of this study was to increase the security in OpenStack platform by adopting cloud technologies, various contribution and organization has worked to put guidelines and recommendations to secure transition to clouding. Aside from protecting the data and the identity, we support one of the open sources platform which facilitate to the user adoption of a platform, these platforms can be used widely and increase the confidence of the user in the system of any cloud service provider. Differences in technologies becomes an issue when adopting or developing a new system because different technologies is hard to control virtualization and manage storage, at the same time the user will have more experience in one well defined system and will have less knowledge when change from provider to another that have the same system. The high support, interesting and adoption of OpenStack let a lot of companies develop and produce new product that supports open stack like Cisco, HP and IBM and even google

Further information will be discussed in the (OpenStack security) section to explain the value, adoption, and managing OpenStack as an open source platform and to increase the number of users which can benefit from this platform. In this study, we

have found the following approaches which the study was made to solve problems which also face the customer of cloud computing:

1- Secure the customer's data during the communication with OpenStack.
2- Prevent the misuse of the identity information and privileges of the user.
3- Increase the security of the transferred data and decrease the exposure to the hacker or unwanted crossing.
4- Increase the number of users that adopt the OpenStack.
5- Using an advanced technology to secure the system and to make it more reliable.
6- The popularity of OpenStack can compete with the biggest systems in the world like amazon and google.
7- Decrease the number of the datacenters that can affect the general cost significantly for company or organization.
8- Add some recommendation to have a standardization of the security part of OpenStack platform.

1.3. Contribution

The tangible value that we added during our study in the thesis will specify the main outcome that gives a lot of benefits to the security of cloud system as the following points:

1- During the study of a different type of authentication, we find a lot of recommendation that needs to be added to OpenStack security so OpenStack become more protected and harder to any attempt of hacking or misuse of identity or privilege.
2- We analysis the main gap that faces the security of cloud system of public and private cloud we find a list of main security issue that estimates the security issue in the cloud technology.
3- During the study, we found the main core point that needs further studying, and we made an analysis to the data management and the identity and who can have access and higher authentication and control.
4- In the study of the access management and the cloud identity we discover the below point:

A- The system still have the same mechanism to log the user (user name and password and to log to such a sensitive system we need to harden the authentication phase with other technology like federation and RSA.

B- The ability of the administrator to Read/Delete the information of the user that can breach the sensitivity information of the user.

C- Lower administrator can get a higher privilege that may affect the permission of the system.

D- A recommendation to encryption most of the data during the upload will be more difficult to be a breach.

E- In this study, we discussed about different authentications mechanisms and identity as the OpenStack is open standard as most of the current authentication have a weak password without no clear policy to protect the system.

F- We need to add more option in keystone service to be more adaptable for federation and can be connected with multiple IDP (identity provider).

1.4. Thesis problems:

In this thesis, we wanted to find a solution to many problems which face keystone as the identity and authentication service of OpenStack platform, our goal was to find the best combination of authentication and encryption for the system, which still needs a lot of improvements, like clear text passwords and weak encryption, by using a new way to check the identity of the user and check privileges, if we didn't consider these points, the system might get breached and compromise the privacy of users, we discussed varies protocol and techniques that can solve these problems. Because the cloud technology is still new we tried to adopt a newer open source system to protect the user information and to ensure the confidentiality of the data transfer inside and outside the system, which helps reducing the high cost of hosting.

1.5. The Clouding

It states the idea of the programs system and with services made available to users, and the software and hardware responsible for providing these services and applications. It has recently drawn attention worldwide. Cloud computing can be

defined in many ways. Gartner defines it as a style of computing where massively scalable IT-enabled capabilities are delivered 'as a service' to external customers using Internet technologies [18]. And NIST (National Institute of Standards and Technology) defines Clouding as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [13]. Even though clouding has a lot of definitions, Gartner and NIST both agree that cloud paradigm focuses on providing every network-accessible computing resource as-a-Service (XaaS).

Cloud computing consists of integration of technologies the old and the new which was available during the early 1990s (grid computing, utility computing and virtualization). Each one of these technologies is employed to form a component in the layers of cloud computing, which would leave the client to compensate only for the resources used rather than paying a set cost. This revolutionary service made various companies and establishments which need additional hard-ware to purchase additional computing power without the need to pay a big amount of money for their proprietary IT-infrastructure. Virtualization technology is a major factor that improved cloud computing, which in turn leaves the resource consuming to be dynamically sized (decreased or increased depending on the requirement).

Despite that cloud computing adds many benefits and minimizes costs, arrangements should have a clear mechanism for data migration. Cloud computing should be approached carefully and the data sensitivity should be considered as well. If organizations has adopted the cloud model, they no longer can control their data, which other classical models provide. By doing thus, they entrust the service provider’s security policy. Protection and privacy problems should be treated at an early phase, because later changes could take many complicated subjects, which can be pricey and speculative.

1.5.1. Definitions of Cloud Computing

It is the feature that gives computational resources through a network. Usually in computing, users hold all software and data to accomplish computing processes on any project. But when we use cloud computing, users only need to have an operating

system with an internet browser and high internet connection speed in order to make computing on files. A lot of companies like VMware, Cisco, IBM, Xen and others are all investing in virtualization technology. Make all their investments not just have a better and easier access to the customer applications, but also focus on the new generation of clouding.

1.5.2. Brief Historical Background of Clouding and Foundation Characteristics

A group of organizations in 2009 including Google, Intel and IBM, developed a cloud open platform to put the first practical use for clouding in Open Cloud Manifesto defining the main clouding characteristics and schemes.

Cloud computing is a collective term given to applications delivered as a service across the internet and the server hardware tasked with their provision. [2]. It is a theoretical performance of a long-held desire to offer a service which is a distributed and highly scalable computing environment, the Cloud's brief history can be traced back to 1999, when the SETI project used an early model of distributed computing. "SETI@home" took a divide and appropriate approach to complex mathematical problems. Individuals would connect to a server that would distribute small components of a larger task and each client would crunch numbers toward a single goal. [22].

In 2006 the first commercial web-based compute and storage facilities were offered by Amazon. [1]. Amazon Web Services (AWS) is renowned for its Elastic Cloud Computation (E2C) which allows scalable computation capacity on demand [6]. Furthermore, services such as the Simple Storage Service (S3) gave access to a scalable, reliable and inexpensive data storage infrastructure. [6]. From a business perspective, the cloud abrogates the need for large capital investment in physical hardware. Instead, adopting a variable model that allows resources to be scaled back if over-provisioned or extended in situations of unexpected demand. [2]. The act of delocalization opens doors to multiple, unlimited venues from elastic computing to on demand provisioning to dynamic storage and computing requirement fulfillment. [3]. So this technology made data sharing between many companies much easier and cheaper for development regarding the risk averse nature of modern business.

The headline characteristics are:

- Ease of scaling and providing a dynamic power with a low and efficient cost,

- Permits users get the best of cloud power without the complexity of new technology,
- It can be hosted inside an organization (Private cloud) or in the Internet (Public Cloud).

The importance of cloud manifesto can be summarized

- **Easy Scalability**: Big organizations always need to keep up to date and make necessary changes to keep up with market demands. One of the most appealing cloud facilities is the ability to solve the issue of scalability. At the time and during the work with computing resources if needs surpass higher or lower than normal sizes, cloud can handle the changes.

- **Streamlining the Data Center**: when an organization invests in a data center of any size, there is a need to buy and maintain the software and hardware elements. When providing the services, the hardware is homed, and there becomes a necessity for employment of IT support. Any organization simplifies its infrastructure by benefiting from clouding internally or by reducing the work high load into the externally clouding

- **Improving business processes:** the cloud can provide a stable infrastructure to develop the business processes by providing the ability to share the data and applications. This helps all members to focus on business processes rather than the structure that hosts it.
- **Minimizing startup costs**: The Cloud startup costs are relatively low. This provides an advantage for beginner organizations and companies in emergent marketplaces, or being use for high technology clusters in big organizations because the organization will start with a ready infrastructure, which will help save time and other resources that will need time and cost on infrastructure of data centers that reduced by using clouding service.
- **Dynamic computing infrastructure**: dynamic simply means scalable, standardized, virtualized and secure hard-ware with high redundancy to provide easy availability.

- **IT service-centric approach**: on the contrary, to a server-centric model, this model can ensure availability, easy access, dedicated instance of an application or service.
- **Self-service-based usage model**: the capability to upload, build, deploy, schedule, manage, and report on provided business services on demand.

- **Minimally or self-managed platform**: it is done by using a software which can manage and automate implementing the following:
  - A provisioning engine for installing services and removing them, recovering resources for efficient reuse,
  - Mechanisms for scheduling and reserving resource capacity,
  - The capability for configuring, managing, and reporting to ensure that resources can be allocated and reallocated to multiple groups of users,
  - Tools for controlling access to resources, and policies for how resources can be used or operations can be performed.

- **Consumption-based billing**: Clouding service charges clients depending on how much recourses they use, business and technology over the years created this approach.

1.6. Cloud Computing Characteristic.

The cloud model contains five primary characteristics, with three are service models and four deployment models. To draw a full definition of cloud, we can sustain the primary characteristics, the five primary features are discussed separately:

1.6.1. On-demand capabilities

Self-service is that it allows clouding resources to be applied as needed without the user having to contact the service supplier. With this service, a user can calendar the procedure of clouding services such as storage and computation whenever he needs, in accession to the deployment and management of these services. The self-service interface is certainly user-friendly so as to be acceptable and effective and provide effective. This means users can alone acquire and use computing abilities, like network resources and server time without an external help.

1.6.2. Broad Network Access

A good internet connection with high-speed links must be available to the user to join to cloud computing and for the process to be efficient replacement to internal data centers. There are several examples of groups that procedure a three-tier model to join a computer, mobile phones, printers, and handheld PCs with (wide area network) WAN. This three level architecture is comprised of follow fundamentals:

- Access switches which linked computer devices to aggregation switches
- Aggregation switches which control flows
- Switches and Core routers, which offer connection to the WAN and traffic management

Nevertheless, during cloud computing using there can be problematic delays as this three-level design brings delay in times of 50μs or more. For optimum performance, the latency time of the switching environment should be 10 microseconds or less. There is the available two-tier approach which would eliminate the aggregation layer, and thus can meet this requirement, by deploying 10Gbit/Sec Ethernet switches or the arriving 100Gbit Ethernet switches.

1.6.3. Location-Independence and Resource Pooling

A big and elastic resource pool is needed to meet the consumer's cloud computing needs, meet service level requirements, and provide economy of scale. Resources are required for the execution of applications, and the resources assigned resourcefully for optimal performance. Service providers actually locate the resources at multi geographic locations and assign virtual computation where required.

1.6.4. Rapid Elasticity

The cloud has the ability to reduce or expand designed resources efficiently with no time to connect the required characteristic of the self-service. This is referred to as

rapid elasticity. This allocation may be automatic and appear as a large pool of dynamic resources to the user which he can pay for when he needs.

1.6.5. Measured Service

The cloud computing service is measured in the understanding that the amount of cloud resources that a customer uses can be monitored and allocated automatically and dynamically. The service provider can then bill the consumer on the basis of a measurable consumption of the cloud resources which were allotted for the specific session only.

1.7. Service Delivery Models of cloud computing:

It characterizes the dissimilar of delivery models that work with deployment model. So that delivery models are the IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service).
We have three kinds of service models in clouding as below:

1.7.1. Cloud Software as a Service

The primary aim of this service is to make the customer the power to use the provider's applications which are passing on a cloud resource. Applications are available on many platforms and clients (web e-mail). But clients are not earmarked to have control over the basic resources of the provider's cloud (cloud network, servers, controlling systems, computer memory, applications, settings, and so on).

1.7.2. Cloud Platform As A Service (PaaS):

Developers can make use of some of PaaS features. PaaS is a way to charge for use of hardware, on which user capable of develop and deploy applications which are developed in programming languages and the tools which can run on the cloud
Clients are not empowered to control the basic resources, but can control the applications and settings used in the application environment. The main aim of this service is to lower the price and the difficulty of paying for, hosting and managing hardware and software for making the client’s application run. Usually in the cloud service development environment is dedicated and specified by the cloud provider

and is in accordance with the platform. Both the client who uses the cloud and the provider of the cloud share security responsibilities.

1.7.3. Cloud Infrastructure As A Service (IaaS)

IaaS is very popular model in the cloud services. IaaS gives clients a way to use an operating system or any application and software. The customer can select which operating system they want to apply, but still they don't have control over the basic infrastructure and they can't handle it too, but they may control the operating system, applications, and sometimes they can pick a basic firewall. The primary destination of the overhaul is to replace the purchase, maintenance and hosting of hardware and software infrastructure. The underlying infrastructure of the cloud cannot be managed or controlled by the consumer, but the same consumer does have access to control the OS, Programs, and storing data, addition to having capability to control selected networking components (for example host firewalls).

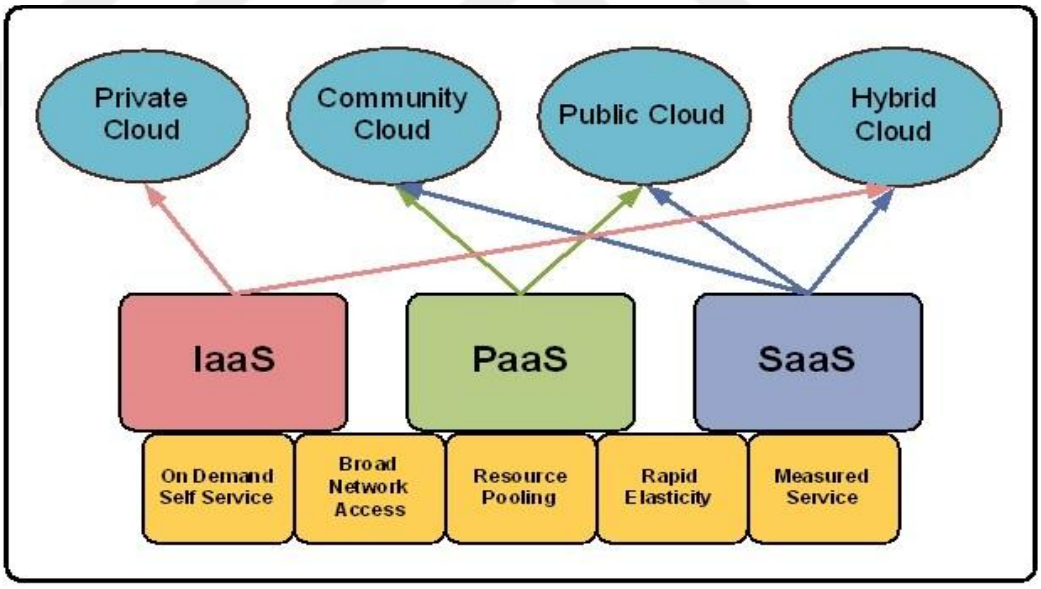

Figure 1.1 Cloud Service and deployment models

1.8. Deployment Models of Clouding

It can either be internal or external implementation and is sorted in the NIST demonstration as below:

1.8.1. Private cloud

providers design this service exclusively for a certain company or organization, which usually means more users (business units). Data centers can host private clouds and they can host at any local housing or business place. Cloud services can be offered by the society itself or a third party. Only the company and designated stakeholders can get the access to function along a particular private cloud, but companies in turn give birth to continue with buying and keeping up all software and infrastructure, therefore cost reducing has been made less effective.

1.8.2. Community cloud

specific users from presidencies have mutual worries (e.g., human right, special technology, and compliance thoughts) provisions the clouding for special use.

1.8.3. Public cloud

the public provisions the clouding for open usage. It could own by group or individual achieved and functioned by a commercial, educational organization, or some grouping of part of it. And this model made services and even facilities available over the internet, it can be possessed, functioned and managed by academic, government or business organizations, and maybe even all of them.

1.8.4. Hybrid cloud

it consists of two or more separated cloud infrastructures (private/public or virtual/actual. The model keeps discrete entities working and these entities are connected together classic techniques such as standardized APIs, and in its turn, delivers easier communication division of tasks in several models of clouds.

1.9. Clouding Drivers

It is a rapidly growing phenomenon in information technology security space business as cloud architectures are being produced widely. The main leading thought cloud suppliers provide in market sector are, Google, Microsoft, VMware, Eucalyptus, Oracle, Salesforce, Citrix, and Rackspace. There are several other vendors which can provide various Cloud services. The cloud providers have different forms in which they provide their services:

- Amazon: Amazon Web Services, including (EC2), (S3), etc.
- Google: Google App Engine - It supports application-programming interfaces for the data store, photo manipulation, Google accounts and e-mail services.
- Microsoft: Windows (Azure Platform): it is a collection of Clouding services that provides a particular fixed service to cloud developers.
- Eucalyptus - open source structure to build a private cloud architecture on existing company.
- IBM: Lotus Live (PaaS).
- Salesforce: (SaaS).
- Rackspace Cloud: (formerly Mosso).
- VMware: Deliver Virtualization infrastructure.

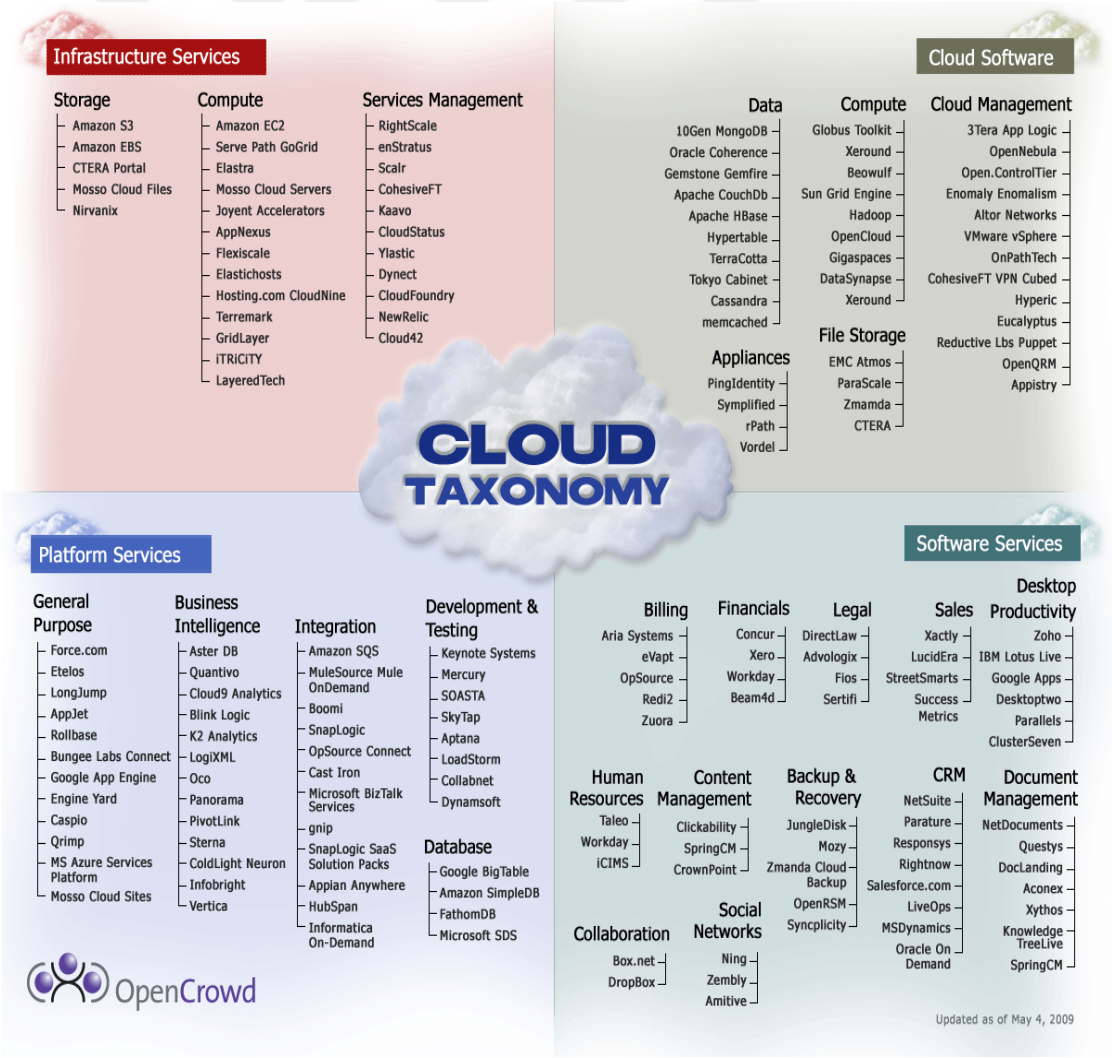

Figure 1.2 Open Cloud Taxonomy

CHA-PTER 2

SECURITY OF CLOUDING

2.1. Background

Cloud computing creates a new technology in data hosting service by a third party it is a good economic model, which raises security concerns.

Elasticity, simplicity, and price model are the only things which separates the service providers from the hosting providers in the cloud.

Nowadays a lot of organizations have been built mainly to provide cloud's security. One of these organizations is (Cloud Security Alliance) CSA at the front, which, in the end of 2009, has published a document entitled "Security Guidance for Critical Areas and Cloud Computing". CSA is a non-profit organization created in late 2008. The creating members are mainly industry representatives and many corporate members such as Google, Microsoft, IBM, and VMware.

Jim Reva founder and executive director of the Cloud Security Alliance stated that enterprises across sectors are eager to adopt cloud computing - but that security standards are needed both to accelerate cloud adoption on a wide scale and to respond to regulatory drivers [5]. A lot of organizations would need to test the security and privacy concerns by using cloud computing as a service for their business crucial important applications. But, the security of business data in the cloud is hard to obtain or sometimes even impossible. SaaS, PaaS and IaaS all expose their own security concerns [21]. The following are famous relevant threats in the cloud which have been recently discovered:

- Threat 1 misuse in cloud computing: IaaS presents to the companies massive computing, storage, and network. The client can use various services right after finishing subscription and paying for providers. Some providers grant free limited trial periods. Some clients have been able to host malware, Trojan horses and software exploits by misusing relative anonymity from the free registration.

- Threat 2 Insecure Interfaces and APIs: Cloud providers give a lot of a APIs that control and act together in clouding. Such involve include: provisioning, monitoring, billing, resource management. The security and availability of cloud depend on the security of these fundamental APIs. Access control and authentication should be built to prevent bad intentional and malicious attempts.
- Threat 3 Malicious Insiders: Most organizations are familiar with the threat of a malicious insider. The provider is not allowed to reveal how he grants employees access or how he monitors the employees [5].
- Threat 4 Shared Technology problems: IaaS providers use shared infrastructure to make their service scalable, even though the components like CPU, GPU etc. are not designed to offer multi-tenant architecture and powerful isolation mechanisms. In order to prevent unauthorized or malicious activities, the provider must monitor their cloud.
- Threat 5 Data Loss or Leakage: accidental deletion of data, unlinking data from a bigger context and losing a key with no backup are famous examples. The threat of data exposure (e.g. unencrypted keys, insecure APIs, shared environments, etc.) increases in the cloud, because of the shared hard-ware architecture which the cloud depends on.
- Threat 6 Account or Service Hijacking: this kind of threats is relatively old because attacking methods like phishing, exploitation and fraud are familiar issues. The cloud is what separates as a new dimension to this threat. Any attacker who steals the user access credentials could easily corrupt data, edit them, and redirect visitors or users to suspicious websites. Victim's account may become a new base for the attacker, since he could launch further attacks from it. [5].
- Threat 7 Unknown Security Profile: Cloud computing is a modern technology which is also for a lot of organizations is a new service for business, which brings with it a difficulty in understanding its pros and cons. We can make a lot of advertisements to promote cloud services, but there is still the question of how we store the client data. An organization should be concerned about patching, auditing and logging their services. Most such

questions are not Cleary answered and leave most of the customers with an unknown risk to their profile which could lead to serious threats. [5].

Moreover, privacy and security risks create an obstacle for clients to adopt this new form of IT. According to IDC (International Data Cooperation) survey in August 2008, security was regarded as the main challenge in cloud computing [4], [14]. Security is one of the major concerns, says consumers of cloud computing who fear that their business information and sensitive IT resources in the Cloud are not immune to attacks. Most cloud security problems arise because of lack of control, lack of trust methods, multitenancy etc. Such problems can be found in third party management models and also self-managed cloud platforms. It is very difficult to deploy security models in cloud computing because of the different sorts of attacks in both the application side and in the hardware components. Actually, many attacks with catastrophic effects only need one security flaw.

2.2. Importance of Security in Cloud

When it comes to privacy, its concept can be different depending on many factors such as the country, culture and jurisdiction. Privacy rights and obligations link to the collection, use, disclosure, storage, and destruction of personal data. But in the end the main idea about privacy lies in how organizations consider the user's data, as well as the transparency to an organization’s practice around personal data. In this study, a research has been made on the security features of cloud platforms using OpenStack as a case study.

Clouding provider and consumers both have different security responsibilities. For instance, Amazon’s AWS EC2 infrastructure as a service offer does include any responsibility regarding security up to the hypervisor, which means security controls can only be addressed as physical security, virtualization security, and environmental security. The information system structure (example) including the OS, Program, and security the responsible for the consumer. As mentioned previously, cost efficiency is one of the attractions of cloud computing.

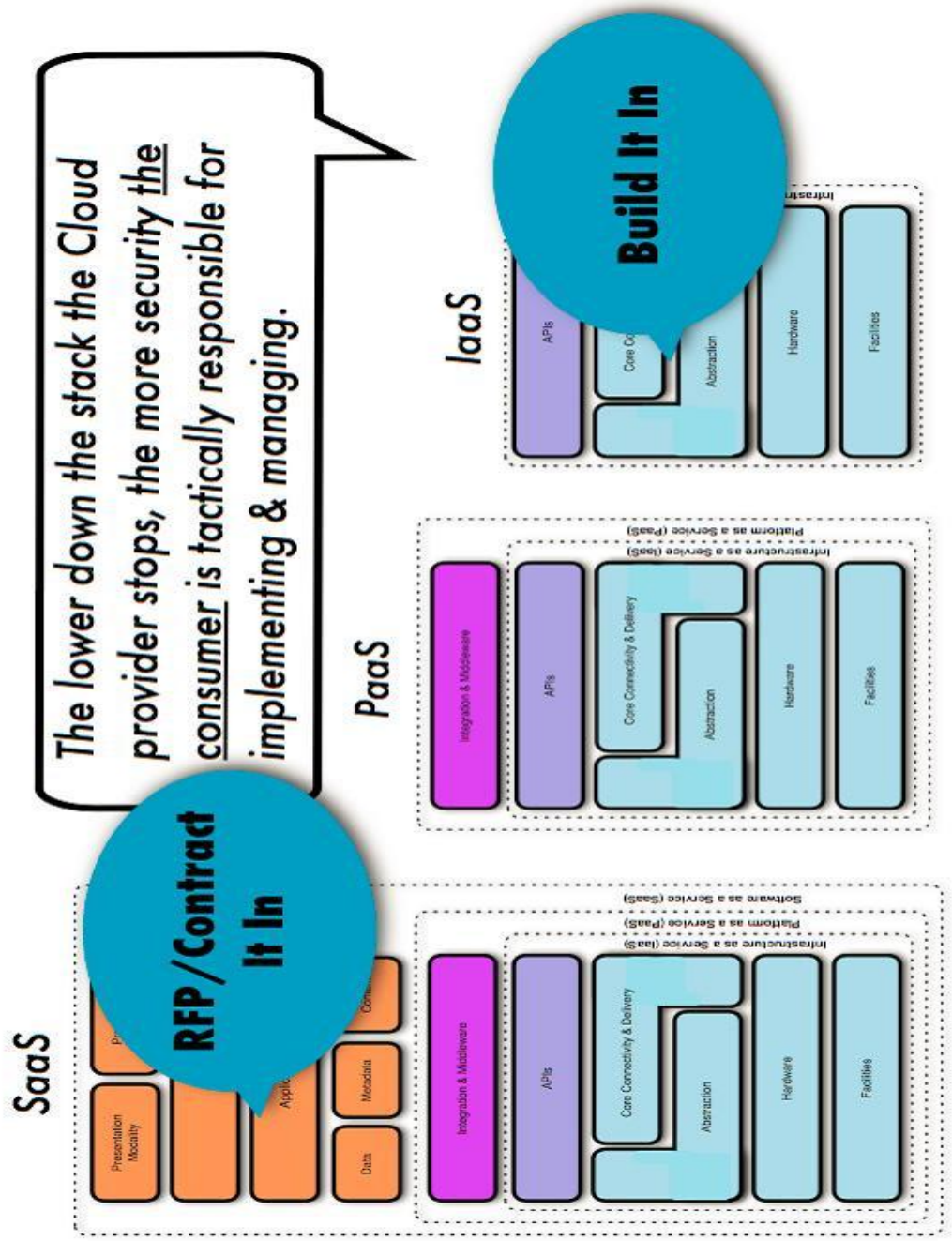

Figure 2.1 Illustration of Security integration

Figure 2.1 illustrates the integration of security in SaaS environments. The cloud providers negotiate security controls in their agreements for service, privacy, service levels, and compliance. The pass is regarded to offer somewhat of a balance in between, where the security of the platform is the provider's responsibility, while the

consumer program security development against the platform and are the consumer responsibility.

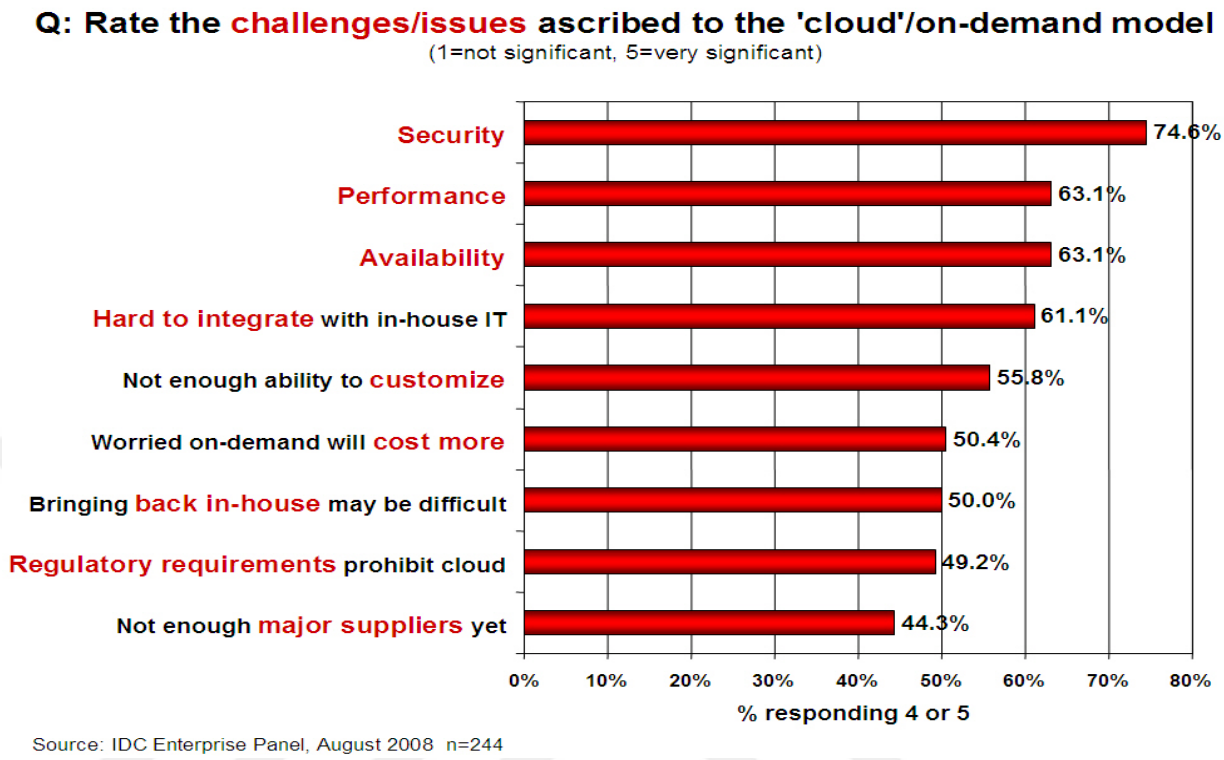

Figure 2.2 The value of Security in Clouding

The Figure 2.2 signifies the result of a survey Carried out by the IDC (International Data Corporation) in August 2008, Cloud Computing for Business and Society, Brad Smith, a member of the Microsoft General Counsel presented results of a survey done by Microsoft for calculating attitudes towards Cloud Computing between business managers and also taking into account the general public. The survey, which was conducted in the first month of 2010, found that while 58% of the general public and 86% of the senior business leaders were very much excited about the capabilities of Cloud Computing, 90% of these same people had great concerns regarding security. The privacy and access of their own data in the Cloud were the main concerns.

The act of breaking into a computer system is called hacktivism, and hacktivists (the people who perform hacktivism) have been known to target such systems as the cloud for social and political purposes. Hacktivists considered the cloud a valuable target because cloud providers have a great number of tenants, many of them are

influential figures in society, whether they be political figures, military, scientists or even celebrities. Getting hold of the information So they think stealing these information will make them rich, which makes hacktivists motivated because they can blackmail tenants of publishing these information, which wins them a lot of money. Accessing the cloud services is also possible for the public as well as from within by signing up for the services. Lim, Coolidge and Hourani believe that cloud suppliers are notorious for not having sound security, but rather their focus is to leverage the economy of scale to provide capabilities in the most cost-efficient ways. LulzSec, a notorious hacktivist group that registered for services on *Cloud Flare* data distribution network, demonstrates one great example of hacktivist activities and aims. LulzSec hacked Sony PlayStation Network and Sony in general, and then went on to publish its Sony hacking work on their website from the back of Cloud Flare. It forwarded information about personal profiles of a million user accounts hacked from the Sony website to the Internet. This made other hacker communities to respond and to conduct A DDoS attack on the Lulz secure website, and that led to bringing the website down, but it was restored after 45 minutes because of using Cloud Flare. The importance of this example is to emphasize that cloud services are used heavily by hacktivists, along with social networking sites.

Finally, the cloud can be and in fact is used by hacktivists as a platform from which to operate by distributing code, announcing their exploits, meeting and exchanging ideas, amongst certain other activities. These kinds of hacktivists rally and mobilize within the cloud. Cloud providers are in unenviable territory when this happens, as was the case with Cloud Flare, as they cannot deny the user's legitimate use of the cloud, but at the same time, cannot efficiently monitor hacktivist activities and prevent them from happening without violating their users' privacy. They provide the base ground for such events as the Sony Hack. The 2011 Data Breach Investigations Report (DBIR) identifies the sources of the acts of internet breaches as shown in Figure 2.3

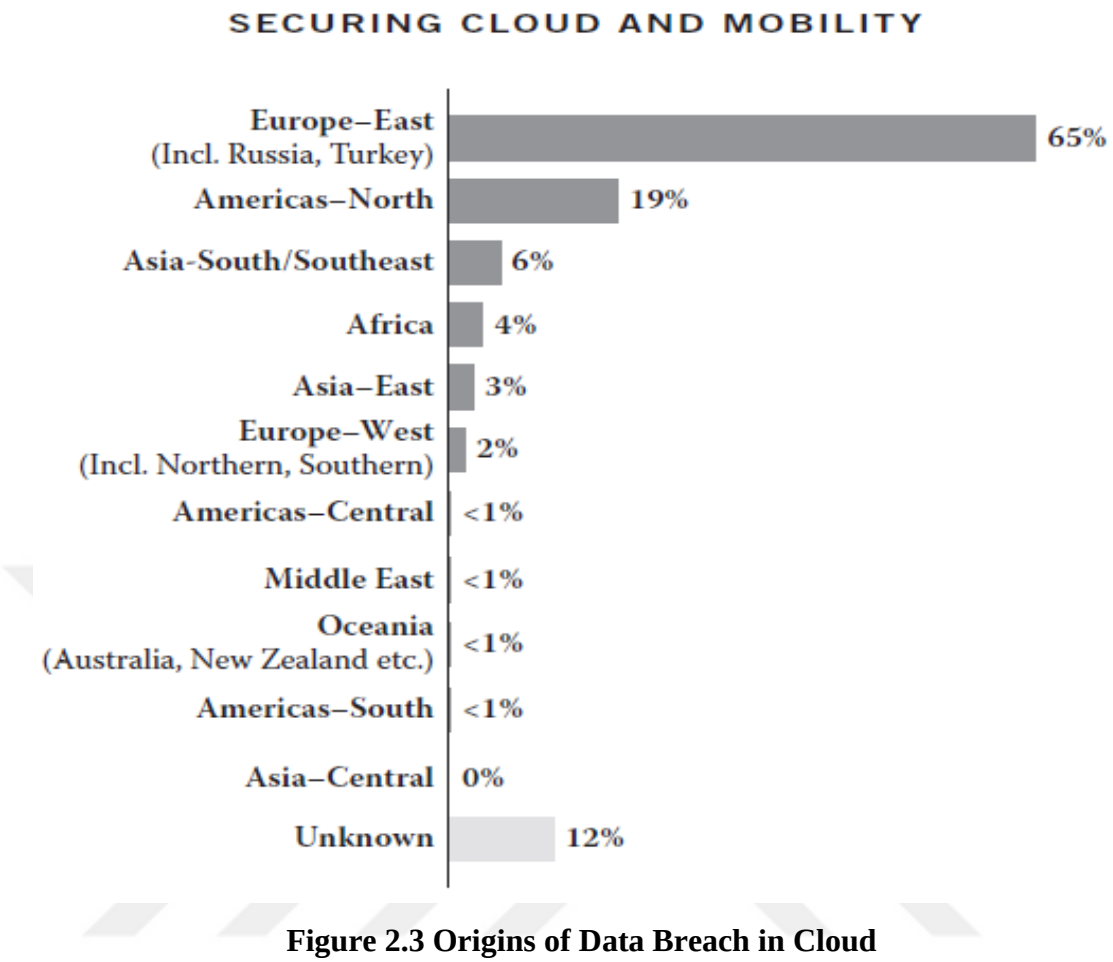

Figure 2.3 Origins of Data Breach in Cloud

Perhaps unsurprisingly, the number Figure 2.3 reveals that the majority of the IP addresses of the cyber-attacks are sourced from Eastern Europe, particularly Turkey and Russia. These statistics seemingly indicate that this geographical region contains both the experienced people needed and the infrastructure for the launch of advanced cyber-attacks as an ongoing business operation. In Russia, major crime families have a deep investment in cybercrime. The matter is not one of hacktivists causing domestic mayhem. These supported criminals, operate services for spamming, phishing, botnet command and control networks, and a plethora of other cyber-criminal activities. Families usually involved in Russian organized crime were recruited by ex-employees of the Russian Federal Agency for Government Communications and Information (FAPSI) when the agency disbanded in 2003. There are strong signs that tell us that these Eastern European cyber-criminals are not just highly-skilled techies, but also have enterprise-grade infrastructure. Their

knowledge exceeds the basic as they have deep knowledge, and of course important contacts within their governments.

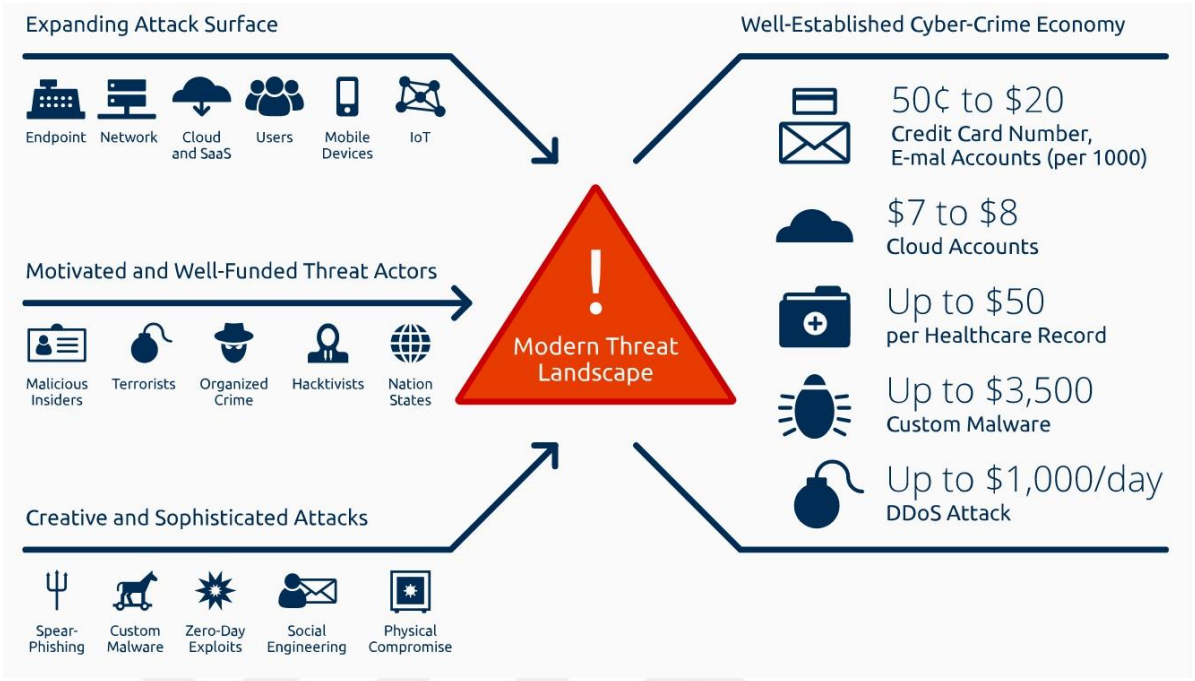

Figure 2.4 the evolving threat landscape

2.3. Cloud security issues

As mentioned above, there are some obvious security concerns that customers have when it comes to adopting cloud services for their businesses or even personally. Generally, these issues can be reflected as follows:

- **Integrity**: Integrity ensures that data held in a system remain in its original form and has not been edited by an authorized or unauthorized person. Backup routines are deployed to ensure no data is lost incidentally. When backed-up, the data is usually stored in a place outside the facility.
- **Availability:** This data makes sure that data is always available and never made unavailable by malicious action.
- **Confidentiality:** Confidentiality ensures that data is not in any way revealed to unauthorized persons. Loss of confidentiality is basically when any unauthorized person can view the data. Loss of confidentiality can happen on the internet or physically.

In order to map its security architecture, one must classify a clouding against the cloud structural model. In addition to taking into consideration business, regulatory, and other compliance requirements. Needs or gap analysis exercise can help reveal results. The outcome controls the main security posture of a service and how it's connected to an asset's assurance and security requirements. Figure 2.5 can be considered as a gap analysis in which it presents a way of how a cloud service mapping can be compared against a catalogue of compensating controls to determine which ones exist and which ones don't, according to the information and design that the consumer, clouding provider, or related company provide.

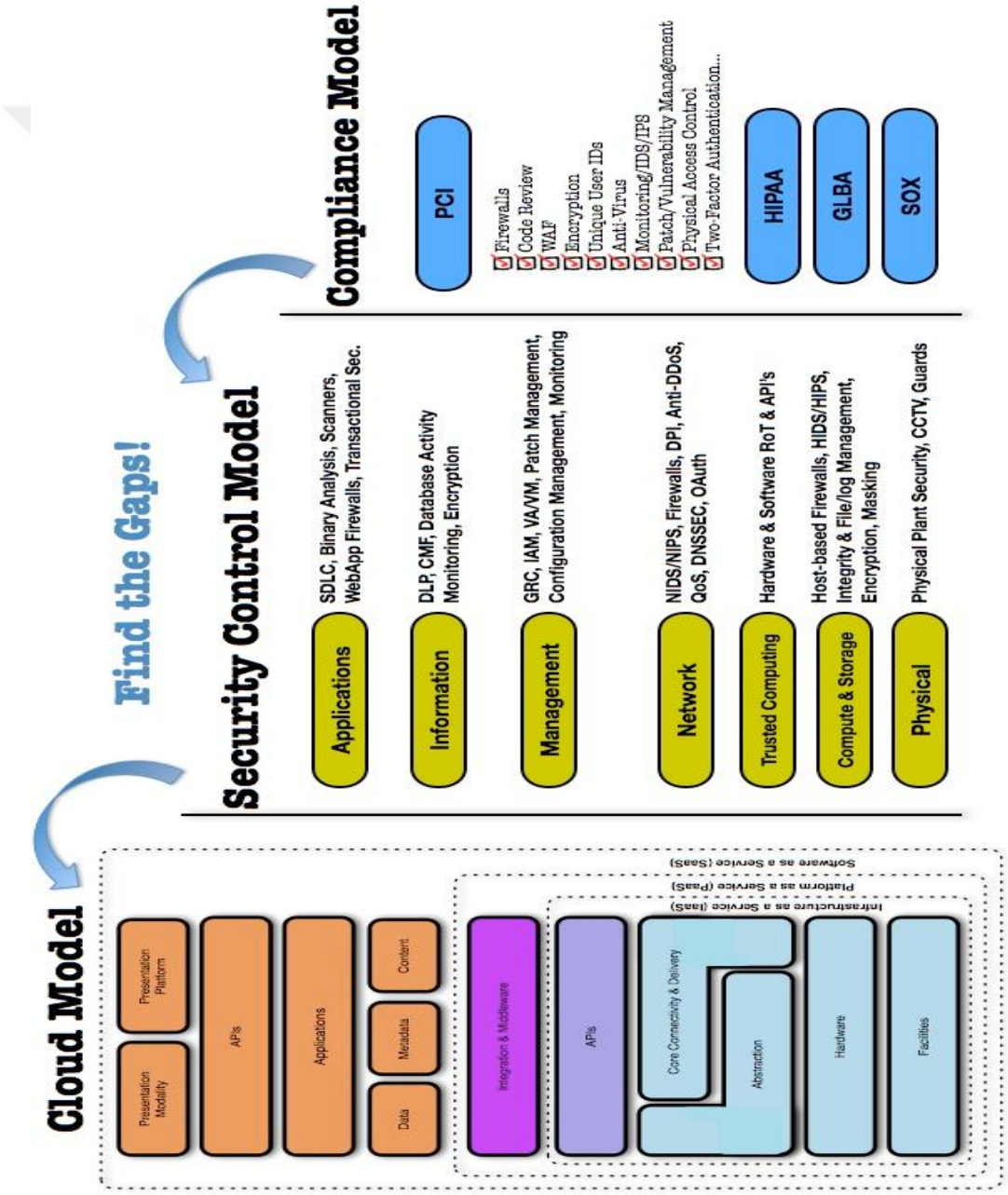

Figure 2.5 Compliance Model and Security Control Mapping the Clouding

As a requirement of the regulatory and compliance mandates this gap analysis is performed. Once it is complete, determining what action is needed becomes much easier, which in turn benefits and acts as feedback used in the risk assessment framework. As the name implies, the risk assessment and gaps are addressed according to this analysis, in terms of whether they should be accepted, transferred, or mitigated.

2.4. The Areas Of Critical Focus in the Cloud

Cloud computing areas of concern are set to detect the strategic and tactical security 'pain points' in a cloud environment, and can be implemented in any cloud combination services and deployment models.

The domains are split into two wide categories governance and operations The operational domains are concerned with tactical security and deployment within the architecture, whilst the governance domains are wide and detect strategic and policy problems within a cloud-computing environment.

The following tables summarize zones of high important focus regarding governance and operational:

1) Governance Domains

Table 2.1

Domain	Guidance deals with…
Governance and Enterprise Risk Management	This domain generally discusses how many organizations can protect their data and monitor their project risk presented by cloud computing. Topics about setting priorities for legal matters concerning agreements, giving users or organizations the ability to assess risk of a cloud provider, how responsibilities should be considered when making a fault of the user and the provider regarding important data protection, and how the boundaries may influence these points.
Legal and	The domain mainly discusses legal concerns about using

Electronic Discovery	Cloud. Possible issues such as protection requirements for data and computer systems, laws related to security breaches, regulatory requirements, privacy requirements, international laws, etc.
Compliance and Audit	Many compliance subjects are related to this domain such as how to maintain compliance during cloud computing use. Problems related to internal security and evaluating how cloud computing impacts compliance, also, the requirements to achieve compliance (regulatory, legislative, etc.) are Discussed here. And it explains methods of proving compliance during an Audit
Information Lifecycle Management	Managing information which is stored in the cloud. Elements related to the identification and controlling data in the cloud, also, physical control alternative which can be useful if we lose control while storing data in the cloud. And subjects related to the responsibility for data confidentiality, Integrity, and availability are discussed.
Portability and Interoperability	The capability of moving service or data from one provider to another, or getting it back to the original provider. Problems related to interoperability between providers.

2) Operational Domains:

Table 2.2

Traditional Security, Business Continuity and Disaster Recovery	The subjects of this domain are how much cloud computing has an impact on the operational processes and procedures which have high security, also disaster recovery and business sustaining. It also focuses on examining the potential risks of cloud computing, to increase the debate on the huge demanding for a safer project risk management model. And, this scale also tries to help people to address cloud computing strong security points which eliminates some security risks, and when it may bring some.

Data Center Operations	Evaluating the provider data center infrastructure and functionality. So this domain focusses on giving clients information about data center attributes which would be essential to on-going services, also attributes which are essential to long-term stability. Incident Response, Notification and Remediation Proper and adequate incident detection, response, notification, and remediation. It is concerned with attempting to identify items which have to be set at the user level and the provider's level to enable proper incident handling and forensics. That would help people to comprehend the complexity which cloud cause to the current incident-handling program.
Incident Response, Notification and Remediation	This domain focus on correct and sufficient incident detection, response, notification, and remediation. So it's attempt to identify objects which must be in the right place in the provider's level and the user's to enhance suitable incident handling and forensics. And it will illustrate the complexity which comes with using cloud to the current incident handling program.
Application Security	It discusses the security of applications which are operating in the cloud or getting developed. Also, it discusses subjects like is migrating or designing an application to run in the cloud a proper way, and if it is, which sort of cloud platforms is more convenient (SaaS, PaaS, or IaaS). Some certain security issues related to the cloud are also discussed.
Encryption and Key Management	Discussing the most convenient way to use encryption and scalable key management. Also, this domain has more information regarding the cause of why these features are wanted and what issues might come from using it, both for protecting access to resources as well as for protecting data, and it is not prescriptive,
Identity and Access Management	Identity management and leveraging directory services to provide access control. This will discuss some issues which

	might happen when an organization extends its identity into the cloud. This section provides insight into assessing an organization's readiness to conduct cloud-based Identity and Access Management (IAM).
Virtualization	This will discuss using virtualization technology in cloud computing. This section will identify subjects like the risks associated with multi-tenancy, VM isolation, VM co-residence, hypervisor vulnerabilities, etc. It will also discuss security issues related to system hardware virtualization, rather than a more general survey of all forms of virtualization.

CHAPTER 3

OVERVIEW OF OPENSTACK

3.1. BACKGROUND

OpenStack is an open source software which anyone can use to create both private and public cloud. OpenStack's main goal is to provide users with an open source cloud platform which should satisfy public and private need of cloud regardless of size, by being simple to implement and massively scalable.

The Platform was founded by Rackspace and that of NASA. It all started when NASA was trying to build a private cloud on top of Eucalyptus, but encountered an issue since Eucalyptus could not be scaled and open as NASA hoped it should be. As a result, NASA has made a decision to make a cloud controller from scratch and started a project called NOVA. While NASA was making its project to be open sourced, Rackspace was about to do the same with both its compute engine and storage controller. This made the two to collaborate and ended up creating an open source cloud software. OpenStack was first launched in July 2010. Later in October of the same year, the first project (Austin) was released and made available to the public.

3.2. OpenStack history

Since it was founded in 2010, OpenStack is a relative newcomer to the cloud technology scene. It was initially a collaboration of components due to be made open source by Rackspace and NASA around the same time, and so was launched with much fanfare by these organizations. They were quickly joined by other major technology companies who were interested in collaborating on a centralized core for open cloud technology, rather than building their own in house. In 2012 the formal OpenStack Foundation was created with the support of several founding organizations in order to provide an organization-independent body. The foundation serves to protect respective trademarks and handle legal issues that could arise within OpenStack, organize the participation of the various organizations involved and to

make sure a healthy development, user and operator community is promoted and maintained. To this end, the OpenStack Foundation is the leader in organizing the OpenStack Summits every six months, handles trademark disputes and even hires development talent to address specific concerns within the community.

The commitment to open source (along with the other three opens: design, development and community) is key to the success of OpenStack. Many of the organizations who now put development resources into OpenStack do not necessarily want to give control to and pay a vendor for their infrastructure stack or are hoping to compete in that market. By using OpenStack, your organization can build their own cloud platform, offering everything from the basic Infrastructure as a Service with a cloud computing offering, to building a comprehensive object storage cloud where users can easily store their files on your scalable storage back end.

OpenStack services interact through a series of APIs that each project supports. Calls to these APIs are handled by a messaging queue that is installed early in an OpenStack installation. All interactions go through the queue so that processing can be reliable and predictable, happening in the order in which they were issued and providing a buffer when there is a spike in usage so that none of the commands issued get lost. In order to interact with this queuing system, most post requests directly, but you will notice that a small subset of the OpenStack components also run with a scheduling daemon.

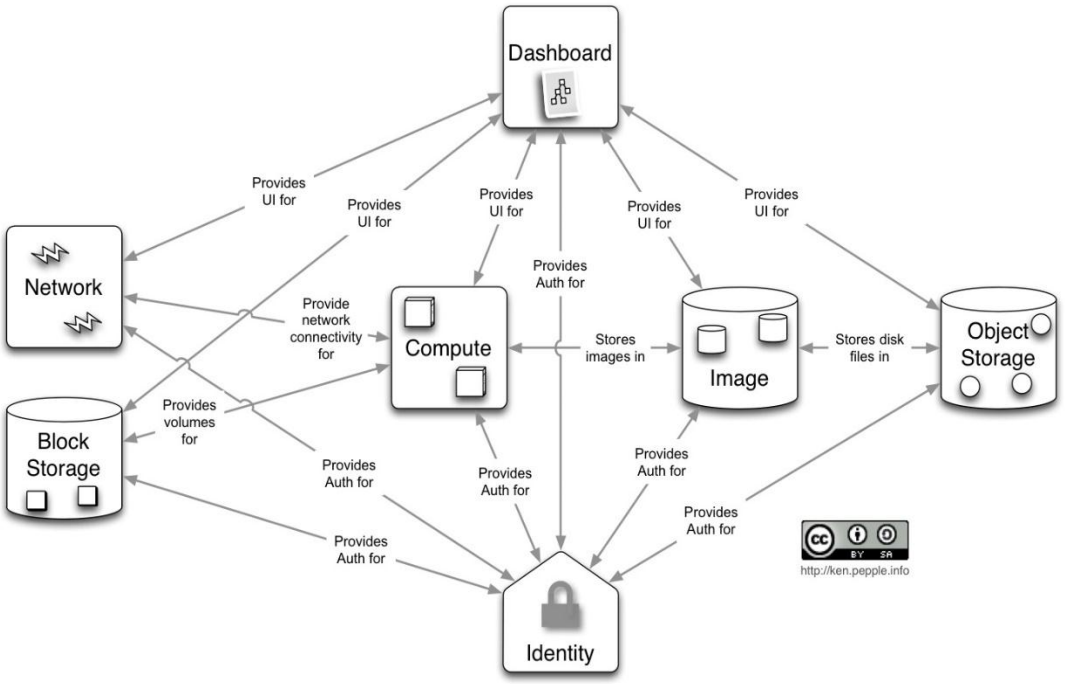

Figure 3.1 The core architecture of OpenStack.

in figure 3.1 Swift makes storing and retrieving files easier. Cinder handles the machine state of virtualized servers and in turn stores it as an image file within Glance. NOVA's main purpose is to make services related to virtualization a lot easier.

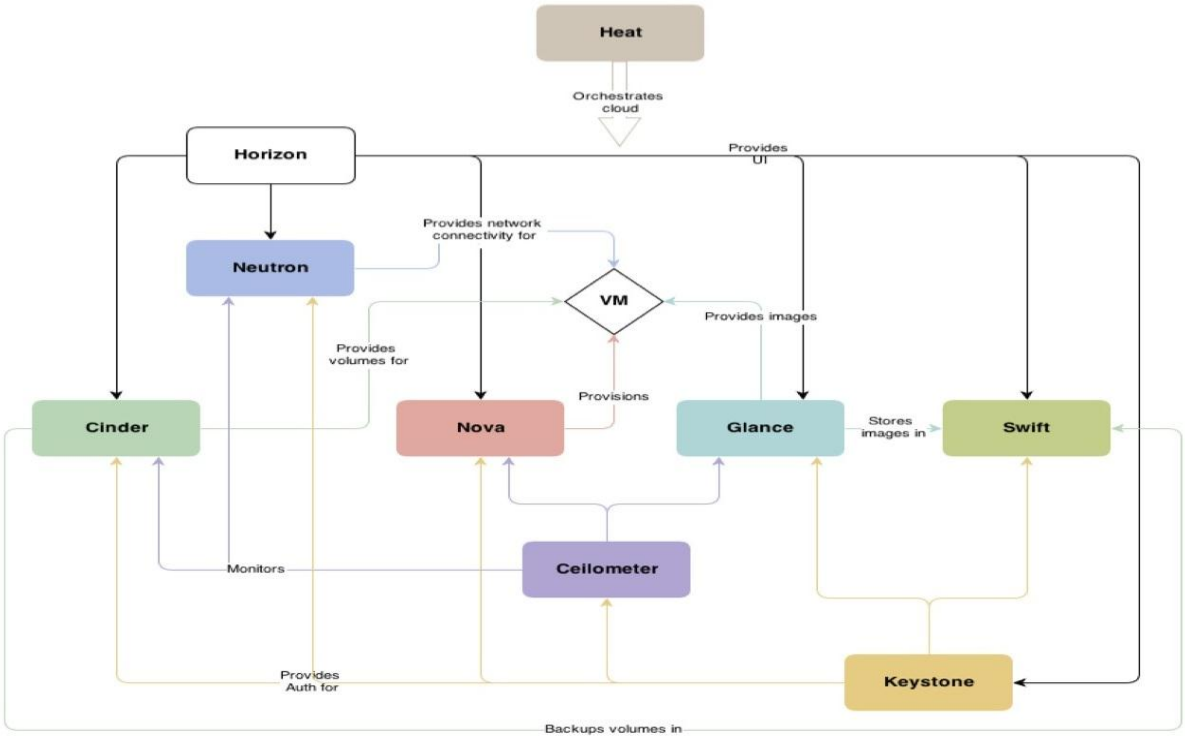

Figure 3.2 OpenStack conceptual architecture. [12]

3.3. Compute service

It is also known for its code name as Nova. It’s from NASA’s compute files and it provides and manages large networks, users and projects. The aim of Nova contain the hard-ware and hypervisor agnostic. Nowadays, it supports different virtualization technologies such as; Citrix XenServer, UML, Microsoft Hyper-V, Xen, KVM, QEMU, VMware and LXC Containers.

Nova was the early name of the project OpenStack compute service. It is a system which permits clients to host and manage clouding and the primary components of IaaS system. No virtualization needed in nova. As an alternative, it defines drivers to communicate with the underlying virtualization mechanisms. Many different

hypervisors are under nova's control by using an API server. The API server can control and command the hypervisor, storage and networking programmatically available to users. All communications between components are done via the message queue, and in turn, it prevents blocking responses which are sent from each component. The main purpose of the nova is to manage instances which are running on the host machines. Nova is the compute component of OpenStack. Along with being one of the most well-known projects in OpenStack, it is also one of the two projects launched when OpenStack was announced as a project in 2010 (the other was Swift for object storage). The compute component handles provisioning and control over the compute servers in OpenStack. As noted earlier, Nova has a variety of drivers that enable it to talk to a variety of VM and container technologies, including: libvirt (supporting qemu/KVM), Xen, VMware, Hyper-V and Docker. This component consists of the following core daemons and services that most installations will have:

nova-compute: Accepts actions from the queue in order to perform common actions on an instance, like starting and terminating.

nova-conductor: In order to avoid nova-compute having risky direct access to a database that can cause irregular data, the conductor controls interactions between the compute daemon and the database.

nova-scheduler: Controls interactions with the messaging queue, picking up requests from the queue, determining which Nova compute instance to send it to and passing it along.

nova-api: The API service that Nova runs so that other services, the CLI and Horizon can interact with Nova.

nova-api-metadata: The API service that responds to metadata requests that returns data about specific instances. Unless you're running a single-server instance of OpenStack (typically only for testing or development), some of these services will be divided across servers. For instance, to provide sufficient isolation, the nova-compute daemon should be on a different server than the database and conductor.

3.3.1. Dashboard (Horizon)

Horizon is the web-based dashboard for OpenStack that provides an interface for both OpenStack administrators and users of the platform. It enables administrators and operators to manipulate various settings related to users, servers, quotas and

more. The dashboard is also easily customizable by organizations and cloud operators who wish to add their own branding to the dashboard. The capabilities of Horizon expand as OpenStack continues to grow and add features with each release. The interface for administrators provides the capability to manipulate users, view system information and adjust defaults and much more Dashboard is a user-facing view that enables individual users to control their virtual machines, networking configuration and components made available to them by an administrator.

While many operators choose to use the command line interface and API for automated control of large fleets, the Horizon dashboard page for managing instances that enables quick and simple deployment of individual instances without having to have a highly sophisticated background in OpenStack. This enables it to be presented to users who have simple needs when it comes to small deployments of instances and provides a clean interface for completing various tasks.

3.3.1 Storage services:

This component is also known for its code name Swift. It is also from Rackspace Cloud Files and provides a storage system for large amounts of static data by the use of clusters of standardized servers. It gives a ReSTful API for uploading and downloading files into the cluster.

OpenStack cloud services have 2 kinds of storage, object (swift) and block (cinder) which supports many types of provisioning options.

Object storage: Swift is built to manage object storage for OpenStack. In contrast to direct file and block storage, object storage is built to be a highly scalable and available storage mechanism for storing files that are then accessible via a RESTful HTTP API. Note that all user-created files being loaded into Swift are called objects.

periodic processes: There are various periodic processes (replicators, updaters, auditors, reapers and more) run by Swift to do housekeeping on the data store itself.

swift-account-server: Handles management of accounts.

swift-container-server: Handles management the mapping of containers or folders.

swift-object-server: Handles management of the objects on the storage nodes.

swift-proxy-server: A service that accepts requests via API and HTTP in order to upload objects, change meta-data, and create containers. It’s also worthy to note, that

unlike many of the other OpenStack components, Swift can be run with an identity service other than Keystone. For all types of identity services Swift uses specially tuned middleware provided by Python WSGI middleware. A more comprehensive overview of how Swift works,

Block storage: Cinder provides the block storage component to OpenStack. Block storage volumes can provide persistent base storage to compute instances or to add additional storage to a running instance.

cinder-api: The API daemon that accepts requests and passes them on to the volume daemon for processing.

cinder-backup: A service that provides backing up of volumes to a storage provider.

cinder-scheduler: The daemon that does the work of selecting which storage node to create a volume on cinder-volume: Using input from the API daemon and the scheduler, this daemon actually does work of interacting with various storage providers. Cinder supports several different types of storage back ends, both open source and proprietary, that are provided by a plug-in infrastructure. More about how Cinder works and how you can use it to

3.3.2. OpenStack Image Repository

Glance is another name for OpenStack image service, which gives clients the ability to discover, register and retrieve VM images. API requests for the server or disk image and its metadata from users or OpenStack compute components are all accepted. Users can store VM images in different repository types, from simple systems in the host machine to object-storage systems like OpenStack swift. All images will Be uploaded and stored in the host machine back-end at /var/lib/glance/images/ [16]. It is possible to store the running instance and sign it as a backup, the backup and the stored image can be used as a template. Therefore, every time a new instance is going to be launched, the glance will provide it with the template which it will use. Actually, glance does not any images, but instead catalogs them and provides metadata from a storage back-end data store. Then, other modules should communicate with glance to fetch the image metadata.

VM images are the key objects which are uploaded and managed by glance. There are many different disk formats for VM images, like raw, qcow2, and so on. The image format of raw is the most basic one and natively supported by both KVM and Xen1 hypervisors. The raw image can be thought as the bit-equivalent block device.

The format of qcow2 stands for QEMU copy-on-write version 2, which is commonly used by the KVM hypervisor. It has smaller size and supports for snapshots. Therefore, this is often used as the image format in OpenStack [17].

Networking (Neutron)

Neutron is the networking component of OpenStack. It provides a variety of networking options for your OpenStack cloud. Defaults will set up a private network that instances live directly on and an Internet routable (or public) network that you will need to connect instances to in order for them to get a public address.

The project began as a subset of Nova called nova-network and eventually branched off into its own project that was previously named Quantum. The previous project name Quantum, Neutron has been built to support a vast array of network configurations made available through a series of plugins. More plugins are added each release but currently include various network bridge types, virtual local area networks (VLANs) and subnets, and plug-ins for several proprietary networking switches and other network tools, both physical and virtual. Neutron requires a daemon to process requests, and you need to run plug-ins and agents to complete specific tasks depending on your deployment..

Telemetry (Ceilometer)

Ceilometer is the telemetry module for OpenStack, meaning it provides the following functions:

-Polls for metering data on a defined set of OpenStack services

-Collects metering data and events by monitoring notifications

-Publishes collected data to specified targets, including messaging queues and traditional data Stores. These functions make it so Ceilometer is not a light-weight addition to your OpenStack deployment. The benefit that Ceilometer brings to your deployment is alerting that is integrated, which you won’t necessarily find in third part alerting systems. To accomplish the collection, metering and more, a large number of services are used, and, unless otherwise specified, they run on a centralized management server:

Ceilometer-agent-central: A horizontally scalable component to poll for statistics from various resources being tracked.

Ceilometer-agent-compute: A service that runs on each compute node to poll it for usage statistics.

ceilometer-agent-notification: A service to consume messages from the queue to build event and metering data.

ceilometer-api: The API that is polled to provide data that is used by the administrator or customer to examine usage.

ceilometer-collector: After consuming data from agents and other OpenStack services, this service is used to dispatch the data to whatever is used for metering data storage.

These services mean that Ceilometer can track usage of services for the purposes of billing or other accountability of users. Ceilometer can be used on its own or in collaboration with the Gnocchi project,

Baremetal (Ironic)

As organizations moved to using OpenStack to manage everything from their virtual machines and containers to their object storage array, it became clear that there was a desire to control physical machines as well. This began with a nova-baremetal project but was quickly moved into its own project, Ironic. Ironic serves as the Bare Metal Service, provisioning physical machines rather than virtual machines and providing a series of drivers that support the most common management tools like Preboot Execution Environment (PXE) and Intelligent Platform Management Interface (IPMI). Additionally,

ironic-api: The API that processes request and sends them to the ironic-conductor.

ironic-conductor: This service completes the tasks of adding, editing and deleting nodes, handling power state, typically with IPMI, and the provisioning, deployment and decommissioning of bare metal nodes.

ironic-python-agent: A python service that is run in a temporary ram disk to provide ironic conductor service(s) with remote access and in-band hardware control. Ironic also has a python-ironic client, which is a command line tool for interacting with the service.

Orchestration (Heat)

Heat is the orchestration service specifically built for OpenStack. By using a series of text-based templates, Heat enables you to spin up a collection of resources in what Heat calls a stack, which may include instances, network components and security rules, and more. Templates can either be in the OpenStack-specific Heat

Orchestration Template (HOT) or AWS Cloud Formation Template (CFN) formats. Like the others, Heat is comprised of a series of daemons and services:

heat-api: A RESTful API that is used to interact with the Heat service.

heat-api-cfn: Used if you wish to support the CFN template.

heat-engine: The core of the product that provides the actual orchestration service.

There is also a Heat command line tool that is strictly used to communicate with the heat-api if CFN is being used.

Containers (Magnum)

Magnum is specifically built for managing the orchestration around containers like Docker and Kubernetes. As mentioned in the section about Nova, there is a virtualization driver that enables Nova to control Docker, but this treats the container like a VM and takes advantage of very few features that make containers really valuable. Magnum seeks to take advantage of container features provided in the orchestration tools that come with Kubernetes or Docker Swarm, including the bays and pods structure you may already be familiar with if you use these container tools.

3.3.3. Identity service:

OpenStack identity service also named Keystone. It is used by other services to confirm the identity of a user. Confirmation of identity can be done through various mechanisms, from a simple user name and password or API key combination to an authentication token provided by Keystone itself. Once authenticated, Keystone has a concept of Projects, Domains and Roles that control what a User or Rather than running as a standalone daemon, today Keystone instead uses a Web Server Gateway Interface (WSGI). By default, Apache with mod_wsgi is used to serve up the Keystone service. It provides central authentication and authorization for other OpenStack services. For instance, when a user is going to access OpenStack services, the user has to go through the following steps [7], as Depicted in Figure 3.3:

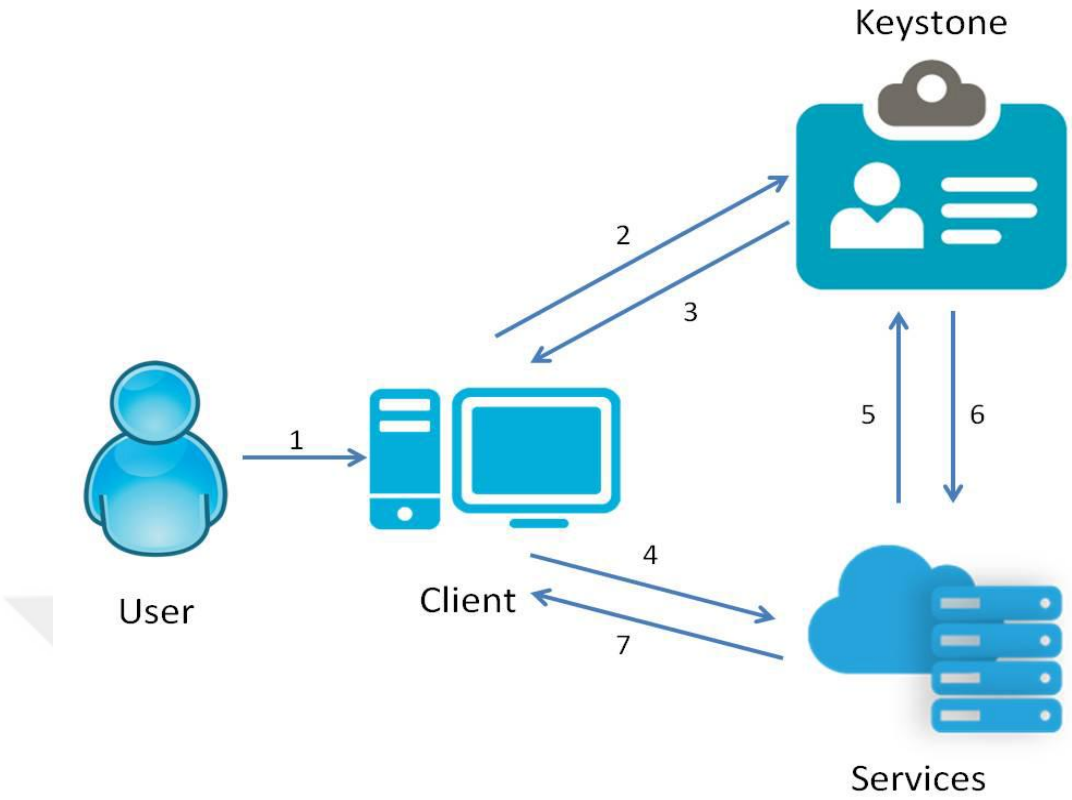

Figure 3.3 OpenStack authentication and authorization

1. The user should use the credential to enroll to the service client.

2. The client sends an authentication request along with the credential to the Keystone server.

3. A scoped token and a list of endpoints to different services are sent back to the user.

4. The scoped token along with the user's request are sent to the service.

5. The scoped token is sent to the keystone server for validation.

6. Authenticate the user with the service.

7. The user's request is processed and the response is sent back to the client. As the authorization and variation will be passed among all other OpenStack services to proceed them to be used, keystone should be the rest service to be configured [9].

Projects, Domains, Roles, Users and User Groups

OpenStack is built to be a flexible infrastructure for a variety of different types of consumers of the cloud. However, the existence of projects, domains, roles, users and user groups within an OpenStack installation can be confusing at first.

Projects—previously called a Tenant, projects are a group of users that have the same quotas and share resources such as cores, memory and storage. An administrative tenant is configured to isolate the administrative user from other users on the system. A project is typically created for each segment of your organization that will share resources.

Domains—High-level account containers for projects, users and groups, separate domains can use different authentication back-ends.

Roles—Define the operations in the form of rights and privileges that a user can perform in particular projects.

Users—Individual accounts used to interface with Keystone for access to OpenStack services.

User Groups—A collection of users. A user is a member of one or more projects and is assigned roles. Roles exist within projects. There is no way to have a role outside of one, with the exception of admin. While it's granted within a project, once granted it's global. Finally, you will need to unset your token-based authentication environment variable and set the new password-based authentication credentials that you just set up to test that Keystone is working as expected. The OS_PASSWORD will be the custom password

CHAPTER 4

INTEGRATION OF THE OPENSTACK WITHIN THE CENTRAL SECURITY SYSTEM

4.1. Background

In this chapter a demonstration of the data a required to integrate the modern OpenStack project with the Central Security System. The first part explains how to integrate out identity management system with Keystone. The second one discusses a method for providing a different access control and authorization, i.e. SAML and XACML, to provide SSO to the OpenStack environment, integration of the LCA is also explained.

4.2. Authorization Techniques

It has been noticed that the cloud itself contains some security problems along with cloud computing, it has been cited many times that security is a major concern in cloud computing. Because organizations were concerned about their safety of remote storage. A lot of authorization techniques have been examined during the past years, developed and abandoned. One of the simplest authentication methods is the username and password combination. But the weakness of this authentication method has been well documented. In order to increase security, the user has to present additional data for multi-factor authentication, this information can be a key from a physical key-generator, shown in Figure 4.1. Even though, these systems provided improved security, they have been considered inefficient, because of the security design and extra costs.

Figure 4.1 A RSA SecurID 700 Authenticator KeyGen. (TokenGuard, 2014)

4.3. Federated Authentication

Nowadays, users are more likely to own up to 10 extra accounts for their different services in the cloud. Furthermore, the great increment in applications number has led to a difficulty in managing passwords which are important to access them. Many customers can't memorize a lot of complicated passwords, so instead they use one password for their accounts. Mass centralized storage of costumer's credentials was criticized along with conducting key-derivation as a high risk, high value target for hackers. Federated management techniques have been developed to share identity attributes between services without centrally storing this information [19]. With the use of a federated identity, cloud's security will increase, and the user can choose a strong single password which might be verified by a specialized IDP (Identity Provider).

Federated Identity Management (FIM) relies on a mutual agreement between service providers on a standardized list of attributes that refer to a user [15]. Furthermore, In addition, to make exchanging of information between services, they should adopt a common protocol. organizations may implement a shared name identifier to assemble user's data between their different domains. The proliferation of FIM methods can be seen with the common use of Facebook, Google and MySpace ID as Identity Providers for third-party services [10].

FIM provides a very good level of access control which has a great value for large scale organizations and their collaboration. If we consider the example of a university library. To access online journals and resources, students can use a single identity. Service providers do not care about the individual's identity of every reader. Instead, service providers tend to make sure that every user accessing materials is from a valid, subscribed institute. universities manage their student's identities, so that becomes a major factor in understanding conceptual model of FIM; identity-related data are not collected nor maintained by service providers. Also, If users were given more control over their (sic) own identity attribute, they will experience more protection and privacy. Actually, users can assume they have a relative anonymity because they get treated as a member of an institution rather than an individual.

Federated technologies were initially developed to facilitate web-based Single Sign On (SSO) [20]. Even though, a conflict in terms might appear, we can think of SSO as a collection of Federated Identity Management; a functionality which results from its successful implementation. Segment 4.4 discusses SSO in greater detail.

Even though, analyzing FIM technologies have showed a big benefit, the rate of adoption in the industry was not big as it was expected to be. If the security token or user's credentials was the fted, it can compromise all of the federation partners. Furthermore, the possibility of token theft undermines security within systems with more secure processes, such as two-factor authentication. Once an identity has been stolen, access to all federated service providers is possible, greatly simplifying the job of the attackers [8]. In a critique of FIM methods, [11], state that federated identity management blurs security boundaries and thus creates liability and privacy risks. Even though, this risk is relatively correct, it comes from the continued use of simple username and password combinations and provides compelling motivation to implement a viable alternative.

4.4. Single Sign on (SSO)

SSO (Single Sign-on) technology was developed to make access controls which link users to many independent services of the cloud. When a user enters his/her login information, they are provided with access multiple resources without the need to re-authenticate. With the use of SSO users don't have to memorize too many passwords, and it also facilitate the administration of a system. Service

providers and IdP (Identity provider) should create a trust relationship to make implementation viable, also known as a Federation. By making this relationship, the service provider can suppose that users are valid and authorized correctly and in response they create a local session. Figure 4.2 Shows a high level conceptual model of an SSO scenario.

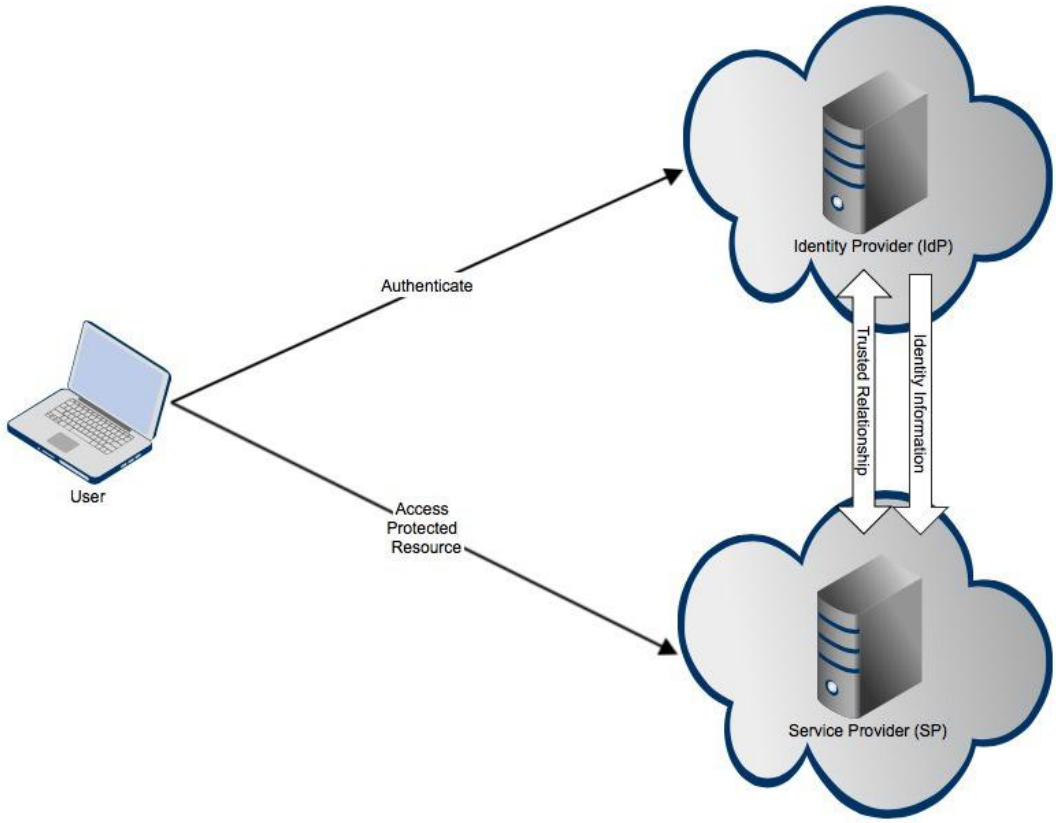

Figure 4.2 A Simple SSO conceptual model.

SSO deployment profiles gives the IdP the ability to be a service provider itself. So, the authentication step shown in Figure 4.2 may contains the use of a service provided by the IdP. In old technologies, browser cookies stored authentication status, however, it was a problem with cross domain access. Because browser cookies were never transmitted between DNS domains, authentication information is lost in translation. The multi-domain SSO (MDSSO) problem is addressed by a number of open source protocols that seek to standardize the transfer of authentication information from one server to another. These include: OpenID, OAuth, SAML, and Kerberos.

4.5. SAML

SAML stands for Security Assertion Markup Language, it is a famous authentication example. A standard which is widely-used in SSO Federated environments. It uses an assertion an encrypted XML file which summarizes attribute and identity data. Because SAML can have a lot of deployment profiles, it can make a choice to drop any direct link between the Service Provider (SP) and the Identity Provider (IdP). So as an alternative, SAML uses a series of HTTP redirect instructions, in order to maintain its trust relationship, it uses the client seeking authentication. Figure 4.3. Shows a SAML configuration using Service-Provider-Initiated SSO

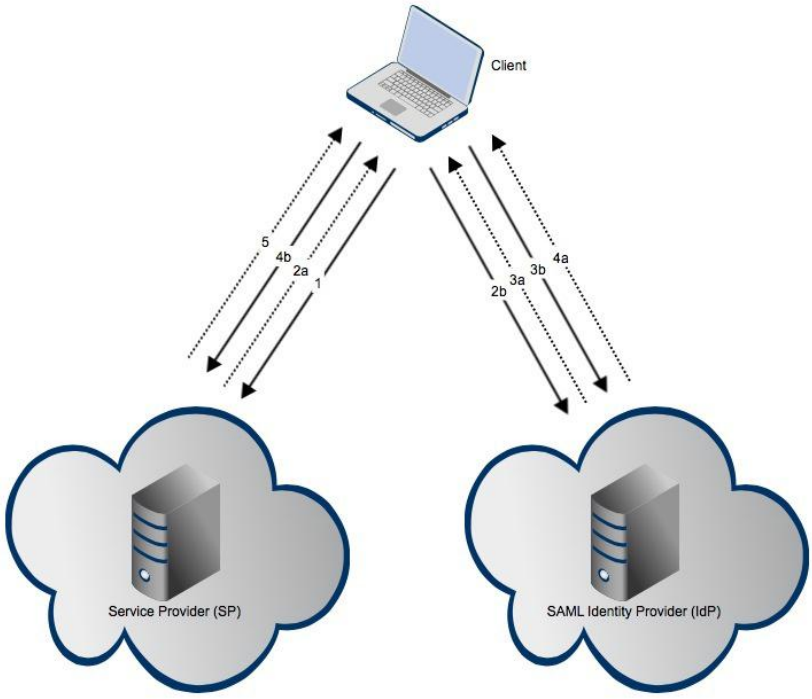

Figure 4.3 A SAML protocol in a Service-Provider-Initiated SSO scenario.

In the deployment configuration shown in Figure 4.3, the SAML protocol adheres to the following authentication sequence:

1- The consumer tries to access a service on the SP (service provider).

2a- User gets redirected by SP with an HTTP 302 status code. the client receives a SAML request.

2b- IdP gets the SAML request from the client in HTTP GET request forum.

3a- If the user already has a session then step 3a and 3b are not required. But, if there is no session, users are prompted for login credentials.
3b- user's credentials are sent.
4a- The IdP returns a SAML assertion along with an HTTP 200 success code.
4b- this assertion is redirected to the SP, which is verified.
5- Finally the user has authorization now and can access the protected service Provider. SAML exchanging happens between 2 parties, an asserting party, which is an entity responsible for asserting, and a relying party, which makes use of the assertion that it has received. In case of SSO, an audience restriction is included in the assertion to give a specific Service Provider (SP) the ability to use it. The assertion caries both important ancillary information (SP's name, access policy used to give access), and the identity of the user (their username and e-mail).

4.6. Keystone - OpenStack’s Identity module.

The Keystone module has two major goals registering the user and their permission, and to give costumers catalogues related to the services and APIs which it can support. The primary service in OpenStack receives authentication from a policy of token-based access, therefore, users should first authenticate themselves before use. The identity module Keystone manages all the tokens, catalogs and policy features which protect access to any of OpenStack services. When the user enrolls in Keystone they might get registration with a project (known as tenant). A project contains configured services for a determent applications. The main purpose for roles is providing a mechanism to set the extent of client's permission to services and a client might assign to any number of roles within a given project. The token's main purpose is to manage information of this complicated combination of authorization. After the user gets authorized, services will refer to that token and grant access applicable. To make that happen, each service in OpenStack should have a trust relationship with Keystone. To get a token using the conventional username and password authentication a user would complete the sequence shown in Figure 4.4.

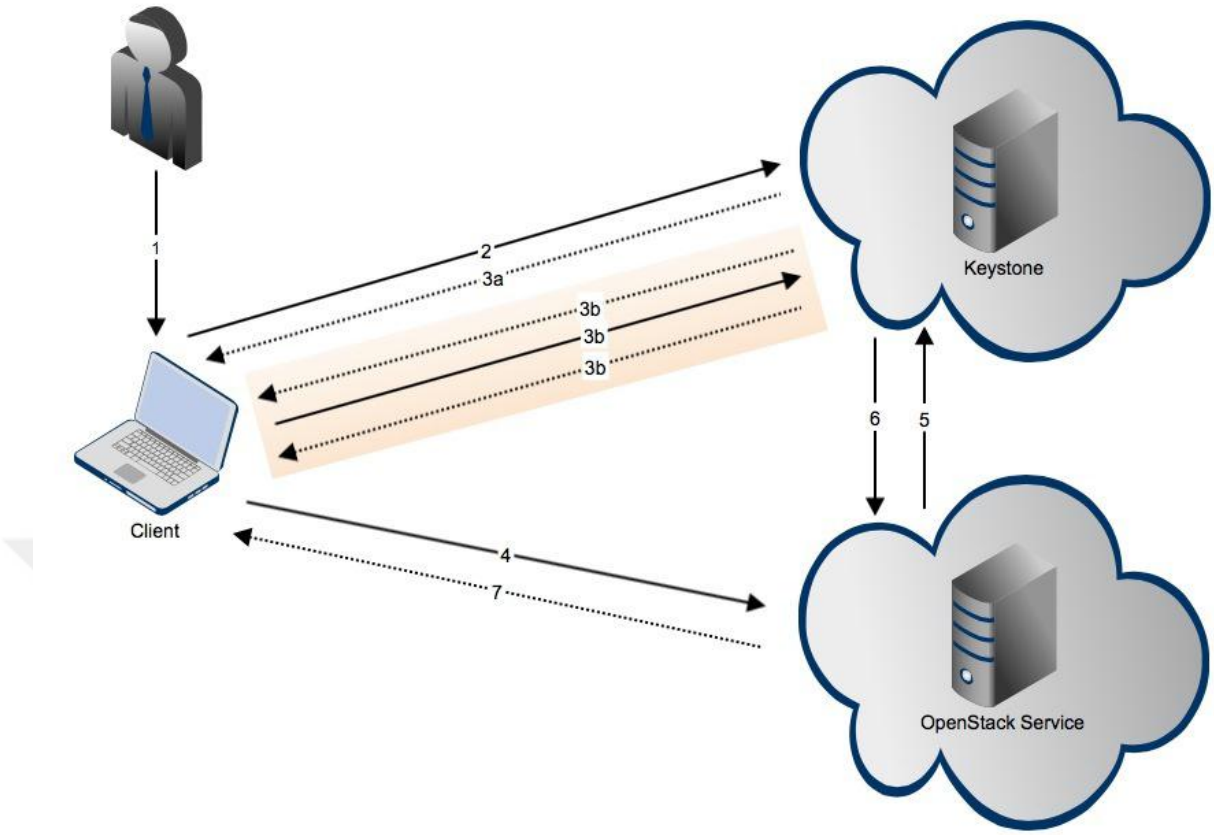

Figure 4.4 The authentication process for 'regular' username/password login.

1- The user opens the client for the planned service. Therefore, they issue a command, Providing their user credentials and an endpoint for the service location.

2- The client makes a request to Keystone in case they have no token, forwarding the necessary Credentials for validation.

3- If the credentials are valid, Keystone may choose one of the following actions:

   a) 3a- A token is created when the user specifies a project and it returns to the user, and it carries with it a menu of endpoint for the services, the project uses them. That it also known as a scoped token.

   b) 3b- If no project is specified, Keystone can only provide a token that validates the user. This UN-scoped-token may be used to list available projects. Once the user specifies a project ID, the un-scoped-token is returned to Keystone and exchanged for a scoped token. In OpenStack,

tokens are scoped to projects. With no scoped token, no access can be granted to any of the services.

4) From the list returned by Keystone, the user chooses the endpoint of the service and makes a request, this time including the scoped token.
5) Here the trusted relationship is utilized, the token sent to the service is returned to Keystone for validation.
6) If everything is valid, users get authorized with the services they need. the project gets returned to the client by Keystone.
7) If the request made by the user who within the scope of their role, user is authorized and the response is returned to the client

When a request gets issued, to avoid repeating the pervious process again, a token is assigned a timeout and can be stored locally within the client on a 'keyring'. When the token expires, the user must repeat the authentication process and acquire a fresh scoped-token. In rare situations, an admin token might be used to do managing tasks but it doesn't carry any clear authorization, and should be disabled once the Keystone server has been configured.

4.7. The Keystone clients.

Users are able to give commands to an OpenStack installation by using either a command-line client or a web-based GUI (graphical user interface) (Horizon). Even though Horizon's feature-set doesn't have the ability to execute batch processes, it can significantly facilitate the user experience. In some cases when the administrator wants to automate a big part of tasks, the command-line can provide an effective form of interaction, with more promising efficiency than the GUI (graphical user interface).

OpenStack's command line utilities harness standardized Representational State Transfer (REST) with Application Programming Interfaces (APIs) for its compatible service. REST architecture simplifies communication between a server and client, with no need for each of them to know each other. REST is described to have no state because all the important data which are needed for a successful request is sent with each communication. In OpenStack instance, the requests which need to use REST get sent to server endpoints in the standard HTTP protocol.

In old versions, a client is needed in order to manage user interactions of OpenStack services. With the release of OpenStack Havana all of those standalone clients have

been replaced with the new universal client. With the release of this new client, OpenStack functionality has been expanded, simplifying compatibility with Havana's v3. 0 Identity API. For maintaining back-compatibility, the standalone clients have been preserved inside the source code, however, it doesn't have the required functionality in order to interact with the latest versions of the API. In the future versions, it will probably be removed and replaced with the universal client. In section 5.0, the difference between the APIs and their clients is explained.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1. Background

In this chapter prototype implementation method of our thesis are discussed using OpenStack. This prototype implementation has been done to explain and test identity and authentication services, depending on one-time password and PKI in Keystone. So, we have to SSL configure the server to be able to implement PKI in the Keystone Web server. Thus, the Local Certification Authority has to pass the certificates for the web server and Keystone.

5.2. IDMS Server Implementation

IDMS server is a standalone server which preform CRUD operations with end-user data and a functional element of our Central Security System. IDMS server also does the registration job for other three elements of our Central Security System. The IDMS server depends on SCIM standard and follows standard protocol. IDMS server communicates with Keystone identity back-end module, also, it is responsible for identity verification. Therefore any user who is registered in IDMS server will also get registered in the Keystone database. Thus, every time a user needs to register himself in the IDMS server, an API call to Keystone identity back-end module is made. However, user role should be set by the PDP server before any call to the Keystone back-end module is made. Which means the PDP sets the role of the user authorization and depending on that user information such as e-mail, username, etc. are sent to the PDP server. PDP and IDMS servers make an API call to the keystone back-end module identity when they are synchronized. After that the response of the Keystone API call get stored in IDMS database, which includes a username, password, role, tenant information and, 32 bits unique random ID along with other important and optional parameters needed in the user registration process. The runtime implementation of the IDMS server for user registration

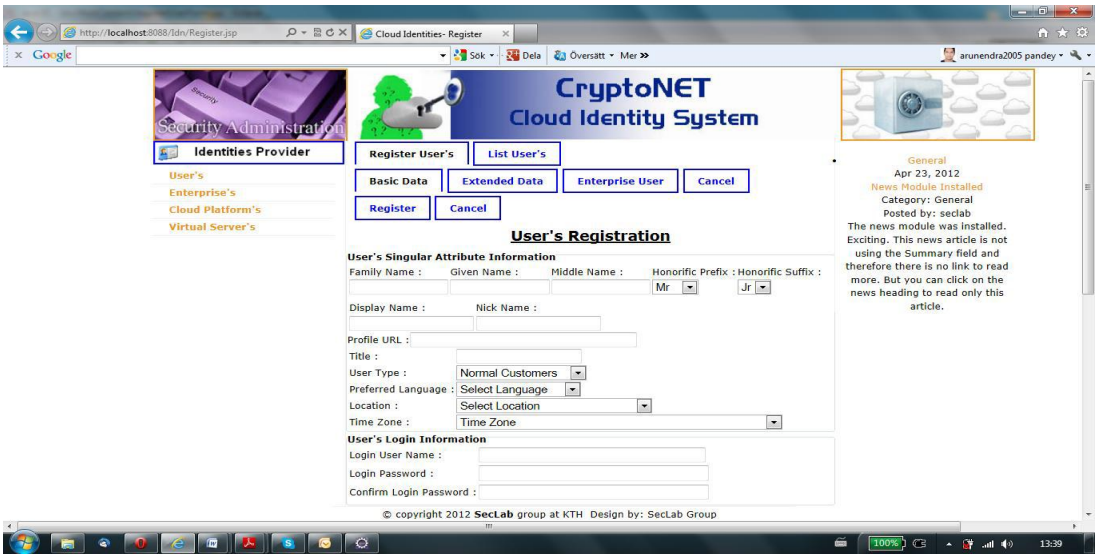

Figure 5.1 IDMS server User Registration Form

5.3. Development of IDMS

The OpenStack clouding system is used Keystone to manage the identity and disturbed the authorization to other OpenStack services. That means Keystone is responsible for all user management by performing CRUD (Create, Read, Update, and Delete). One of our aims is to isolate the identity information from the users that result of that providing identity services through our CSS, the CRUD management operations had provided by the CSS (central security system) to the users through the Web Services and we hide Keystone in the back-end modules. This satisfies any need of separating the identity providers because IDMS be the Central Security System which does the identity provider job, the user must register in IDMS first, so it can login OpenStack, then the IDMS sends API message send to Keystone inform the system of new user information and his authority. The IDMS is built on System for Cross-domain Identity Management standard, which offers user registration. After the user being registered, and depending on the credentials as ID, all the information like, email transferred to (Authorization Server) the IDMS, which depends on his record to give corresponding user Authorization SAML ticket to make sure that the user can access to the services and resources of OpenStack. All information must be saved in PDP server that make a real time synchronized between IDMS and PDP server, so both can send message to the Keystone to identify the user. The reply of the Keystone message stored in an IDMS server database, that contain all necessary information like user name, authorization, password, the unique

ID (32- bit), tenant information combined with some optional parameters required in the process of user registration.

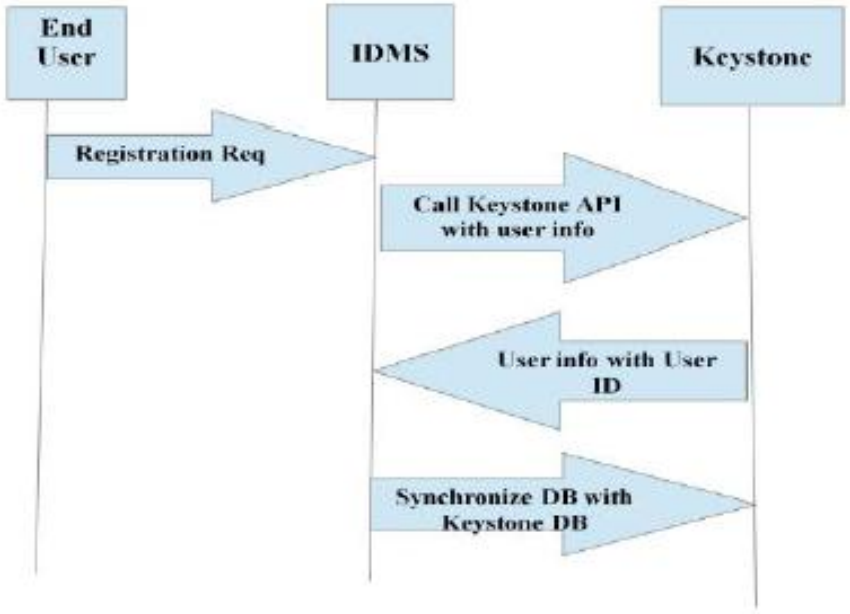

Figure 5.2 Integration of IDMS with Keystone

5.4. Authentication using PKI

We implement an LCA server to be the authentication server of Keystone, the methodology is to make all connected components certificated based authentication. Local Certification Authority server and IDMS server must be synchronized to make successful user authentication based on certificates. LCA server is a part of the Pre-shared key hierarchy. User information should be in IDMS to generate a key - pair, the public key of the user is used to make the certificate request. And the private key still hidden in the user system then the certificate and can be verified by the IDMS server.

5.5. SAML Token with SSO / Authentication

Keystones are used as interface for OpenStack provide integration of SAML protocol for SSO into OpenStack. That means after we connect OpenStack libraries and plug-ins with keystone identity, we can use SSO/authentication based on SAML token.

We have to synchronized all components of the OpenStack environment to run authentication/SSO based on a SAML token, IDMS server is registered all OpenStack services and components. And even all requests made by user with

identity verification. LCA server generate X509 certification after successful identity verification user. And send it to the SAML server, then depending on the login information that determine the authority and identity information passed to the SAML several new ticket then be sent. SAML ticket consists of user information include (identification information, Tenant, the keystone establishes a secure channel with any OpenStack service request to verify SAML ticket and provide a response.

5.6. Authorization based on the XACML Policy

In our study the Authorization depends on the XACML policy that studied on the idea of making all the Security system well-organized and efficient in a cooperative situation. That means all the configurations and component of the keystone identity must be synchronized by the Central Security System. We choose a single Policy Decision Point to have the authority of controlling the authorization processes. The responsibility of management of groups is made by PDP server, the security administrator defines the roles of the XACML policies, depending on authority and how access to the system. This gives the process that when the user needs a services PEP check the request and send XACML authorization request. The message contains Tenant information, and the permission. Depending on the security administrator policy, PDP server checks the request. And the answer is sent back to the PEP server. PEP has now accepted or refuse the demand, depending on the answer of the PDP.

5.7. SAML Implementation

We can modify Keystone to make it support SMAL, but unfortunately, we have to replace many components to achieve that, not just a plug-in upgrade for authentication. And what's worth mentioning is that, Keystone and other OpenStack components have reasonably determent APIs and interfaces, which can be used to integrate it to the PDP server for SAML implementation. So, we have to build a plug-in which communicates with the PDP server on top of the existing Keystone projects. Sadly, because of time shortage for this study and lack of information and experience on Keystone back-end modules, we had to leave SAML implementation for other research. PDP server can be able to assign the role to registered user in our IDMS server. Runtime implementation of PDP server for assigning roles is shown in Figure 5.3.

Figure 5.3 User Role Registration

5.8. Combination of Keystone and PKI

PKI system is depending on private and public key using the standard certification of X509. Keystone save the Public and Private Keys and certificates. After the system is connected and synchronized, any connected user can receive certificates from the Keystone by using REST web services. And the certificates can hack into the public network (internet). But if OpenStack service connects to PKI mode, all related services get public key from Keystone that can be used by the services to communicate. When the user uses OTP to authenticate to OpenStack system, Keystone generated a token in a PKI mode. Keystone makes a token JSON object which includes token's metadata, that is more secure by using Keystone's private key to encrypt it and then signs it by MD5. When the user has Keystone public key it can decrypt and verify the data. Because the data information was inside encrypted token, on this project we don't require for the user to Interfering Keystone message to have the token and therefore we provide both scalability and delegation of the authentication protocol. The one-time password in Ubuntu Runtime authentication of user with Keystone is shown in Figure 5.4.

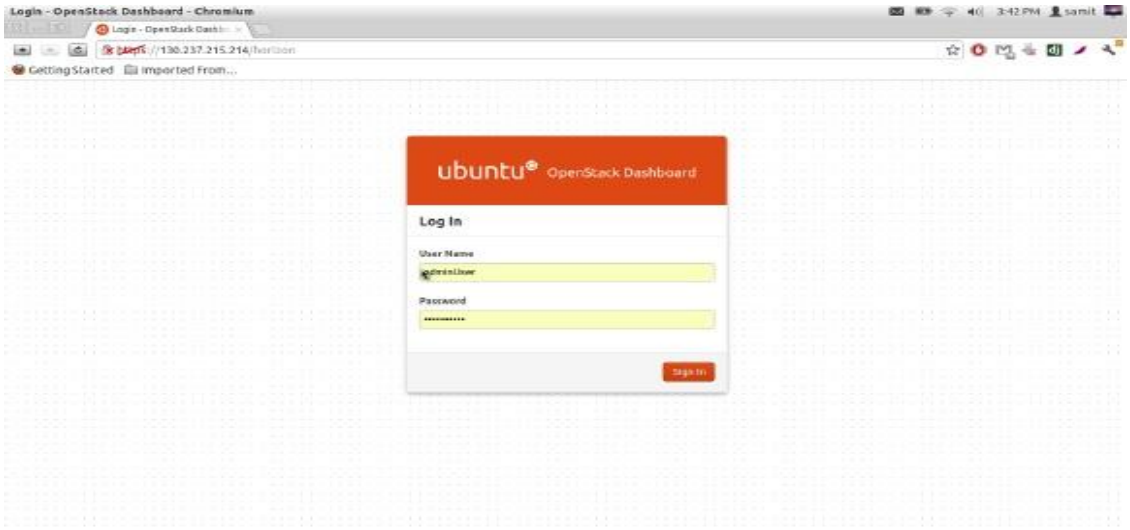

Figure 5.4 Authentication using One-Time Password in HTTPS

Figure 5.5 An Encrypted and Signed Token Provided by Keystone in PKI mode

5.9. Configure Keystone to run under Apache and enable Federation

We add some plugin to Keystone service, like lpsilonor shibboleth, so it can work with federation and SAML identity providers. Keystone need to work with Apache. Below is the main point that we work to config federations. AS we need to enable the federation extension, we need to config as follows:

- Add the federation extension driver to the [federation] section in the keystone's file.

```
[federation]
keystone.contrib.federation.backends.sql.Federation
```

- Add the *saml2* authentication method to the *[auth]* section in the *keystone.conf* file.

```
[auth]
methods = external,password,token,saml2,oidc
saml2 = keystone.auth.plugins.mapped.Mapped
```

- Add the *federation_extension* middleware to the *api_v3 pipeline* in *keystone-paste.ini (*enabled by default in OpenStack Liberty release*)*. This must be added after *json_body* and before the last entry in the pipeline.

```
[pipeline:api_v3]
pipeline = sizelimit url_normalize build_auth_context token_auth
admin_token_auth json_body ec2_extension_v3 s3_extension simple_cert_extension
revoke_extension federation_extension service_v3
```

- Create the federation extension tables if using the provided SQL backend

```
./bin/keystone-manage db_sync --extension federation
```

5.10. Configure the Keystone Identity Provider

This setup is required only when configuring OpenStack Keystone to act as the Identity Provider to another Keystone instance that is acting as Service Provider.

To enable Keystone services to trust each other, they would have to share secure certificates. The first step is to generate a self-signed cert-key pair. The IdP shares the resulting certificate with its SPs (through the IdP metadata), and signs all the outgoing SAML assertions with the key.

```
$ openssl req -x509 -newkey rsa:2048 -keyout /etc/keystone/ssl/idpkey.pem -out
/etc/keystone/ssl/idpcert.pem -days 9999 -nodes</span>

$ ls /etc/keystone/ssl/

certs  idpcert.pem  idpkey.pem  private
```

- With the certificates generated, and the federation extension enabled from the previous step, update the SAML configuration to configure keystone

```
# /etc/keystone/keystone.conf
[saml]</span>

certfile=/etc/keystone/ssl/idpcert.pem

keyfile=/etc/keystone/ssl/idpkey.pem

idp_entity_id=http://${idp_service_ip}/v3/OS-FEDERATION/saml2/idp

idp_sso_endpoint=http://${idp_service_ip}/v3/OS-FEDERATION/saml2/sso

idp_metadata_path=/etc/keystone/keystone_idp_metadata.xml

...
```

- With this configuration, generate the IdP metadata that will be shared with the SP Keystone

```
keystone-manage saml_idp_metadata > /etc/keystone/keystone_idp_metadata.xml
```

- setup the keystone SP in the IdP – This step inform the IdP about the URL to use when creating a SAML assertion for the SP.

```
$ curl -s -X PUT -H "X-Auth-Token: $OS_TOKEN" -H "Content-Type:
application/json" -d '{"service_provider":
{"auth_url":"http://${sp_keystone}:5000/v3/OS-
FEDERATION/identity_providers/beta/protocols/saml2/auth",
"sp_url": "https://${sp_keystone}:5000/Shibboleth.sso/SAML2/ECP", "enabled":
true}}' http://localhost:5000/v3/OS-FEDERATION/service_providers/sp_name |
python -mjson.tool
```

5.11. Configure the Keystone Service Provider

- Add the IdP Entity ID and metadata to the keystone SP by updating the configuration in */etc/shibboleth/shibboleth2.xml*:

```
<ApplicationDefaults entityID="https://<sp_ip or sp_hostname>/keystone">
        <Sessions lifetime="28800" timeout="3600" relayState="ss:mem"
                  checkAddress="false" handlerSSL="false" cookieProps="https">
            <SSO entityID="https://<idp-host>/keystone/v3/OS-
FEDERATION/saml2/idp">
                SAML2 SAML1
            </SSO>
```

- Add the IdP's metadata provider – Loads and trusts a metadata file that describes an IdP and how to communicate with it

```
<MetadataProvider type="XML" validate="true"
file="/etc/keystone/keystone_idp_metadata.xml"/>
```

- Add attribute mappings to the attribute-map.xml file. This takes attributes from the incoming SAML assertion and translates them to parameters that are exposed to keystone as CGI environment variables on the wsgi request.

```
<Attribute name="openstack_user" id="openstack_user"/>
<Attribute name="openstack_roles" id="openstack_roles"/>
<Attribute name="openstack_project" id="openstack_project"/>
```

Upload the keystone mapping. This maps the remote attributes sent by the SP to the local attributes in the IdP

```
{
    "mapping": {
        "rules": [  {
                "local": [
                    {     "user": { "name": "{0}"    }  },
                    {     "group": { "id": "1234"    }   }
                ],
                "remote": [
                    { "type": "openstack_user",
                        "any_one_of": ["devuser@pf9.net"]     }
                ]      }
        ]  }
}
```

This setup can now be tested to get a unscored token from the Identity Provider.

With this series of steps, the Keystone service provider can now authenticate via an external Identity Provider that can possibly be another Keystone service. Typically, the OpenStack Horizon service is also setup to enable web single sign-on

CHAPTER 6

EVALUATION AND FUTURE WORK

6.1. Introduction

Cloud computing presents significant profits to the company that's seeking advantage in the economy. A lot of companies start working in this area, competition between companies has led to reduce the price with time. Affordable prices have made it easier for company teams to focus on other responsibilities and it also facilitated paying for users only when they consume. Because of these special features and the lowest prices with the pay-as-you-go model, this model has drawn attention from people.

Even though, cloud computing has some security issues, it has a lot of interesting features. The security concern is about how clouding shapes the border of internal security and external threats. We have to test the security services in the cloud in order to check how much companies' data are secured. And we have to understand how much resources are available from providers because they can also be vulnerable to attack which will cause services to stop.

The study discusses the issues which can come from deploying or moving to the cloud and concentrate on OpenStack security authentication and identity. After doing an analysis on OpenStack security issues, we can consider some parts of it safe and others need further research. It is sad to say that OpenStack can't support complicated passwords, Moreover, they are all stored in the plain test file. OpenStack doesn’t provide a control which is needed in the case of accessing sensitive files which contain passwords. And all the data that transmits is not secure because of it is not encrypted and not transmitted securely by secure socket line SSL

The design and implementation of central security system architecture affect system evaluation dramatically. The system as a whole get evaluated in terms of usability, expandability and security architecture, and this obviously assume all the part of our central security system are operational.

6.2. Evaluation of Usability and Scalability

The system security service reliability can be upgraded and improved by getting the security related services the application level, and it will also improve the system's architecture. The central security system offers security services by its means and these services has standardized protocols in their architecture. Which means we can integrate these security services with all the other services of OpenStack. Keystone is the service that provide identity and access control, so we should hide all the Keystone internal services from front end-users. We can improve system reliability, deploy ability and usability by adding a services, like IAM (Identity and Access Management) and SSO (Single-Sign-on) that can offer by a central security system inside OpenStack platform. And also, the presentation of the LCA (Local Certification Authority) server and the integration of PKI with Keystone, give a high scalability to the OpenStack.

6.3. Conclusions and Future Work

The following points are security threats which should be taken into account by the Central Security System.

6.3.1. Authentication and SSO Evaluation

As we use a public network to exchange the SAML ticket between the end user in the internet and application services for single sign-on , we need a mechanism to protect the date from multi-type of threat like man in middle ,unknown replay , Dos attack, so need the application to make a replay depending on the PDP server decision ,this can be done by pre-established of the a trust relation with the user before transfer the data by using PKI model with user digital signing with privet and public need to be predefined so both side can verify the data by the signature of the other side so we provide the authentication and integrity for SMAL ticket and the exchange of ticket become over a secure communication channel using SSL the solve all the threat of hacker .with keystone services.

6.3.2. Authorization Evaluation

After we protect the authentication process, we need to protect the authorization message to prevent illegal permission we protect the message of request-response of PDP service that communicate with other application services through internet theses message are subject to threat like man in middle, modification, and Dos attack. the implementation of secure SSL channel that we proposal give us a way to stop these attacks. And the authenticate of PDP and PEP before and exchange of XAXML message so they transfer on secure channel. a randomly session ID is produce to prevent the replay attack. All the OpenStack component are now connected through PKI model to ensure the that the authentication between XACML message and between PDP and PEP server are secure and trust relationship are trust.

Security Services	Threats			
	Impersonation	Man-in-the-attack	Repudiation	Replay attack
Authentication/SSO	x	x	x	x
Authorization /XACML	x	x	x	x

Table 6.1: System Security Threats Evaluation

AS the cloud computing still new and there are a big innovation to changes the way we pay for technology consuming, therefore there is a lot of issues which should be considered. Some are of technological origin, therefore growing mechanisms for data processing, performance, reliability and management, and on the other hand, other problems still need to be checked like economic and legal issue. Clouding has brought a lot of difficulties along with some big chance for the implementation a solution to the new need of centralization of data center which being used. Thus, researching still working there and a lot of researching will still working in this subject. Swift project has been discussed in this research, is just a beginning which requires some improvements, but it can also give a boost for further explorations. In

the future security scripts should be compatible with the deploying in OpenStack service. We need to make sure that read/write permissions are checked and SSL connections are setup.

We have been focusing on building a standard and high secured structure for cloud platforms so we can protect identity information and provide easy access control. Some of them are mentioned below:

1) Further research cloud be done regarding identity protection, user's anonymity and privacy.

2) System constructions discussed has just be test and deploy for IaaS. Additional analysis with a studying of clouding need to be done for other type SaaS and PaaS to check the applicability of proposed contraction.

3) The general performance of the system should also be assessed in a scalable environment so we can determine throughput and lags. Also, the uptime and downtime of the system can be assessed.

4) To implement SMAL more research should be done.

We need to have a big understand of architectural of implementation especially when deploying Clouding services. It is difficult to address more exhaustive issues wisely but to add some guideline and procedure can help and headline the risk that can exposure the information of the user during the use of clouding.

Table 6.2: Security and Privacy Issues and Precautions

1	Governance	It is concerned with improving organizational practices related to policies, procedures, and standards which they use them to develop application and services provisioning of clouding, Also, the designing with execution and testing, and monitor of development. And the methods and tools which are used to guarantee that the company exercise are done during the system lifecycle.
2	Compliance	This scope focuses on discussing different types of laws and regulations which imposes on organizations some privacy and

		security commitments and probably affects the clouding initiatives, specify the involve of data place, security controls with privacy, and electronics discovery requirements. Checking about organizational requirements and assessing the services which providers offer to ensure that they satisfy organizational needs and to be sure about contract conditions meet the needs .
3	Trust	Allowing the user to learn more about the security and privacy controls which are implemented by the cloud provider and including them in the contracts. Areas of risks have been growing, This created the need to establish a risk management program which can adapt to new changes.
4	Architecture	Cloud providers use a complex technology to provision services which should be understood by users, also, they should understand the implication which technical controls related to security might bring, with respect to the full lifecycle of the system and for all system components.
5	Identity and Access Management	making sure that there are enough safeguards to secure authentication, authorization, And other identity and access management functions.
6	Software Isolation	Understanding virtualization technology and other software isolation technologies which providers use, and to evaluate any risk which may appear.
7	Data Protection	Assessing the compatibility of the providers data management solution with the related Organizational data.
8	Availability	Making sure that if any medium or long-term damage occurred, critical operations should be continued instantly and that all operations can be eventually Reconstructed in a timely and organized manner.
9	Incident Response	Ensuring that the contract provisions and procedure for incident response, which an organization may need, get negotiated and understood.

REFERENCES

1. Amazon (2014) [Online]. About AMS. http://aws.amazon.com/about-aws/ [Accessed on8/07/2014].
2. Armbrust, M. et al. (2010). Comm's of the ACM. A View of Cloud Computing. 53(4). pp. 50-58.
3. Behl, A. & Behl, K. (2012). An analysis of cloud computing security issues. Information and Communication Technologies (WICT), 2012 World Congress on. pp.109-114.
4. Cloud Security Alliance (CSA). Survey by IEEE and Cloud Security Alliance Details *Importance and Urgency of Cloud Computing Security* Standards.http://standards.ieee.org/news/2010/cloudcomp.html (available: March 2010).
5. Cloud Security Alliance (CSA). Top Threats to Cloud Computing V1. https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf (available: March 2010).
6. Cloud, Amazon Elastic Compute (2011). Amazon web services. [Online]. http://dclug.tux.org/200611/AmazonEC2.pdf [Accessed 08/07/2014].
7. D. W. Chadwick, K. Siu, C. Lee, Y. Fouillat, and D. Germonville, Adding federated identity management to OpenStack,Grid Computing, vol. 12, pp.327, Mar. 2014
8. Jensen, J. (2012). "Federated Identity Management Challenges," Availability, Reliability and Security (ARES), 2012 Seventh International Conference on, pp.230-235.
9. K. Jackson and C. Bunch, OpenStack Cloud Computing Cookbook, 2nd ed Packt Publishing, Oct. 2013.
10. Ko, M.N., et al. (2010). "Social-networks connect services." Computer 43(8). pp. 37-43
11. Landau, S. & Moore, T. (2011). "Economic Tussles in Federated Identity Management," Proc. 10th Workshop Economics of Information Security (WEIS 11).

[Online].http://weis2011.econinfosec.org/papers/Economic%20Tussles%20in%20Federated%20Identity%20Management.pdf [Accessed 13/07/2014].

12. Long. Y. et al. (2010). "Attribute mapping for cross-domain access control," Computer and Information Application (ICCIA), 2010 International Conference on. pp. 343-347.
13. Mell, P.; Grance, T. The NIST Definition of Cloud Computing. National Institute of Standards and Technology (2009).
14. Minqi Zhou, Rong Zhang, Wei Xie, Weining Qian, Aoying Zhou: *" Security and Privacy in Cloud Computing*": A Survey, 2010. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5663489. (Accessed date 10/10/2012
15. Oasis. (2014). [Online]. Security Assertion Markup Language (SAML) V2.0 Technical Overview. https://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf. [Accessed 13/03/2014].
16. OpenStack Installation Guide for Red Hat Enterprise Linux 7, CentOS 7,and Fedora 20 - Juno, Manual, OpenStack Foundation, Mar. 2015, accessed on: 19.3.2015. Available at: http://docs.OpenStack.org/juno/installguide/install/yum/content/ch_basic_environment.html.
17. OpenStack Virtual Machine Image Guide, Manual, OpenStackFoundation, Apr. 2015, accessed on: 1.4.2015. Available at:http://docs.OpenStack.org/image-guide/content/index.html.
18. Plummer, D. C.; Bittman, T. J.; Austin, T., Cearley, D. W.; Smith, D. M. Cloud Computing:Defining and Describing an Emerging Phenomenon. Gartner, Inc., 1-9 (2008)
19. Shim, S.S.Y., Bhalla, G. & Pendyala, V. (2005). "Federated identity management," Computer. 38(12), pp.120-122
20. Smith, D. (2008). The challenge of federated identity management [Online], Network Security, 2008(4). pp. 7-9. http://dx.doi.org/10.1016/S1353-4858(08)70051-5.

21. Subashini S.; Kavitha V. A Survey on Security Issues in Service Delivery Models of Cloud Computing. Journal of Network and Computer Applications 34, 1-11 (2011)
22. Weiss, A. (2007). Computing In The Clouds. Networker, 11(4).