Journal of Computer Science 10 (1): 23-33, 2014
ISSN: 1549-3636

doi:10.3844/jcssp.2014.23.33 Published Online 10 (1) 2014 (http://www.thescipub.com/jcs.toc)

CRITICAL REVIEW OF OPENSTACK SECURITY: ISSUES AND WEAKNESSES

Hala Albaroodi, Selvakumar Manickam and Parminder Singh

National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, 11800, Penang, Malaysia

Received 2013-10-09; Revised 2013-10-28; Accepted 2013-11-07

ABSTRACT

The purpose of this study is to examine the state of both cloud computing security in general and OpenStack in particular. Conducting a reassessment of cloud computing security can provide a greater understanding of how cloud computing functions and what types of security issues arise therein. This study is divided into two parts; in the first part, the background of cloud computing and its different deployment models are discussed. This section also describes various security challenges that affect organizations' decisions to adopt cloud computing. In the second part, an overview of the security issues in OpenStack is presented.

Keywords: Security, Cloud Computing, Software as a Services (SaaS), Platform as a Services (PaaS), Infrastructureas a Services (IaaS), OpenStack

1. INTRODUCTION

Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction (Kandukuri *et al*., 2009). Cloud computing utilizes three delivery models in which different types of services are delivered to the end user. The three delivery models are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS), which provide infrastructure resources (Shey *et al*., 2009), application platforms and software as services to the consumer. These delivery models are shown in **Fig. 1**. These service models also place different levels of security requirements upon the cloud environment. IaaS is the foundation of all cloud services, PaaS builds upon IaaS and SaaS, in turn, builds on PaaS. As capabilities are inherited by successive models, so too are information security issues and risks.

There are important differences between each model in terms of merged features, complexity and security. Cloud service providers can provide the basic security architecture; consumers are responsible for implementing and managing the provided security features. Cloud Security Alliance (CSA) and Institute of Electrical and Electronics Engineers (IEEE) report that small-and medium-sized enterprises in the public sector are careful when adopting cloud computing, although those securities are needed together to accelerate cloud adoption on a broad scale and to respond to regulative drivers. Organizations using cloud computing IaaS prefer to examine security and confidentiality threats to their business as critical insensitive applications. In addition knowledge and its management is a foundation for creating competitive advantages in organizations (Mamaghani *et al*., 2011). However, ensuring the security of an enterprise's data in the cloud is difficult, but not impossible, if they supply services such as SaaS, PaaS and IaaS. Each of these services has its own security issues (Kandukuri *et al*., 2009). SaaS service providers ensure that services are available to customers on demand.

The SaaS model provides customers with important benefits, such as improved functional efficiency and reduced costs. SaaS is rapidly emerging as a powerful delivery model capable of meeting the needs of enterprises. Most enterprises are examining the security aspect of the SaaS model with respect to the lack of visibility of data, data storage and security. According to the Forrester study, security is the most common reason for enterprises to adopt SaaS Services (Shey *et al*., 2009). Therefore, enterprise security concerns have emerged as the biggest challenge to the acceptance of SaaS applications in the cloud (Subashini and Kavitha, 2011; Kaur, 2013).

Corresponding Author: Hala Albaroodi, National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, 11800, Penang, Malaysia

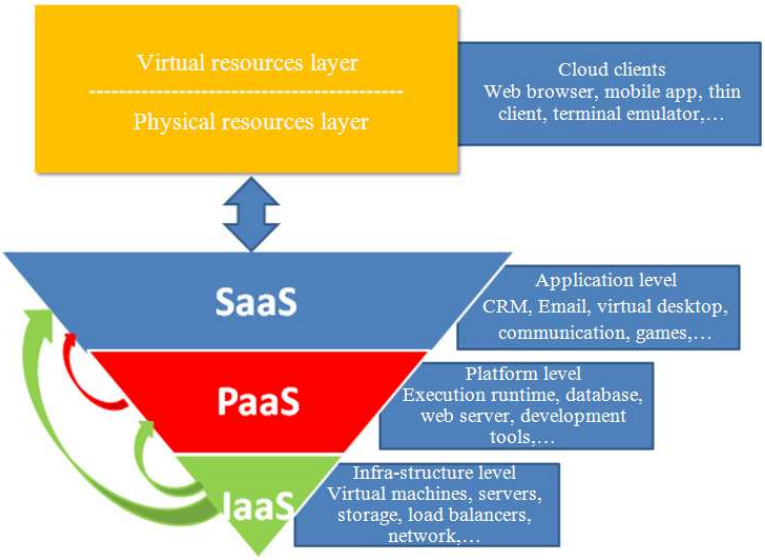

Fig. 1. Cloud computing architecture

One issue that must be addressed directly is customer and vendor concerns about application and data security. There are strong concerns about insider divisions as well as vulnerabilities in application and system availability that could be causing the loss of money and sensitive data. These challenges can discourage enterprises from adopting SaaS applications within the cloud. IaaS completely changes the developers' perceptions. Rather than spending large amounts on infrastructure to build their own data centers or hiring host companies and renting operational staff to initiate the project, developers can go to Amazon Web Services or one of the other IaaS providers to gain access to a virtual server while paying only for the use of resources Amazon, 2013.

Cloud brokers could provide accurate scaling; they could easily expand without worrying about scaling and security (Buyya *et al*., 2009). In brief, IaaS and other related services have enabled start-ups and other businesses to focus on their strengths without worrying about the development and management of infrastructure. IaaS has fully abstracted the hardware underneath it and allows users to use infrastructure as a service without being concerned with the underlying difficulties. The cloud has a binding value hypothesis in terms of cost; although IaaS supplies infrastructure security and applications, activities within the cloud will require higher levels of security to be provided to consumers (Grivas *et al*., 2010).

PaaS provides a level above IaaS and abstracts out everything up to OS, middleware, this offers various development environments in which developers can build their applications without understanding what happens behind the scenes (Grivas *et al*., 2010). Furthermore, the developers offer a service that provides complete software development life-cycle management, from a to z (including planning, design, building applications, deployment, testing and maintenance). However, everything else is hidden from the developer's view.

1.1. Security Issues in SaaS

In SaaS, the client's security measures are dependent on the provider. The provider should ensure that each user's data are hidden from all other users. Security measures must be in place and the client must be confident that the application will be ready for use when needed. In SaaS, the cloud client will often replace old software applications with newer ones. Therefore, the focus lies not upon the portability of applications but rather upon protecting or developing the security functionality of legacy applications and attaining successful data migration (Subashini and Kavitha, 2011; Seccombe *et al*., 2009). Vendors of SaaS services may host applications on their own private servers or use cloud computing IaaS provided by a third-party (e.g., Amazon, Google). The use of cloud computing, along with the pay-and-go approach, helps application service providers reduce the cost of infrastructure services and allows them to focus on providing the best possible service to customers.

In the past decade, computers have grown more popular among enterprises as it services and computing have become commodities. Enterprises today can strategically view data and business processes (such as records, transactions and pricing information) themselves and protect these processes with compliance policies and access control. Furthermore, if the SaaS provider is leveraged as a public cloud computing service, the enterprise's data should be stored together with the data of other unconnected SaaS applications. In addition, the cloud providers should duplicate and store data in multiple locations across different countries for the purpose of maintaining high availability. Most enterprises are familiar with the traditional on-premise model, in which data are stored within the premises of the enterprise and are governed by the enterprise's policies. Thus, many businesses are uncomfortable with the lack of control over and knowledge of how their data are stored and whether it is secure in the SaaS model. There is great concern that problems involving data availability or data breaches could lead to financial and legal liabilities (Anding, 2010). **Figure 2** depicts the layered stack for a classic SaaS vendor as well as important data security issues that span multiple layers. Security components should be considered essential parts of the SaaS application development and data deployment processes, including security, network security, locality, integrity, segregation, access, authentication and authorization, confidentiality, web application security, breaches, virtualization vulnerability, availability, backup, identity management and sign-on processes. The different security issues of SaaS are illustrated in **Fig. 2**.

1.2. Security Issues in PaaS

In PaaS, developers build applications on a computing platform controlled by the provider. In addition, any security issues beneath the application level, such as network and host intrusion prevention, are under the control of the provider, who must offer strong guarantees that the data cannot be accessed by other applications (Subashini and Kavitha, 2011). As a result, PaaS offers more flexibility than SaaS at the expense of customer-ready features. This trade-off extends to security features and capabilities, in that built-in capabilities are less complete, but, simultaneously, there is more flexibility to incorporate additional security. Applications which are sufficiently complex to take advantage of an Enterprise Service Bus (ESB), but which need to secure the ESB directly, benefit from protocols such as Web Service (WS) security (Oracle, 2013). In addition, is very beneficial to use PaaS for Successful Executive Information System Development for Education Domain (Kamaruddin, 2011). The capability to segment ESBS is not present in PaaS environments. Standards should be introduced to regulate the effectiveness of application security programs. Between direct application and security, specific metrics available patch coverage and vulnerability scores. These standards can indicate the quality of application coding. Attention should be paid to how malicious entities are adapting to new cloud application architectures that hide application components from their view. Hackers are likely to attack obvious code, although this is not necessarily restricted to code running in the context of the user. They are likely to attack the infrastructure and perform comprehensive black box testing. Service Oriented Architecture (SOA) applications, which are increasingly being distributed within the cloud (Cao *et al*., 2009).

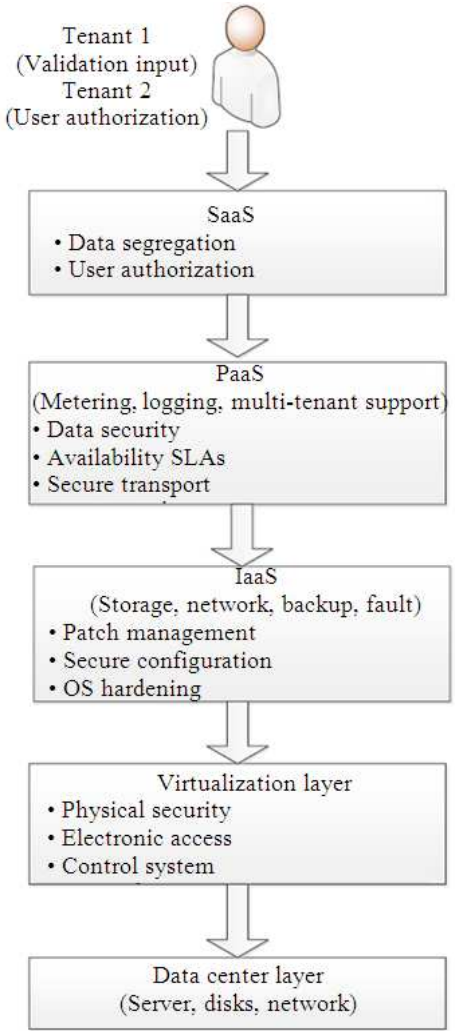

Fig. 2. Security elements in the stack layers (SaaS, PaaS, IaaS)

1.3. Security Issues in IaaS

In IaaS, the developer has the best control over security, as long there is no security hole in the Virtualization Manager (VM). While in theory virtual machines might be able to address these issues as they arise, there are many security problems in practice. An additional factor is the reliability of the data stored in the provider's hardware. Due to the growing virtualization of the information society, enabling owners to maintain control over their data regardless of its physical location will become a topic of extreme interest. To obtain maximum trust and security on a cloud resource, several techniques need to be practiced (Descher *et al*., 2009). The security obligations of both the provider and the consumer vary greatly between cloud service models. Amazon's Elastic Compute Cloud (EC2) infrastructure presents an example in which the vendor's responsibility for security extends only to the hypervisor. This means that they can only address security controls such as virtualization security, physical security and environmental security. The consumer is responsible for the security controls corresponding to the system, including the applications, OS and data (Seccombe *et al*., 2009). IaaS gives rise to security issues whose severity depends on the cloud deployment model through which the services are delivered. The physical security of the infrastructure is extremely important; disaster management plans are necessary to prevent damage, either natural or intentional, to the infrastructure.

Infrastructure includes not only the hardware in which data are computed and stored but also the paths by which it is obtained or transmitted. In a standard cloud environment, data will be transmitted from source to destination through numerous third-party infrastructure devices (Ristenpart *et al*., 2009). However, the complexities arising from the various service deployment models of IaaS are illustrated in **Table 1**.

Cloud architectures are built upon underlying technology. A cloud built over the Internet inherits all of the internet's inherent security risks. The foundations of cloud technology force consumers and providers with different physical locations to virtually access resources over the Internet (Prautzsch and Graves, 2011; Sehgal *et al*., 2011). Even if an enormous amount of security is established in the cloud, data must still be transmitted via the underlying internet technology. Therefore, the security concerns threatening the internet also threaten the cloud. However, the risks to cloud computing are especially great. The vulnerability consideration and asset value of the resources and asset value of the resources and their nature of them settling together. Cloud systems still use normal internet protocols and security standards but require greater levels of security. Although secure protocols and encryption cater to current needs to a certain extent, they are not context oriented (Mell and Grance, 2009). A strong set of policies and protocols is necessary to secure data transmission within the cloud. Concerns regarding the intrusion of external non-users into cloud databases should also be considered. Standards should be established to construct a secure, private and isolated cloud environment in the internet that is capable of avoiding attacks by cyber criminals.

The focus of this study is to inspect and evaluate the possibility of implementing cloud computing using OSS technology and, in particular, OpenStack, the pioneer product of OSS. Moreover, this study contributes to the swift project, which is part of the OpenStack project, by strengthening its security arsenal. Swift is the OpenStack object storage project, the purpose of which is to offer cloud storage software in which users can store and retrieve large amounts of data in virtual containers.

1.4. OpenStack

This section gives an overview of OpenStack, its components and the nature of its security mechanisms.

1.5. Overview on OpenStack

In October 2010, the initial "austin" release of OpenStack was published. It consisted of only two projects: Object storage and compute. Object storage was ready for production and compute was intended for testing. In February 2010, an updated version of OpenStack was released under the name "bexar". With bexar's release came a new component, called "OpenStack image service". In addition to releasing the new project, the development teams also made some enhancements to the previously announced projects. For example, the object storage (swift) project introduced a means of authorizing and authenticating users, known as "swAuth". The third release, code named "cactus", announced the addition of two features to the object storage project: The option to serve static content and the ability to perform content checksum validation during get object actions. At the same time, OpenStack was performing quick enhancements on and providing additional support for virtualization technology. the fourth and, at the time of this writing, latest OpenStack release, "diablo", was announced in September 2011, at

which point the OpenStack community included over 1500 people and 87 companies. At this time, the number of product deployments began to increase. Although the project teams improved scalability, availability and stability, many security concerns were still pending. OpenStack is open-source software for building private and public clouds (Wen *et al*., 2012; Beloglazov *et al*., 2012a). OpenStack consists of three main projects. The relationships between these projects are depicted in **Fig. 3**.

The core services are compute, storage, networking and dashboard, whereas the auxiliary services are identity and image:

- Nasa developed OpenStack compute (NOVA), which provides and manages networks of virtual machines. Public cloud service providers offer Infrastructure as a Service (IaaS), while private clouds offer services within Organizations. Tools, such as Hadoop and High-Performance Computing (HPC) applications are examples of services with which OpenStack compute is compatible (OpenStack, 2013). Following is a partial list of OpenStack compute
- Commodity servers, including CPU, memory, disk and network interfaces, can be managed
- Local Area Networks (LAN) are Organized, including flat, flat DHCP, VLAN DHCP, ipv4 and ipv6 networks works
- Virtual machine image management tools include importing, sharing and querying
- Floating IP addresses can be assigned (and re-assigned) to VMs
- VM image caching on compute nodes enhances the efficiency of VMs

Rackspace developed and contributed to OpenStack object storage (swift and cinder). OpenStack storage saves objects and blocks for servers and applications. Object storage, implemented via a distributed storage system, is designed to house static data, such as virtual machine images, backups and archives. These objects and files are saved in disk drives throughout the OpenStack cloud.

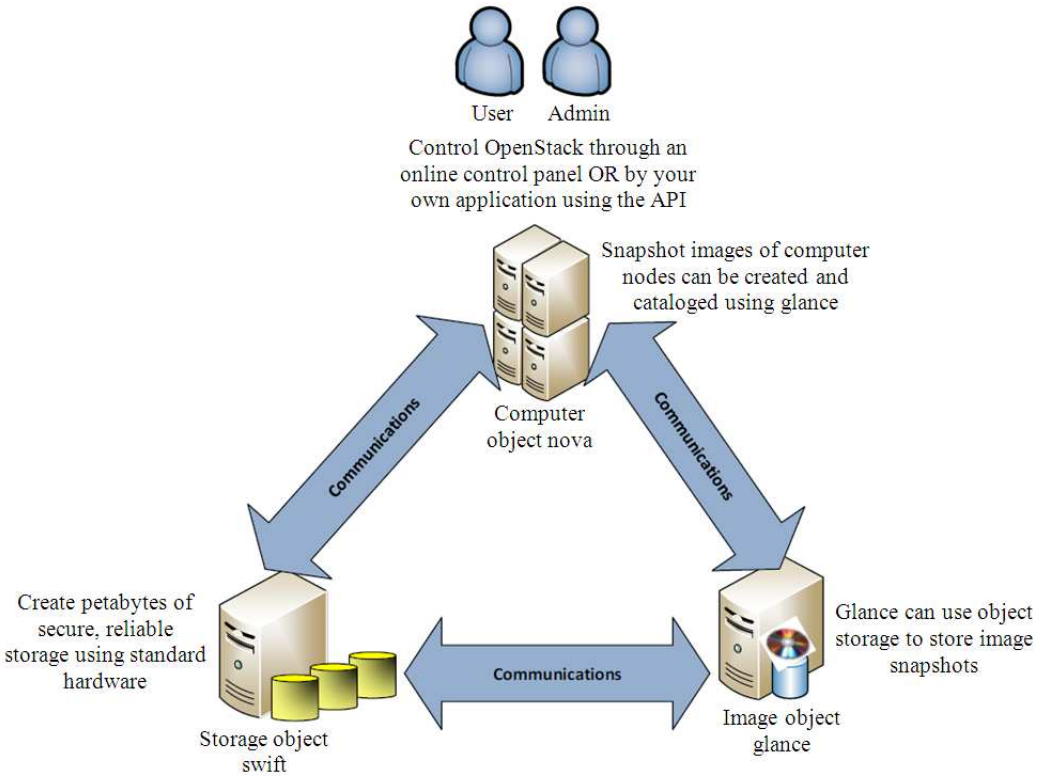

Fig. 3. The communication between the project elements

Hence, scalability and repeatability are achieved. OpenStack likewise provides constant block-level storage devices for computing tasks that require high performance storage, which is often required by databases, expandable file systems, or servers that access raw block-level storage (Baset, 2012). The features of OpenStack storage are as:

- Commodity hard drives reduce the storage cost per byte
- It is capable of self-healing because data are copied to different sectors of the cloud; thus, the storage system becomes highly redundant and reliable
- It can store data on a very large scale; multiple petabytes of data and billions of individual objects can be stored
- Amazon s3 (elastic block storage) API is supported
- Utilities enable the management of account, container and storage monitoring features

OpenStack image repository (glance). This component enables discovery, registration and delivery for disk and server images. Base image templates can be created for use in new instances users and administrators can also construct and store snapshots of images, which can be saved in raw, VHD (Hyper-v), VDI (VirtualBox), qcow2 (Qemu/KVM), VMDK (VMware) and OVF (VMware, others) formats (Baset, 2012).

OpenStack networking (quantum). OpenStack networking is an API-driven system for cloud networks and IP addresses. Its features include the following:

- Static, DHCP and floating IP addresses are managed
- It supports several networking models, such as flat networks and VLANs
- It creates and manages users'
- It supports SDN technology (i.e., openflow)

OpenFlow (SDN). A systems architecture, SDN stands for "software-defined networking". Although SDN has become widely recognized only recently, its defining architecture has been extensively used. A conventional network device contains hardware and software. Users, however, could not independently define a network because of the lack of an Application Programming Interface (API).

Hence, OpenFlow was introduced. This technology is capable of enabling SDN. It is not a networking method that provides specific functions, such as l2 (layer 2) switching or IP routing. Instead, OpenFlow is simply an interface that can be installed on a network device. Promoting the use and standardisation of SDN, the Open Networking Foundation (ONF) defined the specifications of SDN (Mell and Grance, 2009), including the components and basic functions of switches and the OpenFlow protocol for managing OpenFlow switches from remote controllers. OpenFlow accesses and manages the API controlling the hardware, although information concerning the latter is not disclosed by the device manufacturers and enables users to independently manage networks. The network framework allows various devices to be incorporated within the cloud, including intrusion detection systems, load balancers and firewalls.

OpenStack dashboard (Horizon). OpenStack dashboard enables administrators and users to provide, manage and control cloud computation, storage and networking resources. Dashboard is used to create users and projects, assign users to projects and decrease the resources required for such projects. It also provides and controls resources allocated to projects. The OpenStack dashboard is an extensible web-based application (Crago *et al*., 2011).

OpenStack identity (keystone). OpenStack identity maintains a database of users and provides authentication services. A common authentication system is provided throughout the cloud and can be integrated with third-party, back-end directory services (i.e., lightweight directory access protocol or ldap). It supports multiple verification systems, such as the standard username and password, token-based systems and web services such as Amazon. OpenStack identity allows cloud administrators to establish policies across users and systems, create users and tenants and grant permission to compute, store and network resources (Beloglazov *et al*., 2012b). All of the core services are illustrated in **Fig. 4**.

1.6. Security in OpenStack

We have found several flaws in OpenStack; these threats may be addressed in the current releases of OpenStack (Slipetskyy, 2011; Cigoj and Klobucar, 2012):

- Users cannot reset their passwords on horizon; regular users can only have their passwords reset by the administrator within the horizon interface. We do not currently know how this flaw will impact
- The administrator of a project on horizon is automatically made the administrator of the whole

system. OpenStack utilizes the concept of projects and tenants to group people into logical units for cloud computing. However, the administrator of a single project is granted managerial rights to all projects, not merely the project at hand, by the interface. The administrator's privileges, including the creation of new users and projects, have the potential to change other projects, remove items

- Cleartext is used in the network API. OpenStackapi endpoints encourage the use of cleartext and no SSL/TLS support is available right now. This allows for easy man-in-the-middle attacks and even "sniffing" passwords over the wire can be trivial
- No authentication in the client-server system. It appears that any host with access to the db and to the AMQP system can act as a compute node and launch VMs
- Usernames and passwords. Passwords and usernames that are used for accessing images will be stored in Cleartext in the db and in external storage. When glance stores images on swift, for example, the username and password of the swift account will be stored as Cleartext in the db together with the URL of the swift object. This could potentially allow the information of any swift user to be accessed and read from the db. This storage of information is unnecessary because the username and password are already stored in the glance configuration file

The problems discussed in this section will be used as the basis for studying cloud security solutions in subsequent sections. While studying the security issues of cloud computing in the previous section, we discovered which issues are often discussed in relation to identity and access management. In this section, we discuss identity and access management.

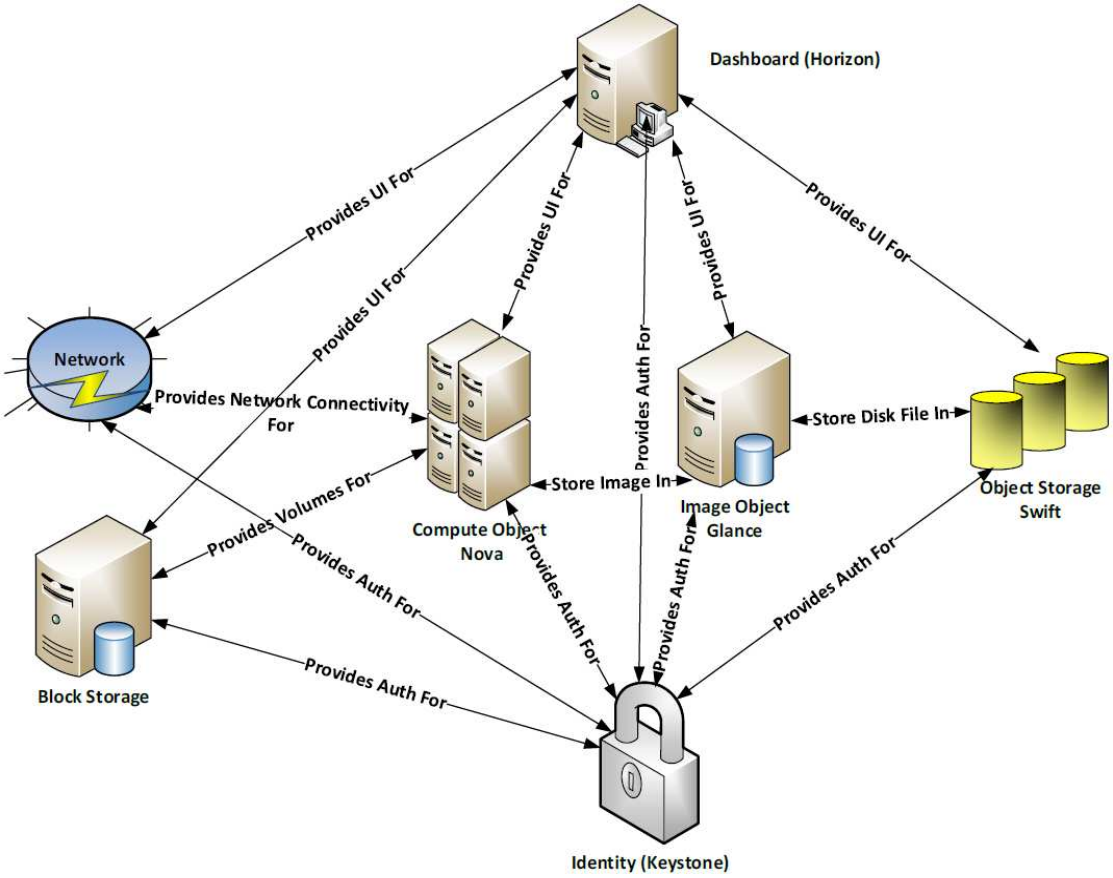

Fig. 4. The communication between the project elements

1.7. Identity

User provisioning is the process of registering a new user with a given system and user de-provisioning is the process of removing a user from the system. OpenStack object storage "swift" offers significant automation of user data management tasks by using authentication/authorization systems referred to as "tempAuth" and "swAuth". The difference between "tempAuth" and "swAuth" lies in the back-end storage of user data. TempAuth uses a configuration file in which user data are saved as plain text. on the other hand, swAuth is meant to be a "scalable authentication and authorization system that uses swift itself in a backing store" Swift/overview, 2013. A swift account is created on a swift cluster and user information is stored in "json-encoded" text files, which are also swift objects. Both swAuth and tempAuth allow on-demand user provisioning and de-provisioning, which is in accordance with industry standards. The characteristics of user management are based on OpenStack object storage "swift" (Cigoj and Klobucar, 2012). The following characteristics are present:

- Users are not given administrative power over any other users
- Provider Admins have admin agreements with all accounts but cannot add other provider admins
- Super admins are powerful users who are able to perform all user management procedures, including adding provider admins

1.8. Authentication

TempAuth and swAuth often use a username and password for the authentication process. When authentication is successfully performed, the user receives a token that will identify him to the system for a period of time. The provided token has a configurable expiration time, the default value of which is set to 4-6 h. All cloud security documents must, allow authentication by accepting confirmations in SAML format; however, this feature is not yet available in OpenStack (Khan *et al*., 2011).

1.9. Strength of Password

Because all OpenStack projects use a password and username system to authenticate users, password strength requirements should receive greater scrutiny. The "Electronic Authentication Guideline" created by NIST supplies guidelines for helping users avoid choosing bad passwords, such as checking a password against a dictionary of commonly used passwords, instituting a minimum password length and requiring the use of certain types of character (such as upper-case, lower-case and non-alphabetic) (Cigoj and Klobucar, 2012; Mell and Grance, 2011). Unfortunately, none of these requirements (dictionary checks, minimum password lengths, or special character requirements) exist within OpenStack, allowing users to register with short passwords containing no special characters.

1.10. Storage of Password

Password storage poses a well-known problem to all information systems using password authentication. A common practice in information security is to require the administrator to guarantee that passwords are encrypted, rather than being stored as cleartext. It is also important to limit access to the location where passwords are stored.

As was mentioned previously, tempAuth stores usernames and passwords in a configuration file in which all passwords are recorded in plain text format. The location of super user credentials is also saved to the same file, as shown in **Fig. 4**. By default, each user in the system possesses reading access to this file. Such access enables system users to gain the passwords of other users and easily obtain access to their accounts. Most of the developers never considered tempAuth to be suitable for production deployment.

SwAuth uses a special configuration file where super admin passwords are saved, unlike tempAuth and swAuth possesses properly configured access permissions for files containing secure password data. The only security threat that arises in swAuth is that the passwords within these files are stored in clear-text. Therefore, an internal attacker could gain access to super user accounts within the system and thus be able to learn user passwords. OpenStack should consider hashing passwords before saving them to the password file (Jackson, 2012; Laszewski *et al*., 2012).

In conclusion, both tempAuth and swAuth lack appropriate password protections. Both authentication systems should implement the following recommendation, taken from NIST's "Electronic Authentication Guideline": Saved passwords and/or usernames should be salted and then hashed with an approved algorithm, so that the techniques used to conduct dictionary or weakness-based attacks on a stolen password file would not be useful for attacking a similar password file. A comparison of the two authentication systems is given in **Table 2**.

Table 1. Cloud deployment models due to the complexity of IaaS

	Public	Private/community	Hybrid
Infrastructure management	Third-party provider	Organization or third-party provider	Both organization and third-party provider
Infrastructure owner	Third-party provider	Organization or third-party provider	Both organization and third-party provider
Infrastructure Location	Off-premise	On-premise or off-premise	Both on-premise and off-premise
Access and conception	Untrusted	Trusted	Trusted and untrusted

Table 2. Comparison of tempAuth and swAuth

	Admin (unprotect password)	User (unprotect password)	Access to configure files	Used in Diablo release	Admin has access to all user data
swAuth	/etc/swift/proxy-server.conf	Hala encoded text files	Owner of the file	Pluggable	Yes
tempAuth	/etc/swi ft/proxy-server.conf	/etc/swift/proxy-server.conf	Everyone	Built-in	Yes

Table 3. Security issues

Security issues	Implications of Security
Trust	This is interrelated to the designated deployment modal because the control of the data and applications is directly supervised by the strict control of the owner
Availability	The capacity of a system to operate upon the demands of a certified entity. This notion implies that the system should be able to function even in the presence of authorities that disobey the regulations. Furthermore, the system must also maintain the capacity to operate even in the existence of a security breach
Integrity	Resources can only be reformed by approved individuals and through official procedures.
● Software	The diverse resources include data, software and hardware
● Data (Authentication, Authorization and Access control AAA)	
Confidentiality	Data in cloud computing are more vulnerable because of the increase in the number
● Software	of individuals, devices and applications that use cloud computing which will in turn
● Data	increase the number of access points. Consequently, authorized individuals and systems are the only entities that are allowed to access the protected data
Privacy	An individual's need to govern the entree to his/her personal information

Recently, all components of "Essex", the latest release of OpenStack, support Identity Service (Keystone), which introduces a more secure way of storing passwords in the database. Customers must be identified by Keystone before they are allowed to use any of the cloud services, which guarantees a unique point of entry. Keystone encrypts usernames and passwords and provides each user with a unique token that enables access to the services for which they are authorized. So far, Identity Service provides the most complete security solution available to Open Source clouds.

1.11. Authentication Tokens

Authentication tokens play similar roles as identifiers for web applications. An API, such as an OpenStack service, is used to authenticate a user. Successful authentication generates a token that is used to authorize service requests. The password and username are given as input to the API interface. When authentication succeeds, the resulting feedback includes an authentication token and service catalogue. Note that tokens remain valid for 12 h. Issued tokens become invalid in two situations:

- If the token is expired
- If the token has been canceled

It is important that the authentication be executed over a secure channel, such as Transport Layer Security (TLS); otherwise, an attacker could obtain a user token by executing a man-in-the-middle-attack and remove the user who received the token from the

authentication system. However, Rostyslav Slipetskyy has subjected the algorithms that are imported for token generation to a more detailed examination. The algorithm imitates the approach used to generate Universally Unique ID (UUID) and utilizes a solid source of randomness that has no known disadvantages and thus is considered to be secure by (Slipetskyy, 2011).

1.12. Susceptibility of Authentication Data

The transfer of OpenStack authentication data from one server to another is not safe. SwAuth has security issues that allow provider admins to view the data belonging to all users who are managed by the admin account. Malicious users are also able to gain access other users' passwords (Lonea *et al*., 2012; Dlamini *et al*., 2012).

1.13. Maliciousof Data

Most cloud providers do not encrypt data before saving it to a cluster. In fact, OpenStack does not provide any data encryption at all; thus, users would need to encrypt their data before uploading it and manage their encryption keys themselves.

It may be difficult to track security issues in cloud computing environments. Therefore, the primary aim of this study is to highlight the implications of the major security issues. **Table 3** provides a summary of these security issues, which are divided into five categories and listed with their implications.

2. CONCLUSION

Cloud computing provides an important benefit to companies looking for an advantage in today's economy. Many providers are offering cloud computing services; this competition will lead to increasingly affordable prices over time. Lower prices enable businesses to use staff for other tasks and allow them to consume resources more efficiently by paying for services only as they are needed. These features, supported by an attractive and economical pay-as-you-go approach, have led to growing support for this model.

One important threat posed by cloud computing is the obscuring of boundaries between internal and external security concerns. To understand how well companies' data are kept safe, security services in the cloud must be closely studied. In second level will be the availability, as providers can be victims of attacks that stop the running of their operations.

This study discusses issues that arise with the deployment model of cloud computing; in particular, this study focuses on OpenStack security issues and threats. Certain parts of OpenStack are considered secure while others need to be improved. OpenStack does not support minimum password complexity requirements and passwords are stored in plain text format. There are no controls to regulate access to sensitive files, including those containing passwords. Information transferred within the cloud is not protected through the use of file encryption techniques.

3. REFERENCES

Anding, M., 2010. SaaS: A Love-Hate Relationship for Enterprise Software Vendors. In: Software-as-a-Service: Anbieterstrategien, Kundenbedürfnisse und Wertschöpfungsstrukturen, Benlian, A., T. Hess and P. Buxmann (Eds.), Springer DE, Wiesbaden, ISBN-10: 383498731X, pp: 43-56.

Baset, S.A., 2012. Open source cloud technologies. Proceedings of the 3rd ACM Symposium on Cloud Computing, Oct. 14-17, San Jose, CA, USA., pp: 1-140. DOI: 10.1145/2391229.2391257

Beloglazov, A., S.F. Piraghaj, M. Alrokayan and R. Buyya, 2012a. A Step-by-Step Guide to Deploying OpenStack on CentOS Using the KVM Hypervisor and GlusterFS Distributed File System.

Beloglazov, A., S.F. Piraghaj, M. Alrokayan and R. Buyya, 2012b. Deploying open-stack on centos using the KVM hypervisor and glusterFS distributed file system. University of Melbourne.

Buyya, R., R. Ranjan and R.N. Calheiros, 2009. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. Proceedings of the International Conference on High Performance Computing and Simulation, Jun. 21-24, Leipzig, Germany, pp: 1-11.

Cao, B.Q., B. Li and Q.M. Xia, 2009. A service-oriented QoS-assured and multi-agent cloud computing architecture. Cloud Comput., 5931: 644-649. DOI: 10.1007/978-3-642-10665-1_66

Cigoj, P. and T. Klobucar, 2012. Cloud security and OpenStack. Proceedings of the 1st International Conference on CLoud Assisted ServiceS, (AS' 12), pp: 20-106.

Crago, S., K. Dunn, P. Eads, L. Hochstein and D.I. Kang *et al*., 2011. Heterogeneous cloud computing. Proceedings of the IEEE International Conference on, Cluster Computing, Sept. 26-30, IEEE Xplore Press, Austin, TX, pp: 378-385. DOI: 10.1109/CLUSTER.2011.49

Descher, M., P. Masser, T. Feilhauer, A.M. Tjoa and D. Huemer, 2009. Retaining data control to the client in infrastructure clouds. Proceedings of the International Conference on, Availability, Reliability and Security, Mar. 16-19, IEEE Xplore Press, Fukuoka, pp: 9-16. DOI: 10.1109/ARES.2009.78

Dlamini, M., H. Venter, J. Eloff and Y. Mitha, 2012. Authentication in the Cloud: A risk-based approach. University of Pretoria.

Grivas, S.G., T.U. Kumar and H. Wache, 2010. Cloud broker: Bringing intelligence into the cloud. Proceedings of the 3rd International Conference on, Cloud Computing, Jul. 5-10, IEEE Xplore Press, Miami, FL, pp: 544-545. DOI: 10.1109/CLOUD.2010.48

Jackson, K., 2012. OpenStack Cloud Computing Cookbook. 1st Edn., Packt Publishing Ltd, Birmingham, ISBN-10: 1849517339, pp: 318.

Kamaruddin, M.A.R.R., 2011. A framework of successful executive information system development for education domain. Am. J. Applied Sci., 8: 997-1003. DOI: 10.3844/ajassp.2011.997.1003

Kandukuri, B.R., V.R. Paturi and A. Rakshit, 2009. Cloud security issues. Proceedings of the IEEE International Conference on Services Computing, Sept. 21-25, IEEE Xplore Press, Bangalore, pp: 517-520. DOI: 10.1109/SCC.2009.84

Kaur, S., 2013. Pushing frontiers with the first lady of emerging technologies-How to Secure Our Bluetooth Insecure World. IETE Technical Rev., 30: 95-101. DOI: 10.4103/0256-4602.110547

Khan, R.H., J. Ylitalo and A.S. Ahmed, 2011. OpenID authentication as a service in OpenStack. Proceedings of the7th International Conference on, Information Assurance and Security (IAS), Dec. 5-8, IEEE Xplore Press, Melaka, pp: 372-377. DOI: 10.1109/ISIAS.2011.6122782

Laszewski, G.V., J. Diaz, F. Wang and G.C. Fox, 2012. Comparison of multiple cloud frameworks. Proceedings of the 5th International Conference on, Cloud Computing, Jun. 24-29, IEEE Xplore Press, Honolulu, HI., pp: 734-741. DOI: 10.1109/CLOUD.2012.104

Lonea, A.M., D.E. Popescu and O. Prostean, 2012. A survey of management interfaces for eucalyptus cloud. Proceedings of the 7th IEEE International Symposium on Applied Computational Intelligence and Informatics, May 24-26, IEEE Xplore Press, Timisoara, pp: 261-266. DOI: 10.1109/SACI.2012.6250013

Mamaghani, N.D., R. Samizadeh and F. Saghafi, 2011. Evaluating the readiness of Iranian research centers in knowledge management. Am. J. Econ. Bus. Admin, 3, 203-212. DOI: 10.3844/ajebasp.2011.203.212

Mell, P. and T. Grance, 2009. Draft NIST working definition of cloud computing. National Institute of Standards and Technology.

Mell, P. and T. Grance, 2011. The NIST definition of cloud computing (draft). NIST special publication, Recommendations of the National Institute of Standards and Technology.

OpenStack, 2013. Compute administration manual-cactus.

Oracle, 2013. WiringthroughanEnterpriseServiceBus.

Prautzsch, F. and S. Graves, 2011. Commercial SATCOM in support of protected connectivity for the Warfighter and first responder. Proceedings of the Military Communications Conference, Nov. 7-10, IEEE Xplore Press, Baltimore, MD, pp: 2296-2301. DOI: 10.1109/MILCOM.2011.6127664

Ristenpart, T., E. Tromer, H. Shacham and S. Savage, 2009. Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. Proceedings of the 16th ACM Conference on Computer and Communications Security, Nov. 09-13, Chicago, IL, USA, pp: 199-212. DOI: 10.1145/1653662.1653687

Seccombe, A., A. Hutton, A. Meisel, A. Windel and A. Licciardi *et al*., 2009. Security guidance for critical areas of focus in cloud computing. Cloud Security Alliance.

Sehgal, N.K., S. Sohoni, Y. Xiong, D. Fritz and W. Mulia *et al*., 2011. A cross section of the issues and research activities related to both information security and cloud computing. IETE Technical Rev., 28: 279- 291.

Shey, H., R. Wang, J.P. Garbini and E. Daley, 2009. The State of Enterprise Software: 2009. Forrester Research, Inc.

Slipetskyy, R., 2011. Security issues in OpenStack. Mrs., Thesis, Norwegian University of Science and Technology.

Subashini, S. and V. Kavitha, 2011. A survey on security issues in service delivery models of cloud computing. J. Netw. Comput. Appli., 34: 1-11. DOI: 10.1016/j.jnca.2010.07.006

Wen, X., G. Gu, Q. Li, Y. Gao and X. Zhang, 2012. Comparison of open-source cloud management platforms: OpenStack and OpenNebula. Proceedings of the 9th International Conference on Fuzzy Systems and Knowledge Discovery, May 29-31, IEEE Xplore Press, Sichuan, pp: 2457-2461. DOI: 10.1109/FSKD.2012.6234218