Patterns for Secure Cloud IaaS (Infrastructure as a Service)

Eduardo B. Fernandez[1], Hironori Washizaki[2], Nobukazu Yoshioka[3]

[1]Florida Atlantic University

[2]Waseda University

[3]GRACE Center, National Institute of Informatics

fernande@fau.edu, washizaki@waseda.jp, nobukazu@nii.ac.jp

Abstract. *We present patterns that describe the security architectures of three hierarchic levels of cloud infrastructure. A Secure Cloud IaaS pattern describes the security mechanisms used in most commercial clouds. The Secure Open IaaS describes the security features of open source clouds, while the Secure OpenStack pattern describes the security mechanisms used in this de facto standard. The Secure IaaS pattern can be considered an Abstract Security Pattern (ASP), while the Secure Iaas is a concrete security pattern, and the Secure OpenStack is a technology.*

Categories and Subject Descriptors

•Software Architecture → Patterns

General Terms

Design

Key Words and Phrases

Cloud security, IaaS, security patterns, OpenStack

1. Introduction

In a cloud system the IaaS layer controls the use of the hardware resources and virtualizes these resources; it has then a fundamental role in providing security to the complete system. We presented a pattern for the IaaS layer (Hashizume 2012), but we did not emphasize its security functions. In (Fernandez, Monge, and Hashizume 2015) we added security to IaaS by analyzing its use cases and postulating the ways they could be attacked as well as their corresponding defenses to produce a Security Reference Architecture (SRA). We then verified that our model contained all the security mechanisms found on commercial clouds. We present now a Secure Cloud IaaS pattern, based on finding common security functions in the IaaS layers of commercial cloud systems as described in their reference architectures or textual descriptions. This analysis complements our previous study of a cloud SRA and helps validate that architecture.

Open source clouds are an important special case of clouds. Open systems have a variety of concrete architectures that use the open source definition as a guideline. We present the Secure Open IaaS pattern, which describes the common security mechanisms used in open versions of IaaS. Open versions of Iaas include OpenStack, Nebula, Eucalyptus, and others. We see them as lower-level (technology-oriented) patterns in a Secure Solution Frame (SSF) hierarchy (Uzunov, Fernandez, Falkner, 2015). SSFs are solution structures that encapsulate and organize security patterns; they realize a set of security requirements. SSFs define horizontal and vertical pattern structures. Horizontal structures, Security Pattern Families (SPFs), correspond to peer-related patterns that complement each other and define different facets of a security policy, while vertical structures are hierarchies of patterns specialized going from ASPs to practical implementations. We show here three levels of a SSF that has IaaS as its root. Pattern families always end with an implementation-specific pattern, called a technology pattern. SSFs can facilitate the work of designers by collecting together all the relevant patterns to realize some security requirements, guiding the designer from an abstract conceptual level to a concrete implementation-oriented level. The idea of combining technologies and patterns comes from (S. Mahdavi-Hezavehi et al., 2011). Figure 1 shows this hierarchy in a pattern diagram. The left hierarchy describes IaaS systems without security while the right hierarchy (in color) is a SSF that shows the three patterns described here plus a possible concrete OpenStack IaaS technology pattern.

These patterns are part of a cloud ecosystem, which describes the interdependencies and connections of subsystems that describe other systems related to a cloud hub in the form of patterns (E.B. Fernandez, N. Yoshioka, and H. Washizaki 2015). Our objective is not to analyze their level of security, several papers have done that, e.g. (Ristov and Gusev 2013), but to describe their common security practices.

Our audience are cloud designers, administrators, application developers, and possibly users. Each pattern repeats some aspects of its higher-level pattern to make it self-contained. Section 2 describes the Secure IaaS pattern, while Section 3 presents the Secure Open IaaS pattern. Section 4 is the Secure OpenStack pattern. We end with some conclusions in Section 5.

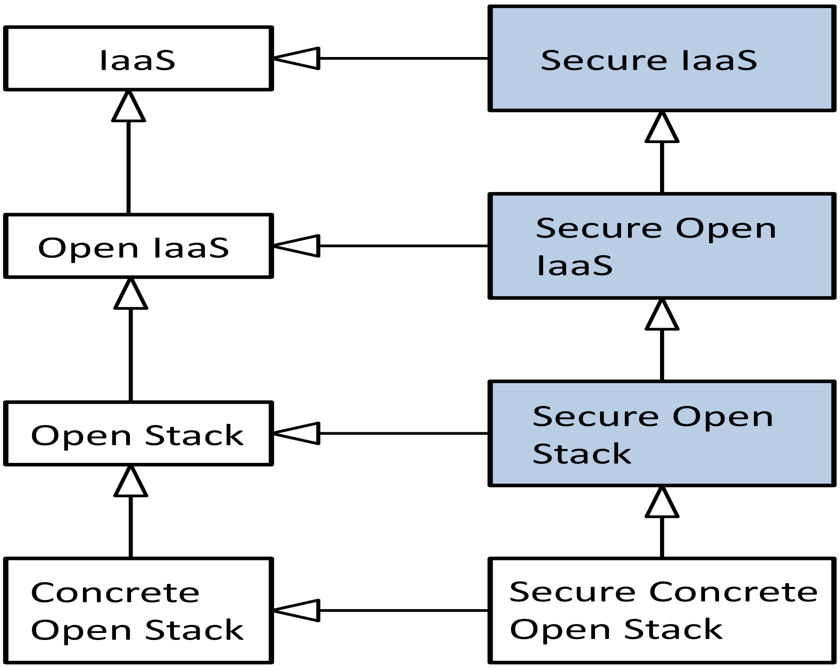

Figure 1. Pattern diagram to relate the patterns in this paper

2. Secure IaaS

Intent

The Secure Infrastructure-as-a-Service pattern describes the architecture required for the sharing of distributed virtualized computational resources such as servers, storage, and networks, including a set of security services.

Context

A cloud system is an attempt to improve the utilization of resources and provide convenient and cost-effective access to computational resources in the Internet. These resources are provided to the users as a form of a virtualized service called Infrastructure-as-a-Service (IaaS). Figure 2 shows its core architecture (Hashizume et al., 2012). The *Cloud Controller* is the main component which processes requests from a *Party*. A Party can be an institution or a user (customers and administrators). A Party can have one or more *Accounts*. The Cloud Controller coordinates a collection of services such as VM scheduling, authentication, VM monitoring and management. When a Cloud Controller receives a request from the party to create a VM, it requests its corresponding *Cluster Controllers* to provide a list their free resources. With this information, the Cloud Controller can choose which cluster will host the requested virtual machine. A Cluster Controller is composed of a collection of *Node Controllers*, which consist of a pool of *Servers* that host *Virtual Machine (VM)* instances. The Cluster Controller handles the state information of its Node Controllers, and schedules incoming requests to run instances. A Node Controller controls the execution, monitoring, and termination of the VMs through a *Virtual Machine Monitor (VMM)* which is the one responsible to run VM instances. The Cloud Controller retrieves and stores user data and *Virtual Machine Images (VMI)*. The *Virtual Machine Image Repository* contains a collection of Virtual Machine Images that are used to instantiate VMs. The *Dynamic Host Configuration Protocol (DHCP)* server assigns a MAC/IP

(Media Access Control/Internet Protocol) pair address for each VM through the Cloud Controller, and requests the *Domain Name System (DNS)* server to translate domain names into IP addresses in order to locate cloud resources.

Problem

The model of Figure 2 does not provide any security to its users. The users share the available resources but there is no separation of their execution processes. How can they execute their applications in a secure way?

The required solution must satisfy the following forces:

Transparency - The security services should be transparent to the users. Users should be able to use the provider's services without understanding their infrastructure.

Flexibility - Different degrees of security may be demanded by users. We should be able to accommodate them.

Contractual security—The required degree of security should be specified in a service contract.

Manageability - In order to manage a large number of users and services, these services should be easy to deploy and manage.

Monitoring—There should be a way to monitor the degree of security provided. Otherwise, the consumer cannot be sure it is getting the required level of service

Certification—Some important or critical services should be able to be certified with respect to security.

Testability—It should be convenient to test the security of the cloud services. Otherwise, it is hard to enforce service contracts.

Shared resources - Many users should be able to share resources in order to increase the amount of resource utilization and thus reduce costs. This sharing should be controlled,

Isolation - Different user execution instances (VMs) should be isolated from each other.

Shared Non-functional requirements provision (NFRs) – Sharing of the costs to provide NFRs is necessary to allow providers to offer a higher level of NFRs.

Solution

IaaS must incorporate mechanisms that provide all the security attributes needed to support application execution: confidentiality, integrity, availability, and non-repudiation. It must also have a way to protect the information needed to apply these controls and the security mechanisms themselves. Providing these attributes imply having a set of security mechanisms that are determined by experience or through a systematic methodology as we showed in (Fernandez, Monge, Hashizume 2015); they typically include:

Authentication—A type of multifactor authentication must be provided.

Authorization—Some of Role-Based Access Control (RBAC) should be provided.

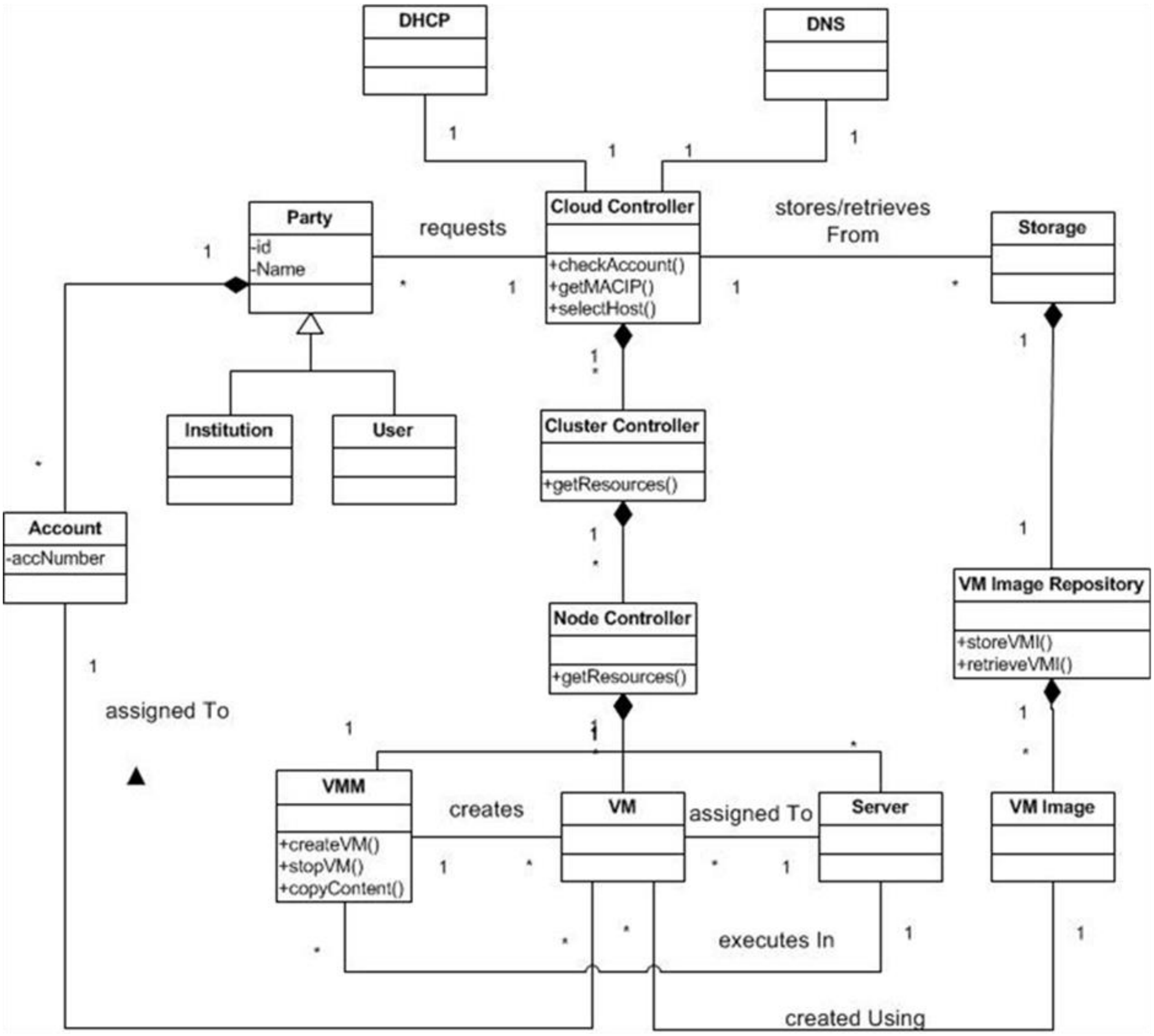

Figure 2: Class Diagram for a Cloud Infrastructure

Logging—We should have secure, tamper-free logging and tools for auditing.

Cryptography—There should be a variety of selectable encryption algorithms to create secure channel and a variety of digital signature algorithms.

Boundary protection—It can be obtained with the use of IDS and firewalls.

Virtual Private Networks (VPNs)—Allow secure remote access to the cloud.

Figure 3 shows the class diagram of the secure IaaS architecture pattern. In this model, the package *Authenticator* is an instance of the Authenticator pattern (Fernandez13) and enables the Cloud Controller to authenticate Cloud Consumers/Administrators. Instances of the *Security Logger/Auditor* pattern are used to keep track of any access to cloud resources such as VMs, VMMs, and VMIs. The *Reference Monitor* enforces authorization rights. The *Filter* scans created virtual machines in order to remove malicious code. At this moment we have secured the VMI administration part of the cloud, the rest of the cloud is secured similarly. The VMM requires a new pattern, Secure VMM, not yet written. The separation of VMs is obtained with the Secure VM Operating System pattern (Fernandez 2013).

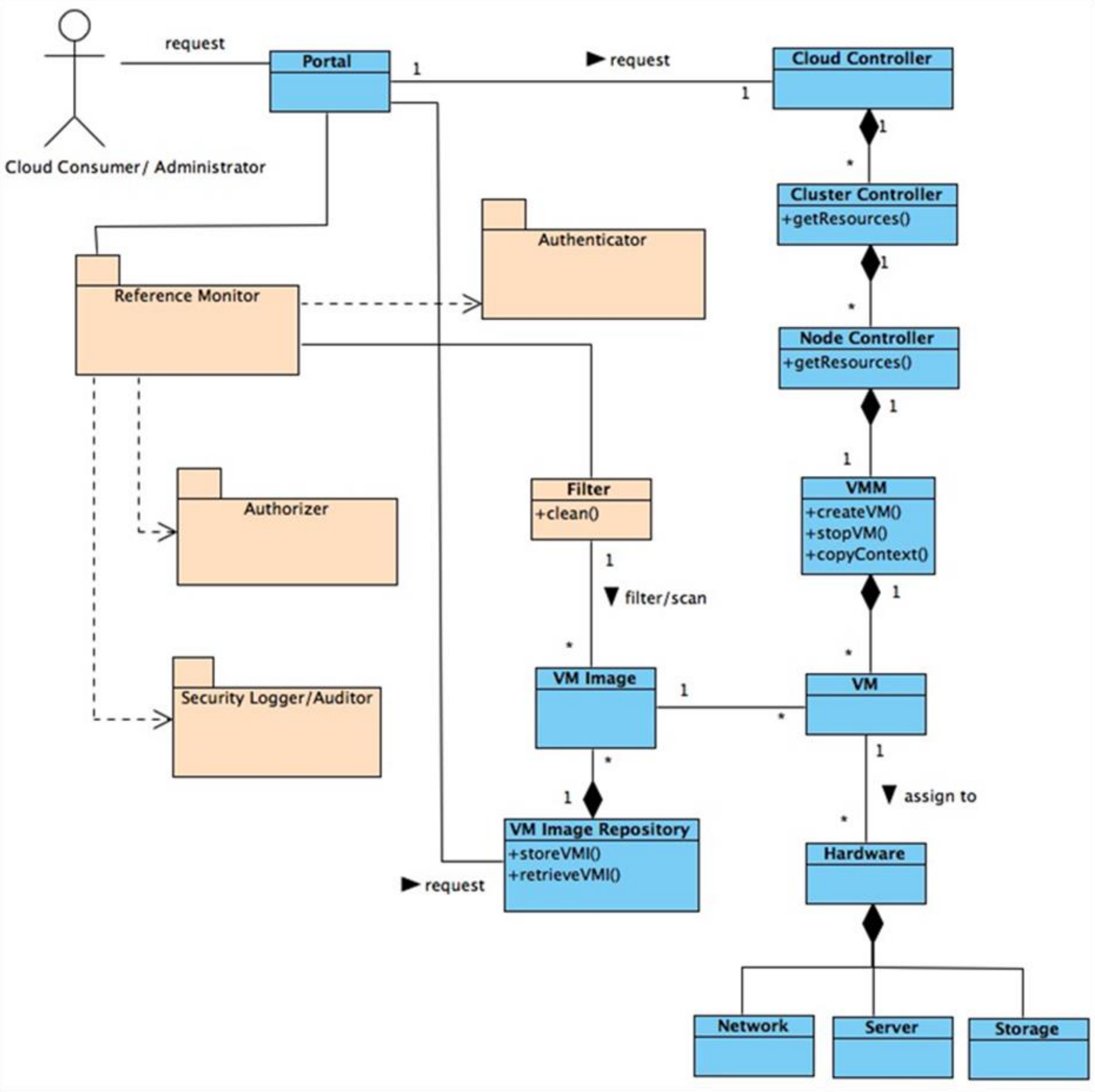

Figure 3: Class diagram of the Secure IaaS pattern

Implementation

Hypervisor security is fundamental aspect because it controls the separation of VMs. It can be obtained by simplicity, clear interfaces, and separation of functions. However, more and more functions are being added which increases its complexity.

Several papers, e.g. (CSA) have proposed *Security-as-a-Service* (SECaaS) to enhance or complement the existing security mechanisms of the service providers.

Known uses

- IBM's Reference Architecture includes virtualization management, monitoring and event management, and image lifecycle management. Its version 3.0 has also a section on security (IBM Corp. 2013). They use LDAP-based authentication, RBAC, encryption, and other mechanisms and follow the OAuth standard.

- Microsoft uses Role-Based Access Control (RBAC) applying a need-to-know policy to access resources, as well as multifactor authentication, and logging/auditing functions (Microsoft 2009]. Interesting features include protection of network DNSs using ACLs. They apply a defense-in-depth strategy and perform penetration testing. They also mention the use of code patterns and tool-based validation.

- (Okuhara et al. 2010, 2011) describe Fujitsu's security architecture which emphasizes the logical separation of computational environments, source code reviews, authentication and identity management (using WS-Federation). They also bring up the need for transparency from cloud providers, the use of a security dashboard for visualization of security functions, and the need to separate logging and monitoring functions. (Kino 2011) indicates that that the Fujitsu cloud separates virtual servers in a demilitarized zone (DMZ pattern in (Schumacher et al. 2006). This feature allows to place the global IP addresses of the virtual servers in a protected zone mapped to private addresses, which prevents their virtual IP addresses to be directly accessed. Fujitsu also provides authenticated VPNs to access cloud applications.

- Amazon describes in (Amazon Web Services 2013) the security aspects of their web services which include their cloud services. They comply with the Payment Card Industry (PCI) Data Security Standard (DSS) and control configuration management as well as standard authentication and authorization functions. The paper also describes the security of their Virtual Private Clouds, of their MapReduce database, and the handling of their firewall protection.

- VMware describe their SRA in terms of their hardware units with a few details of their functional aspects and its mapping to the PCI architecture (VMware and SAVVIS 2011).

- Oracle describes their SRA in (Oracle Cor. 2013). They have three versions of it, oriented to data security, fraud detection, and compliance.

- Cisco has a SRA called SAFE (Cisco 2009). They claim to apply principles like defense in depth, modular design, an best practices but they don't offer much detail of their SRA.

Consequences

The pattern has the following advantages:

Transparency - The users can use the provider's services without understanding their infrastructure. They fill their VMs with software through a menu without need to look at low-level details.

Flexibility - Degrees of security may be adjusted by tailoring encryption algorithms, fineness of authorization, and similar measures.

On-demand-service – Security services are an inherent part of the cloud services.

Contractual security—The degree of security can be specified in a service contract. However, monitoring is needed for its enforcement.

Manageability – It is possible to have a security dashboard to perform convenient administration of security services as is the case in several implementations, e.g. (Okuhara, Suzuki, Shiozaki, Hatori, 2011) and (OpenStack).

Monitoring—Clouds usually have monitoring services which can be specified in the service contracts.

Certification—Service providers can offer certified services.

Testability—A clear architecture based on controlling threats should allow an easy test of security.

Shared resources – Shared resources like servers or storage allow for the provision of expensive service controls.

Isolation – Hypervisors can provide good isolation. If this isolation is not enough clouds may offer options such as Virtual Private Clouds (Amazon Web Services 2013).

Shared Non-functional requirements provision (NFRs) – All clouds apply this business model to share the costs of security.

Some liabilities are:

Since security mechanisms are usually selected from industrial practice and not from a systematic analysis as in (Fernandez, Monge, Hashizume 2015), there may be missing or redundant defenses. Missing defenses affect the security of the system while redundant services may reduce performance.

Related patterns

- Secure Open IaaS—These are clouds where their source code is open; some of them are associated to companies that provide updates and improvements, others are strictly open source.
- Secure Open Stack—These are clouds that are open source and follow the Open Stack standard.
- Authenticator (Fernandez 2013)---when a user or system (subject) identifies itself to the system, how do we verify that the subject intending to access the system is who it says it is?
- Authorizer (Fernandez 2013)-- describes who is authorized to access specific resources in a system, in an environment in which we have resources whose access needs to be controlled. It indicates for each active entity, which resources it can access, and what it can do with them.
- Cloud IaaS (Hashizume, Fernandez, Larrondo-Petrie 2012 and E.B.Fernandez 2013)--describes the infrastructure to allow the sharing of distributed virtualized computational resources such as servers, storage, and networks.
- The Virtual Machine Operating System (Fernandez 2013)--describes the VMM and its created VMs from the point of view of an OS architecture.
- Security Logger/Auditor (Fernandez 2013)--How can we keep track of user’s actions in order to determine who did what and when? Log all security-sensitive actions performed by users and provide controlled access to records for Audit purposes.

3. Secure open IaaS

Intent

Describe the architecture required for the sharing of distributed virtualized computational resources such as servers, storage, and networks, including a set of security services. The implementation of these services is open source and different open source architectures may have different security services.

Context

The same context as in the Secure IaaS pattern with the addition that the source code of the IaaS implementation is openly available. Some open clouds may have maintenance performed by a company but not all.

Problem

Same as in IaaS with the specific problems brought upon by open source code, which include the fact that potential attackers can study the code to find vulnerabilities. This is particularly serious for those systems which are not maintained by any company.

Solution

IaaS must incorporate mechanisms that provide all the security attributes needed to support application execution: confidentiality, integrity, availability, and non-repudiation. It must also have a way to protect the information needed to apply these controls and the security mechanisms themselves. Open source systems may have a similar variety of security mechanisms their security level may be different because of the exposure of their code, although this level is a controversial subject (see Ristov and Gusev 2013).

Implementation

Open Source clouds try to comply with the ISO 27001:2005 standard. (Ristov and Gusev 2013) found that CloudStack conforms to all of the control objectives of this standard and in this sense it is better than Eucalyptus, OpenNebula, and OpenStack.

Known uses

Open source IaaS architectures include:

OpenStack is an open source IaaS architecture for the management and monitoring of infrastructure resources. Its architecture is described below.

Eucalyptus (Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems) is free and open-source software for building Amazon Web Services (AWS)-compatible private and hybrid cloud computing environments marketed by the company Eucalyptus Systems. Eucalyptus Systems announced a formal agreement with AWS in March 2012 to maintain compatibility. In September 2014, Eucalyptus was acquired by Hewlett-Packard. TCP/IP security groups share a common set of firewall rules. This is a mechanism to firewall off an instance using IP address and port block/allow functionality. At TCP/IP layer 2 instances are isolated.

Apache's *CloudStack* is open source software designed to deploy and manage large networks of virtual machines, as a highly available, highly scalable IaaS platform (CloudStack 2012). CloudStack is a turnkey solution that includes the entire "stack" of features: compute orchestration, Network-as-a-Service, user and account management, an open native API, resource accounting, and a User Interface. It is organized in a hierarchic structure: VMs run in Compute Nodes; several Compute Nodes run in a Pod (typically one rack); and several Pods with secondary storage form an Availability Zone (a data center). Configuration and administration are done by a Management Server in each Availability Zone.

Synnefo (Qevani et al., 2014), includes firewalls, Twitter-based authentication, and federated authentication using Shibboleth.

Consequences

This pattern has the following advantages:

Inspection by users—Because the code is open, potential users can inspect it to make sure it does not have vulnerabilities.

Certification—Open source service providers usually have an ISO 27001:2005 certificate. That standard is general and is being revised to be more specific to clouds (Ristov and Gusev 2013).

Possible liabilities are:

Inspection by attackers—Attackers can study the code to find vulnerabilities.

Contractual security—Unless they are supported by some company, open source clouds usually do not offer any guarantees about their level of security.

Related patterns

- Secure IaaS—Describes the security facilities of commercial cloud systems.
- Secure Open Stack—These are clouds that are open source and follow the Open Stack standard.

4. Secure Open Stack

Intent

Describe the architecture required for the sharing of distributed virtualized computational resources such as servers, storage, and networks, including a set of security services. The implementation of these services is open source and different open source architectures may have different security services. OpenStack defines a standard that contains a set of security services but specific implementations may have additional security services.

Context

Figure 4 shows the architecture of OpenStack which is organized as a set of components (OpenStack, Y. Zhang et al., 2014):

- Compute (Nova—provides virtual machine management.
- Object storage (Swift)—supports data management

- Block storage (Cinder)—provides persistent storage
- Image (Glance)—provides functionality for disk image management
- Web dashboard (Horizon)—provides a dashboard for user to use the services

Problem

Same as for Secure IaaS and Secure Open IaaS. A new force would be:

Interoperability—Clouds built to a similar standard can interoperate more conveniently and efficiently.

Easier design—The standard is a type of pattern that guides designers when building a new product.

Security by design—Design-based security is always a strong basis for security and can control the effect of code vulnerabilities.

Solution

OpenStack includes the following core security services (OpenStack):

- Identity (Keystone)—supports authentication and authorization, including different types of authentication. Keystone is an identity service providing authentication and high-level authorization for OpenStack. Congress (governance) can leverage Keystone as an input for policies. For example, an auditor might want to ensure that the running system is consistent with current Keystone authorization decisions.
- Barbican is a REST API designed for the secure storage, provisioning and management of secrets (Barbican).
- Networking (Neutron)—provides IP management, DNS, DHCP, security groups, firewall policies, and VPN management.
- ***Security Domains*** *:*

1. Public—the untrusted portion of the infrastructure. Any sensitive data in this domain must be protected by external controls.
2. Guest—refers to all data produced by applications running in the cloud VMs as well as internal communications between them.
3. Management—considers data and communications involving services in the OpenStack core.

4. Data—refers to information concerning the storage services of the cloud.

It is clear that the security services apply only to Public, Management, and Data domains. The use of a dashboard helps manage security.

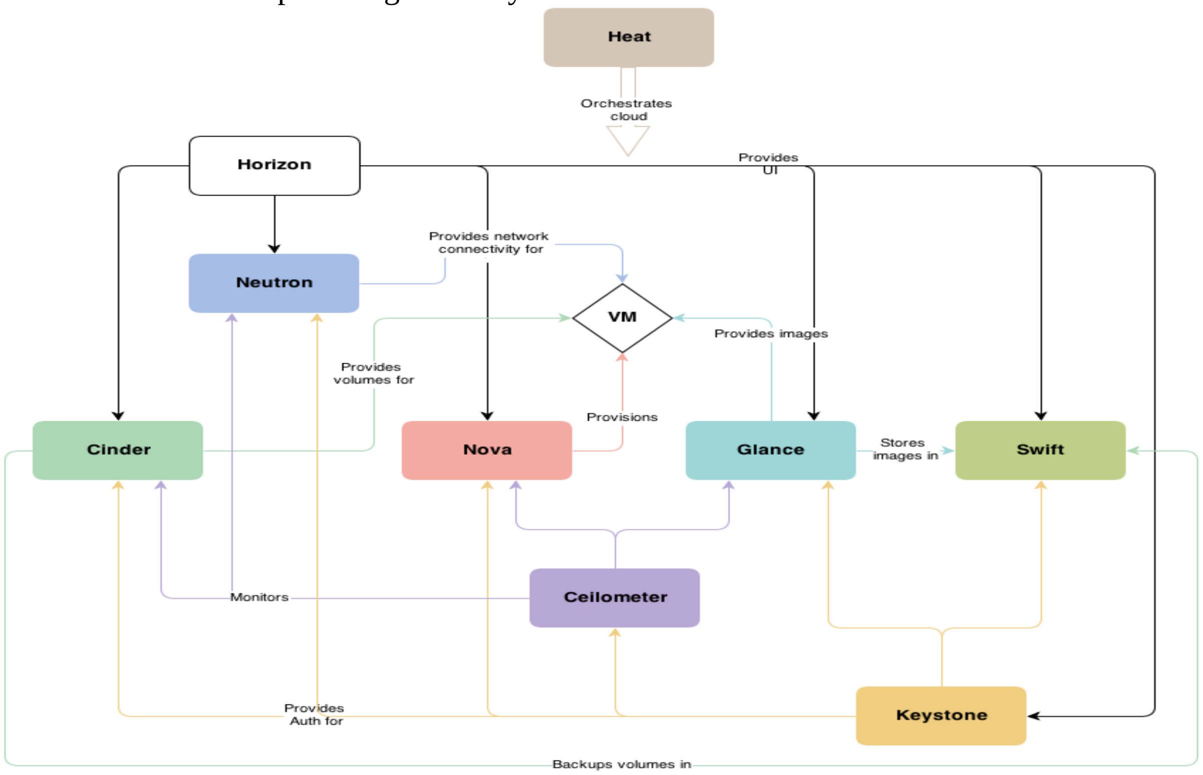

Figure 4. Conceptual architecture of OpenSack (from (OpenStack))

Known uses

The Nebula One private cloud system was built on the OpenStack open source cloud framework (Nebula). This company is now out of business.

HP's Helion platform uses OpenStack (HP).

Huawei has adopted OpenStack in its newer cloud systems.

Mirantis (Mirantis) is a company from California that focuses on the development and support of OpenStack.

Consequences

Use of domains may improve security for some applications but there is no analysis of this possibility. Being a De Facto standard has advantages and disadvantages.

The required isolation between virtual machines precludes cooperative work where resources need to be shared. Several models for information and resource sharing are described in (Zhang, Krishnan, and Sandhu, 2014).

Related patterns

Secure IaaS—Describes the security facilities of commercial cloud systems.

Secure Open IaaS—These are clouds where their source code is open; some of them are associated to companies that provide updates and improvements, others are strictly open source.

Several architectures for OpenStack have been presented, e.g. (Rosado and Bernardino, 2014), but none of them describes its security services.

A study of authentication in Keystone is presented in (B. Cui and T. Xi, 2015).

5. Conclusions

WE can make hierarchies of patterns, going from abstract patterns, independent of any implementation, to technologies, based on specific realizations such as MySQL databases, EJBs, and similar. We have presented here three hierarchic patterns that describe the abstract security aspects of any cloud IaaS, the security features of open source clouds, and the OpenStack cloud, a standard for open clouds. This analysis of commercial cloud systems complements our previous work on a conceptual security reference architecture for clouds and puts in perspective the possibilities we have when selecting service providers or designing new products.

Acknowledgements

We thank our shepherd, Teddy Chen, for his careful and insightful comments that improved the quality of this paper. The participants in the AsianPLoP workshop provided useful comments. The work of Prof. Fernandez was supported by travel grants from NII in March of 2015 and 2016.

References

[1] M. Anisetti, C. A. Ardagna, E. Damiani, F. Gaudenzi, R. Veca “Toward Security and Performance Certification of OpenStack,” IEEE CLOUD, 8th IEEE International Conference on Cloud Computing June 27 - July 2, 2015, New York, USA

[2] Amazon Web Services: Overview of security processes, June 2013. http://aws.amazon.com/security

[3] Baojiang Cui and Tao Xi, “ Security analysis of OpenStack Keystone”, 9th IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IEEE, 2015, DOI 10.1109/IMIS.2015.44

[4] Cisco, “Cisco SAFE: A Security Reference Architecture”, 2009

[5] CloudStack, “CloudStack open source cloud computing”, April 2012, http://cloudstack.org

[6] E.B.Fernandez, “Security patterns in practice: Building secure architectures using software patterns”, Wiley Series on Software Design Patterns, 2013.

[7] CSA, Cloud Security Alliance, http://cloudsecurityalliance.org

[8] E. B.Fernandez, N. Yoshioka, H. Washizaki, and J. Yoder, "Abstract security patterns for requirements specification and analysis of secure systems", Procs. of the WER 2014 conference, a track of the 17th Ibero-American Conf. on Soft. Eng.(CIbSE 2014), Pucon, Chile, April 2014

[9] E.B.Fernandez, Raul Monge, and Keiko Hashizume, “Building a security reference architecture for cloud systems”, Requirements Engineering, 2015 Doi: 10.1007/s00766-014-0218-7

[10] E.B. Fernandez, N. Yoshioka, and H. Washizaki, “Patterns for Security and Privacy in Cloud Ecosystems”, Procs. 2nd. Int. Workshop on Evolving Security and Privacy Requirements Engineering (ESPRE 2015), 13-18. Track of the 23rd IEEE Int. Requirements Eng. Conf., August 24-28, 2015, Ottawa, Canada.

[11] K. Hashizume, E. B. Fernandez, M. Larrondo-Petrie, Cloud Infrastructure Pattern, First International Symposium on Software Architecture and Patterns, in conjunction with the 10th Latin American and Caribbean Conference for Engineering and Technology, July 23-27, 2012, Panama City, Panama

[12] HP. The HPE Helion platform, http://www8.hp.com/us/en/cloud/hphelion-platform.html?jumpid=ps_dqbty6ph9g&gclid=CJGf1d_agcwCFYhTgQodTaoJvw&gclsrc=ds

[13] IBM Corp., “IBMSmart Cloud.” Available: http://www.ibm.com/cloud-computing/us/en/. [Accessed: 12-Sep-2012].
[IBM13] IBM Developer Works, IBM CCRA-3.0-Security-External.pdf, June 24, 2013

[14] T. Kino, “Infrastructure technology for cloud services”, Fujitsu Sci. Tech. J., vol. 47, No 4, October 2011, 434-442.

[15] S. Mahdavi-Hezavehi, U. van Heesch, P. Avgeriou, A Pattern Language for Architecture Patterns and Software Technologies Introducing Technology Pattern

Languages, in proceedings of the 16th European Pattern Languages of Programming (EuroPLoP) 2011, July 13-17, 2011, Irsee, Germany.

[16] Microsoft Global Foundation Services, Securing Microsoft Cloud infrastructure, May 2009, https://cloudsecurityalliance.org/securing-the-MS-Cloud.pdf

[17] Mirantis. https://www.mirantis.com/products/mirantis-openstack-software/

[18] Nebula One, https://en.wikipedia.org/wiki/Nebula_%28company%29

[19] M. Okuhara, T. Shiozaki, and T. Suzuki, "Security architectures for cloud computing", Fujitsu Sci. Tech. J., vol. 46, No 4, October 2010, 397-402.

[20] M. Okuhara, T. Suzuki, T. Shiozaki, and M. Hattori, "Fujitsu approach to cloud-related information security", Fujitsu Sci. Tech. J., vol. 47, No 4, October 2011, 459-464.

[21] OpenStack , http://www.openstack.org

[22] Oracle Corp., Security in Depth Reference Architecture, March 2013

[23] E. Qevanis, M. Panagopoulou, C. Stampoltas, A. Tsitsipas, D. Kyriazis, M. Themistocleous, "What can OpenStack adopt from a Ganetti-based open-source IaaS?, 2014 IEEE Int. Conf. on loud Computing, 833-840.

[24] S. Ristov and M. Gusev, "Security evaluation of Open Source clouds", Proc. of Eurocon 2013, Zagreb, Croatia. 73-79.

[25] T. Rosado and J. Bernardino, "An overview of OpenStack architecture", IDEAS'14, July 07-09, 2014, Porto, Portugal, 366-367.

[26] M. Schumacher, E. B.Fernandez, D. Hybertson, F. Buschmann, and P. Sommerlad, Security Patterns: Integrating security and systems engineering", Wiley Series on Software Design Patterns, 2006.

[27] A. Uzunov, E. B Fernandez, K. Falkner, "Security solution frames and security patterns for authorization in distributed, collaborative systems", Computers & Security, 55, 2015, pp. 193-234, doi: 10.1016/j.cose.2015.08.003

[28] VMWare and SAVVIS, "Securing the cloud: A review of cloud computing, security implications and best practices", White paper, 2011.

[29] Y. Zhang, R. Krishnan, and R. Sandhu, "Secure Information and Resource Sharing in Cloud Infrastructure as a Service", WISCS'14, November 2014, Scottsdale, AZ, 81-90.