Article can be accessed online at http://www.publishingindia.com

Towards Establishing Trust in Public Clouds through Real-time Client Feedback

Deepak Shukla*, Jogesh K Muppala, Subrota K Mondal***, Pranit Patil******

Abstract

Cloud computing, owing to its vast array of technological and commercial benefits, is being aggressively adopted by companies worldwide to meet their computing needs. Virtualization technology is the main enabler of cloud computing services making it economical and scalable for end-users. However, on the contrary, cloud services due to their inherent abstract nature pose significant security threats for user's data and applications; the most critical threat being the "malicious insider's threat" - the primary reason for lack of trust between a Cloud provider and its customers. In this paper, we analyze a cloud provider's basic internal operations required to provide IaaS services in order to understand and address the insider threat.Towards this goal, we inspect the virtualization stack and all the basic VM operations, the role of a cloud system administrator, their interactions with the virtualization ecosystem and therefore identify the scope of their possible malicious activities. We then review the present mechanisms that are adopted to implement trust in Clouds. Finally, we propose a Real-Time Client Feedback System (RTCFS)in the context of preventive and detective control in securing trust, aimed at increasing visibility and transparency for customers into public Clouds.We also suggest the use of job segregation for cloud administrators in order to restrict their individual capabilities to a minimal level. Both these mechanisms can help fill in the trust gap between a cloud provider and its customers.

Keywords: Virtualization, Malicious Insider, Preventive Control, Detective Control, RTCFS, Job Segregation, Transparency, Trust, Logging

1. Introduction

Virtualization is the enabling technology allowing the creation of a virtual machine on a physical server. There are usually multiple VM's running on such server's, essentially a multi-tenant environment at every server or node. This allows efficient resource sharing (compute, storage and network) on a large scale in a typical data-center environment which in turn is formalized as Cloud Computing.

In addition to providing scalable and flexible infrastructure to users, virtualization in cloud poses several security threats as well as concerns that prevent the users from trusting the cloud vendor completely. Malicious Insiders threat is one such critical threat which occurs every now and then. Also, cloud services are offered in three

* Assistant Professor, School of Technology, University of Technology & Management, Shillong, Meghalaya, India & Student, Dept. of Computer Science & Engineering, Hong Kong University of Science & Technology, Hong Kong, China. E-mail: deepak-shukla@outlook.com, deepak-shukla@live.com

** Associate Professor, Dept. of Computer Science & Engineering, Hong Kong University of Science & Technology, Hong Kong, China. E-mail: muppala@cse.ust.hk

*** PhD Scholar, Dept. of Computer Science & Engineering, Hong Kong University of Science & Technology, Hong Kong, China. E-mail: skmondal@cse.ust.hk

****Network Engineer, Cisco Systems, Bangalore, Karnataka, India. E-mail: pranit_patil@live.com

basic modes namely, SaaS – Software as a Service, PaaS – Platform as a Service and IaaS – Infrastructure as a Service. Recently, cloud vendors have come up with the term XaaS, meaning Anything-as-a-Service, where any computing service may be offered to the user over the cloud.

For our analysis, we only consider the IaaS model for analysis in this paper as it has the least abstraction amongst all the cloud offerings and allows a user to choose or employ security mechanisms as per their desired levels. There are significant security risks for sensitive data and/or applications hosted in clouds and therefore we try to investigate these issues and suggest suitable mechanisms to strengthen the overall security level in virtualized infrastructure of Clouds.

The rest of the paper is organized as follows: Section 2 discusses related work done in the context of cloud security esp. those dealing with threats related to the virtualized software stack and insider's attacks. Section 3 describes our views on the definition of trust in cloud computing. In section 4, we investigate the implementation of trust using present controls and mechanisms. We emphasize on supporting trust by implementing its sub-components and describe how the insider threat relates to increasing the trust level of cloud services. Section 5 extends our attempt to understand the malicious insider's issue by examining basic VM operations and the role of a cloud system administrator in the execution of these functions. We then discuss the threats and relevant suggestions identified during this process. Finally, we conclude by identifying those components of trustworthy clouds that have been successfully achieved and suggesting appropriate mechanisms for the ones that still need to be addressed. Finally, we try to identify those missing components that will help achieve maximum trust of cloud consumers.

2. Related Work

Correia etal.[6] described the use of TPM, both in hardware and software as a mechanism to establish trust amongst remote systems and as a critical tool for creating trustworthy cloud systems. However, while the TPM mechanism is capable of contributing to the creation of trustworthy clouds, the insider's threat still remains an unaddressed challenge.

Zhang etal.[7]did significant work in the area of cloud security by proposing a new Cloud architecture named "CloudVisor", which displaces the hypervisor and runs in the privileged mode, while the hypervisor or VMM along with management VM runs in the guest mode. This significant change in architecture of virtual resource management guarantees good protection for all the communication between a guest VM and VMM by separating resource management and security services.

Though Cloud Visor has the capability of making VM to VMM communication very secure, it does not increase the visibility into cloud operations, but only adds an extra layer of virtualization (nested virtualization), which in turn makes it even more difficult to log system and operational details for auditing purposes. Tracking back an event to clear details is very crucial for forensic investigations, external audits etc. Without these mechanisms, it would be impossible to tackle cybercrime cases that take place in the clouds. Therefore, we believe that we need simple and efficient methods that can monitor and record VM operations that will make forensic investigations easy and also feed clients with sufficient details with the help of these logs. This will eventually increase their visibility into the cloud operations and hence the overall trust.

Since our proposal of a client feedback system based on real-time log feeds, mainly aims at large enterprises which demand a higher level of trust than SME's or individual users, it can act as a tool to verify and ensure that the dynamic service workflows in the cloud infrastructure [20] meet the enterprises security and compliance needs as the client will be able to see what is happening to its data and/or applications.

3. Trust in Cloud Computing

It is essential to discuss the overall idea of trust in clouds before investigating any specific issues related to it.

3.1 Components of Trust in Cloud Computing [12]

The components of trust are individual information security goals that together are responsible and contribute to a user's overall trust on an information security system or mechanism.

To make Clouds trustworthy, it is apparent that all the four components of trust viz. Security, Privacy, Accountability and Auditability must be ensured completely. The goals of security and privacy have been addressed to a great extent by using techniques like data encryption (both in storage and network transit), multi-factor user authentication, key rotation, and role-based access for a group of users [14]. Accountability is supported by providing access logs on client's request, security processes, and risk and compliance agreements in the form of whitepapers[17]. In case of providing Auditability, individual administrative actions may not be logged and therefore can be very difficult to trace back to the source of the event in case anything goes wrong.

Accountability and Auditability are inter-related in the sense that without appropriate and complete accounting mechanisms or procedures, it is impossible to audit any unwanted incident and identify the culprit. Moreover, even if the actions are logged at the cloud provider's side, it may be accessible by the same administrative staff whose actions are being monitored and therefore could be modified for obvious reasons. Hence, we need a mechanism that will enable clients to have clear visibility into critical Cloud operations that are related to their own resources and therefore can put in sufficient confidence.

A summary of the mechanisms employed to implement individual components of trust are shown in TABLE I. Accountability and Auditability are mainly monitored on a periodic basis e.g. weekly or monthly and therefore, in their present form, provide only a consolidated summary of past events and present security controls being used. They do not enable a client to have sufficient visibility into the service provider's operations and the concomitant low levels of trust. Therefore, it is essential to introduce additional controls to completely address this issue of trust.

In order to understand the checkpoints at which the clients need visibility in VM operations and the scope of malicious activities of an insider, we investigate the basic VM operations and draw our observations that form the basis of our proposed system.

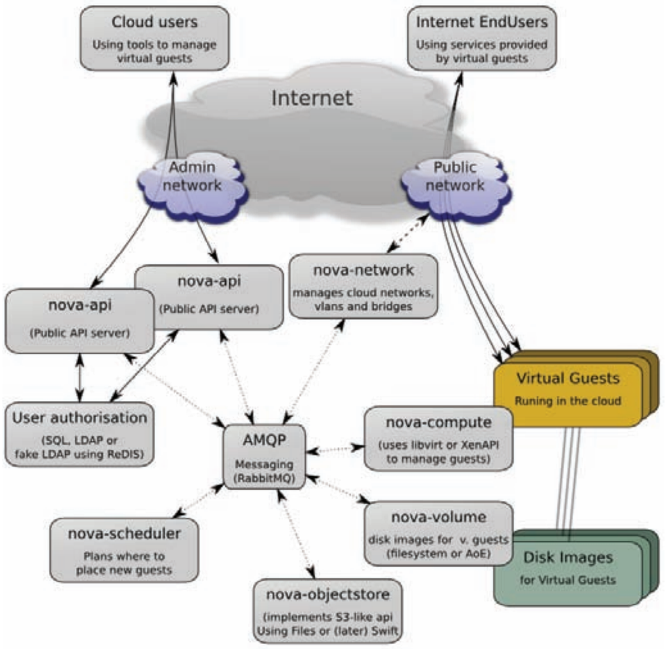
4. VM Operations and Observations

We consider OpenStack[18], which is a group of open source projects, that together aim at creating and managing public as well as private clouds. OpenStack helps to create IaaS in a convenient manner by providing generic API's that can work with different cloud vendors and thus prevent vendor lock-in issues. OpenStack is not a hypervisor but one of its components controls the hypervisor and other underlying functions. Due to the scope of the paper, we will not discuss details of OpenStack and will restrict our discussion to the critical issues in VM operations. The OpenStack architectural diagram giving an overall idea of cloud operations in general is shown in Figure 1.

Table 1: Mechanisms Used to Implement Trust Components

Trust Component	*Mechanisms Adopted*
Security	Physical Security, Firewalls, Intrusion Detection Systems, Remote Attestation using Public-Key Infrastructure etc.[17]
Privacy	Role-based access, Multi-factor authentication, VM Isolation, Key Rotation, data and VM encryption etc.[17]
Accountability	System Logs that mainly focus on server's status and overall network status reports. Service-level Agreements, practices as per the International regulations viz. Payment Card Industry (PCI) and Certification Authorities like VeriSign etc.
Auditability	Internal and External Audits

Figure 1: Open Stack Service Architecture


Cloud users
Using tools to manage virtual guests
Internet EndUsers
Using services provided by virtual guests
Internet
Admin network
Public network
nova-api
(Public API server)
nova-api
(Public API server)
nova-network
manages cloud networks, vlans and bridges
Virtual Guests
Runing in the cloud
User authorisation
(SQL, LDAP or fake LDAP using ReDIS)
AMQP
Messaging (RabbitMQ)
nova-compute
(uses libvirt or XenAPI to manage guests)
nova-volume
disk images for v. guests (filesystem or AoE)
Disk Images
for Virtual Guests
nova-scheduler
Plans where to place new guests
nova-objectstore
(implements S3-like api Using Files or (later) Swift)

In this diagram, *Cloud Users* refers to cloud system administrators, other cloud employees who reside inside the cloud provider's premises and are directly or indirectly related to user's assets in clouds. *Internet End Users* imply either individual cloud customers or an enterprise consuming cloud services. *Admin network* is a kind of internal network that is limited only to communication between the cloud components and carries only control information and not the instance information or any end user communication to running instances. Whereas the public network is one where end users can get connected to their instances using assigned public IP addresses (by nova-network manager component); it consists of traffic generated due to end user interaction with running instances.

Other components are typical cloud components like object storage, network controller, compute manager and so on. Specifically, the main management module in OpenStack is named as "nova" and thus the relative names of other components. Customers typically consume and interact with cloud services through the API's as shown above. In the following section, we briefly describe the basic VM operations and then describe our observations. For clarity we would like to emphasize that Virtual Cloud Controller or VCC is the combination of the main compute module, cloud management API's and the system administrator that operates on these two entities i.e. VCC = (nova-compute module + cloud management API + system administrator).

4.1. Virtual Machine Operations

(a) *Creation:* In this phase, a cloud user gets authenticated via 'user-authorization' module. It is a single-sign-on authorization where a cloud user can access all the assosicated services from various components of openstack. To create an instance, a cloud user sends request to VCC along with a predefined or customized VM template that contains basic OS parameters, RAM size etc.

Here, nova-scheduler picks up a computing node from the pool of available resources depending upon its scheduling algorithm and then the execution phase of the instance starts on that computer node (VMM).

(b) *Execution:*The VM starts executing an instance by using customized OS templates provided by the user or by fetching an available VM image stored in Image Store (Swift). After starting an instance, a keypair is generated and gets attached to it and nova-network component assigns a public IP addresses from the available pool of addresses. Finally, thisinstance is provided to the internet users along with the required information about the instance.

(c) *Migration:* It consists of moving a running instance between different physical servers(VMM) without disconnecting the client. It includes migration of memory, storage and network connectivity of a virtual machine.

(d) *Termination:* In this phase, a cloud user or internet user sends request to terminate the running instance. The termination process consists of stopping the instance, saving user data and removing any temporary data if used by virtual machine.

4.2 Classification of Threats

At different stages of a VM lifecycle, a cloud user or administrator may have varying capabilities to mount certain attacks. Therefore, it is essential to gauge these threats and take appropriate preventive and detective control measures.

To get a better understanding of these threats, we have classified them based on the VM lifecycle and the phase in which they are likely to occur. This is different from all the earlier approaches where threats have been classified based on perimeter security viz. Insider's threat and Outsider's threat. Following this specific classification basis helps us to understand each execution phase of VM more closely and also indicates which phase in the lifecycle of a VM is more susceptible to an insider attack. This is very important as it would allow us to know which critical parameters should be logged and monitored. A summary of these threats is presented below:

4.2.1 Creation

1. A cloud user can create and inject a customized VM template, which on execution can consume all the resources of a VMM and cause other running instances to starve and prevent them from running new instances on that VMM.
2. A malicious insider can add a malicious OS images in the ImageStore, which on execution can

provide unprivileged access to the user data to the insider.

4.2.2 Execution

1. Whena VMI has been booted in a VM, a key-pair is attached to this VM (SSH key-pair) that is used to get root access to the VM. Probably, this key-pair is visible to VCC. Thus a VCC may easily get the root access of a VM.
2. A system administrator can connect to the hypervisor of a server that hosts some sensitive data/applications. From the hypervisor, the admin can access the memory space of the VM. This enables him to access sensitive dataand can delete all VMs and can invalidate backups.
3. A VMI stored in the image store is visible to the VCC/Admin. Thus, any sensitive data stored in a VMI can be observed and captured by the admin.
4. Inter VM communication can expose sensitive data if it is not encrypted which is usually the default mode.

4.2.3 Migration

1. Admin can also take memory snapshot of a running instance and can migrate the snapshot to another VM right away and can execute that VM to recover details like OS parameters; software configuration files; user data if any.
2. During the migration VCC-VMM communication can be hijacked by mounting a "TCP session hijack" and therefore can change critical parameters of the process like manipulating the object code of sshd's authentication routines that can give root access of VM to any user.
3. Insiders can easily tap into internal network, capture and modify sensitive information in between migration if encryption is not used.

4.2.4 Termination

1. When cloud user/end user sends request to terminate the VM, instead of terminating, an insider can send false report about termination and can take complete control over it.
2. As some storage is attached to a running instance, it is possible to have some user information on it after termination of VM, so an insider can still read the information after termination of VM.

It should be clear by now that by default, the number of privileges bestowed upon a cloud system administrator outweigh their responsibilities and thus they have sufficient scope for malicious activities. Also, their actions that lead to such attacks are never logged or tracked on a real time basis, so as to react to it appropriately and in a timely manner. To address these major issues, we propose a "Real-Time Client Feedback System (RTCFS)" in the next section, which aims at monitoring and recording administrative actions using system-centric, data-centric and user-centric logging.

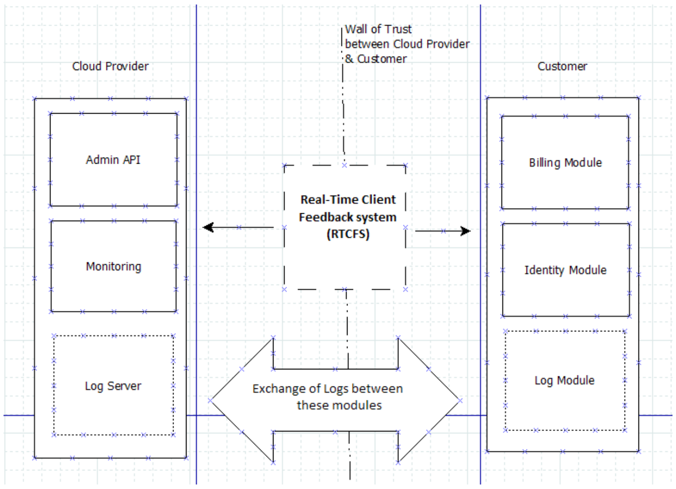
5. Real Time Client Feedback Systemand Job Segregation

To restrict the individual capabilities of the cloud system administrators, in a way that restricts them from mounting possible attacks on VM's but still allows them to perform their basic operations; we propose the use of "job segregation controls or role-based controls". OpenStack[19] provides such role-based cloud user configuration. For extremely sensitive applications and/or data, joint-authentication mechanisms can be added on top of job segregation [21].

While system logs have been used since the era of traditional IT systems, it is mostly system-centric and monitors the server's health and important applications. But the logging mechanism needs to adapted to suit the cloud ecosystem and implemented in a way that enables both cloud providers as well as enterprise customers to have a consistent view of the cloud operations. As we have argued throughout this paper, accommodating enterprise customers in the process of logging and sharing critical parameters is of immense importance in order to increase the visibility of cloud operations from the customer's perspective.

It is necessary to log critical administrative actions in a data-centric, system-centric and user-centric way to ensure traceability of the event, in case if anything goes wrong. Additionally, these logs should be ideally fed to the clients demanding it on a real time basis, which is the main goal of our proposed Real Time Client Feedback System. Feeding log files at real-time to the clients will help them monitor each security event which in turn can be used to analyze and anticipate security threats at real time and therefore generate system alerts in the event of a malicious activity. These log files can be directed to the client's log module as a network stream which can then be used to analyze typical operations. Also, such type of historical data at client's disposal will help them model suspicious activities and adapt accordingly to the evolving cloud threats in the future.

Figure 1.2: RTCFS High-Level Structure



Another important aspect of log generation is that this process should be dynamic and automatic in nature. It should not be possible for unauthorized cloud users including system administrators to read or modify these logs. If feasible, an independent group consisting of administrators from both sides (Cloud provider and Client) can supervise the logging workflow to achieve consistent results.

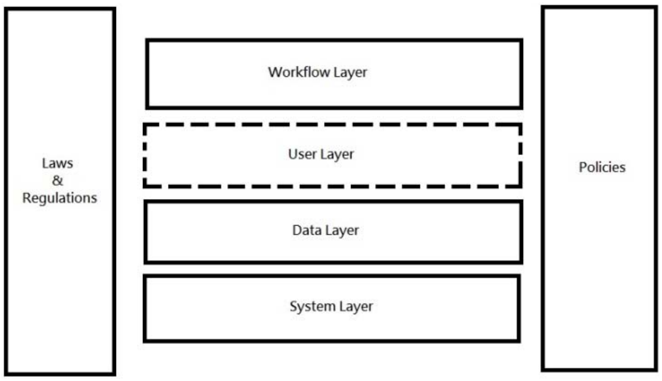
The parameters to be logged can be drawn upon by careful analysis of detailed VM operations and possible threats that can occur as shown in previous section. The previous section provides a very limited number of possible threats due to the scope of this paper, but conveys the basic idea behind classifying threats based on VM execution stages and drawing out critical log parameters from it.

A high-level diagram giving the basic idea of our approach is shown inFigure 2. This figure extends the basic conceptual architecture of OpenStack[18] by introducing two additional modules on both sides of the problem domain i.e. Customer side and Cloud provider's side. The "Log Server" on Cloud provider's side will be responsible for dispatching formatted logs to the concerned customers as per their demands. On the other hand, the "Log Module" on customer side shall be responsible for receiving these incoming logs on a real-time basis and also generating appropriate alerts, which would imply specific messages to the customer's staff.

Implementing these additional functionalities establishes a real-time channel that enables a customer to remain updated on the status of their data/applications as if the processing was being done in in-house data centers or systems. This in turn increases visibility from the client's perspective and transparency from Cloud provider's perspective.

RTCFS intends to exploit different types of logs to achieve its larger goal of providing transparency thereby increasing trustworthiness of cloud operations. TrustCloud framework [12] identifies three layers for providing accountability in clouds: System, Data and Workflow layer. We augment this framework by adding an extra layer, User Layer, to specifically account for user logs and their individual administrative behaviors. Following is the adapted version of this framework;

Figure 1.3: Adapted Version of TrustCloud Framework [12]



(a) *System Layer:* It consists of system related logs of OS, file systems and kernel. For e.g. in Linux/Unix, they can be found in; "/var/log/message" : General messages and system related logs, "/var/log/kern.log" stores kernel logs etc.

(b) *Data Layer:* This layer is responsible for logging application specific logs which is the most important part which needs be analyzed by cloud user or clients. Data Layer logs will come from files like "/ var/log/maillog" ," /var/log/mysqld.log" etc.

(c) *User Layer:* This layer consists of user related information, User can be system user or can be a cloud user. User's user id / group id are stored in /etc/password and all users password are encrypted and are stored in /etc/shadow . Apart from this, most of user related information is stored in users' home directory /home/user/;e.g. /home/user/.bashrc is responsible for behavior of interactive shell personalized to user. /home/user/.bash_profile can consist of any startup command to be issued after executing bash.

(d) *Workflow Layer:* This highest layer will be capable of tracking the overall flow of the application through the cloud infrastructure e.g. modules that the application has gone through, the number of clusters it has been migrated from etc. It will help ensure that the application and data flow through the cloud adheres to compliance and regulatory requirements. E.g.ForOpenStack compute service, the log can be found in *"/var/log/nova"*and*"/var/log/nova/nova-compute.log"* which logs computer component related information logs.

RTCFS is aimed at increasing an enterprise's visibility into cloud operations with respect to security of its data and applications. This very concept of providing visibility into operations brings the offsite cloud operations closer to in-house data center operations in the sense that with RTCFS, the client is able to see what is happening to its data and/ or applications which is true for traditional in-house data centers. Therefore, it is obvious that an enterprise will have more trust for such transparent cloud providers. In context of preventive and detective controls, RTCFS has the ability to act as a preventive as well as detective control depending on the way the logs are used by the clients.

5.1 RTCFS as a Preventive Contol

Real-time log feeds can be analyzed to anticipate typical threats in the past. The idea is to model these typical threats as a sequence of logical steps that one may carry out in order to mount a particular attack and then use these models to identify typical threat patters from the log data.

With appropriate models, a sequence of malicious actions can be related and a security alert can be triggered to act preventively in such a situation. [29][30]

5.2 RTCFS as a Detective Control

Log data has been traditionally used as a detective control. It is used to trace back specific security incidents to the root cause or during external and internal audits to verify if all the organizational processes and individual actions are in accordance to the compliance requirements of the organization. Different types of logs may contain details of various systems, users and applications but the primary goal of using logs has always been to track individual actions and events and trace back the events to original owner of that event. It acts as a tool to ensure accountability and Auditability.

5.3 Generic Requirements for RTCFS

Log management [29] is a fairly established idea and deals with large volumes of computer-generated log messages (audit records, audit trails etc.). It covers log collection, centralized aggregation, long-term retention, log analysis (in real-time and in bulk after storage) as well as log search after reporting.

The reasons driving log management are security, system and network operations and regulatory compliance. These are the exact goals for a system like RTCFS which eventually aim at making public clouds trustworthy. We believe that to in order to design and implement a large system like RTCFS, both the cloud provider and the corresponding enterprise will have to collaborate to agree on common system goals, minimum logging requirements, common log formats, analysis tools etc.

Frameworks like Flume [33] and Apache Hadoop HBase[34] can be used in the development of RTCFS. Hadoop is widely used for log processing as it has the ability to ingest, process and analyze terabytes of log data. Such available and successful frameworks make it possible to design and implement a real-time system like RTCFS.

6. Conclusion

To conclude, the present cloud infrastructures need mechanisms like RTCFS to implement appropriate logging practices to achieve their goals of trustworthiness and cloud adoption. Future work in this direction includes implementing RTCFS using relevant established tools and frameworks as discussed earlier.

References

1. Jansen, W. A. (2001). *Cloud Hooks: Security and Privacy Issues in Cloud Computing*. In Hawaii International Conference on System Sciences, (pp. 1-10). 2011 44th Hawaii International Conference on System Sciences.
2. Morsy, M. A., Grundy, J. & Müller, I. (2010). *An Analysis of the Cloud Computing Security Proble*. In PROC APSEC 2010 Cloud Workshop. 2010.
3. Dawoud, W. (2010). *Infrastructure as a Service Security: Challenges and Solutions*. Takouna, I. Meinel, C. Hasso Plattner Inst. IEEE Inc. (pp. 1-8).
4. Lombardi, F. & Di Pietro, R. (2010). Secure virtualization for cloud computing. *Journal of Network and Computer Application*. doi:10.1016/j.jnca.2010.06.008
5. Griffin, J. L., Jaeger, T., Perez, R., Sailer, R., Van Doorn, L. & Caceres, R. (2005). *Trusted Virtual Domains: Toward Secure Distributed Services*. In Proc. of the First Workshop on Hot Topics in System Dependability (Hotdep05), Yokohama, Japan, June 2005. IEEE Press.
6. Rocha, F., Abreu, S. & Correia, M. (2011). The final frontier: Confidentiality and privacy in the cloud, security and privacy in an online world. *Computer*, September, 44(9), 44-50 .
7. Zhang, H. C. F., Chen, J. & Zang, B. (2011). *Cloud Visor: Retrofitting Protection of Virtual Machines in Multi-tenant Cloud with Nested Virtualization*. In To Appear the 23rd ACM Symposium on Operating Systems Principles, SOSP 2011.
8. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2009). *Above the Clouds: A Berkeley View of Cloud Computing*. UC Berkeley Reliable Adaptive Distributed Systems Laboratory, February 10, 2009.
9. Zissis, D. & Lekkas, D. (2012). Addressing cloud computing security issues. Future Generation Computer Systems, March, 28(3), 583-592.
10. Pearson, S. (2011). Toward accountability in the cloud, view from the cloud, IEEE internet computing. *IEEE Computer Society*, July/August, 15(4), 64-69.

11.Building Confidence in the Cloud: A Proposal for Industry and Government Action to Advance Cloud Computing", Microsoft, January 2010.

12. Ko, R. K. L., Jagadpramana, P., Mowbray, M., Pearson, S., Kirchberg, M., Liang, Q. & Lee, B. S. (2011). TrustCloud: A Framework for Accountability and Trust in Cloud Computing. HP Laboratories, HPL-2011-38.
13. http://en.wikipedia.org/wiki/Cloud_computing
14. Mell, P. & Grance, T. (2011). *The NIST Definition of Cloud Computing. NIST*. US Department of Commerce.

. Microsoft. (2010). Building Confidence in the Cloud: A Proposal for Industry and Government Action to Advance Cloud Computing.

15. http://en.wikipedia.org/wiki/Cloud_computing#Infrastructure_as_a_Service_.28IaaS.29
16. http://en.wikipedia.org/wiki/Cloud_computing#Deployment_models
17. Amazon, A. W. S. (2011). *Overview of Security Processesr*. Whitepaper.
18. http://openstack.org/
19. http://docs.openstack.org/trunk/openstack-compute/admin/content/users-and-projects.html
20. Mace, J. C., van Moorsel, A. & Watson, P. (2011). *The Case for Dynamic Security Solutions in Public Cloud Workflow Deployments*. Workshop on Dependability of Clouds, Data Centers and Virtual Computing Environments (DCDV 2011).
21. http://www.pinnacle-international.com/pdf/windows-host-access-tech-brief-us.pdf
22. N. Santos, K. P. Gummadi, and R. Rodrigues. (2009). *Towards Trusted Cloud Computing*. In HotCloud '09:The 1st USENIX Workshop on Hot Topics in Cloud Computing, (pp. 1-5).
23. Brodkin, J. (2008). *Gartner: Seven Cloud-Computing Security Risks*. Infoworld.
24. Vouk, M. (2008). *Cloud Computing-Issues, Research and Implementations*. In Proc. 30th International Conference on Information TechnologyInterfaces, (ITI 2008) IEEE, (pp. 31-40).
25. Pearson, S. (2009). Taking Account of Privacy When Designing Cloud Computing Services. In Proc. 2009 ICSE Workshop on SoftwareEngineering Challenges of Cloud Computing, IEEE ComputerSociety, (pp. 44-52).
26. Pearson, S. & Charlesworth, A. (2009). Accountability as a way forwardfor privacy protection in the cloud. *Cloud Computing*, 131-144.

27. Haeberlen, A. (2010). A case for the accountable cloud. *ACM SIGOPS Operating Systems Review*, 44(2), 52-57.
28. Catteddu, D. & Hogben, G. (2009). *Cloud Computing Risk Assessment.* European Network and Information Security Agency (ENISA).
29. http://en.wikipedia.org/wiki/Log_management_and_intelligence
30. http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf.
31. Wang, C., Ren, K. & Wang, Q. (2012). *Security Challenges for the Public Cloud*. IEEE Computer Society.
32. Hoffman, P. & Woods, D. (2010). *Cloud Computing: The Limits of Public Clouds for Business Applications*. IEEE Internet Computing.
33. https://github.com/cloudera/flume/wiki
34. http://hbase.apache.org/
35. http://www.cloudera.com/blog/2011/02/log-event-processing-with-hbase/