Achieving Accountability and Secure Logging to Increase Trust in Cloud Environment

Prof.Manish M.Potey
Computer Dept(K.J.S.C.E)
K.J Somaiya College of Engineering,Mumbai.

Deepti D. Nikumbh
K.J.S.C.E
K.J Somaiya College of Engineering,Mumbai.

ABSTRACT

Clouds are becoming an interesting alternative to dedicated IT infrastructure.The advantages of cloud computing are appealing, but it also carries certain degree of risk for its customers as well as the cloud service providers.There is a lack of trust in cloud by potential customers which acts as a barrier in widespread adoption of cloud computing technology. In order to increase trust in cloud, we need to make clouds transparent and accountable.However, current systems does not provide full transparency and accountability.They are unable to track user activities and data transfers effectively within cloud environment. In this paper we propose a framework which achieves accountability by generating logs for every user activity, further we provide a secure logging mechanism for safeguarding of logs, thus protecting confidentiality of users and integrity of logs from dishonest cloud providers.

Keywords

Cloud computing, logging, accountability, trusted cloud, detective mechanism, files centric logging, cloud forensics.

1. INTRODUCTION

Cloud computing has attracted a lot of attention in recent years. A commonly-accepted definition for cloud computing is provided by US National Institute of Standards and Technologies: "Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.".Cloud computing has generated huge market opportunities and is set to transform IT. Not only it provides cost saving through scalability and pay-as-you-use service model, but also reduces business risk since the organizations no longer need to borrow money for setting up infrastructure. To help realize these benefits, the potential barrier i.e. lack of consumer trust ,which prevents the adoption of cloud architecture should be addressed. Cloud computing requires companies and individuals to transfer some or all control of computing resources to cloud service providers (CSPs).Such transfers poses concerns since customer relinquish control over his data and computation and retains some control over virtual machines which can be manage remotely by network connection [1].Since user's does not have direct control over their resources, trust becomes a critical issue for them.Thus,customers have increased expectations that their data should be handled in responsible way i.e. its integrity ,security and privacy should be maintained..

Cloud's could be exposed to malicious insider attack or malicious outsiders attack i.e. hacker hacking the system. For example, the servers of the red hat Linux distribution were recently attacked and the intruder managed to introduce a vulnerability and even sign some packages of Linux operating system distribution [2].Unauthorized access can also occur when no hackers are involved e.g. those resulting from software malfunction at the provider end. Such data breach occurred in Google Docs during march 2009.Another example where data integrity was compromised as a result of provider malfunction is a recent incident with Amazon S3 where user experienced silent data corruption [2].

In a recent 2010 survey by Fujitsu Research Institute on potential cloud customers, it was found that 88% of potential cloud consumers are worried about *who* has access to their data, and demanded more awareness of what goes on in the backend physical server[3]. The cloud computing research community, particularly the Cloud Security Alliance, has recognized this in its *Top Threats to Cloud Computing Report (Ver.1.0)* ,and listed seven top threats to cloud computing[3]:

1. Abuse and nefarious use of cloud computing
2. Insecure application programming interfaces
3. Malicious insiders
4. Shared technology vulnerabilities
5. Data loss or leakages
6. Account, service and traffic hijacking
7. Unknown risk profile.

Accountability can be a key to address such issues. Methods that promote accountability and auditability of CSPs, such as the tracking of file access histories, will empower service providers and users to reduce five of the above seven threats: 1,2,3,5 and 7.

Further, safeguarding of accountability data i.e. logs generated is very crucial in cloud environment since these logs are helpful for carrying out cloud forensics [5]. CSP can collude with other users and can tamper the generated logs. For example, he can deleted some log entries, reorder some entries or even add some fake log entries [5]. In this paper we present a framework that achieves accountability and secure logging. Firstly we present a novel file-centric logging mechanism, this mechanism records all the file-centric access happening on VM's and data transfer occurring between the VM's and outside world. Further, a secure logging scheme is provided, which enable the user to check whether the logs provided by CSP are not tampered.

2. TRUST IN CLOUD ENVIRONMENT

2.1 Component of Trust in Cloud Computing

Following components are together responsible for achieving trust in cloud environment [4].

2.1.1 Security
Security prevents any unauthorized access of information occurring in cloud. Security, in current cloud environment is achieved to a large extent by sophisticated cryptographic methods.

2.1.2 Privacy
Privacy allows only authorized users to access cloud. Privacy is also achieved to a large extent by different authentication techniques.

2.1.3 Accountability
Accountability record's and track's each individual action happening on service provider's end. In current systems accountability is limited only to generating system logs. More research related to cloud accountability is needed as it is important for enabling auditability and transparency in organization.

2.1.4 Auditability
Goal of auditability is to trace an action back to the owner. Internal and external audits are both crucial, since it increases trust and transparency in CSP.
Following presents a table which shows different mechanisms adopted by each trust components.

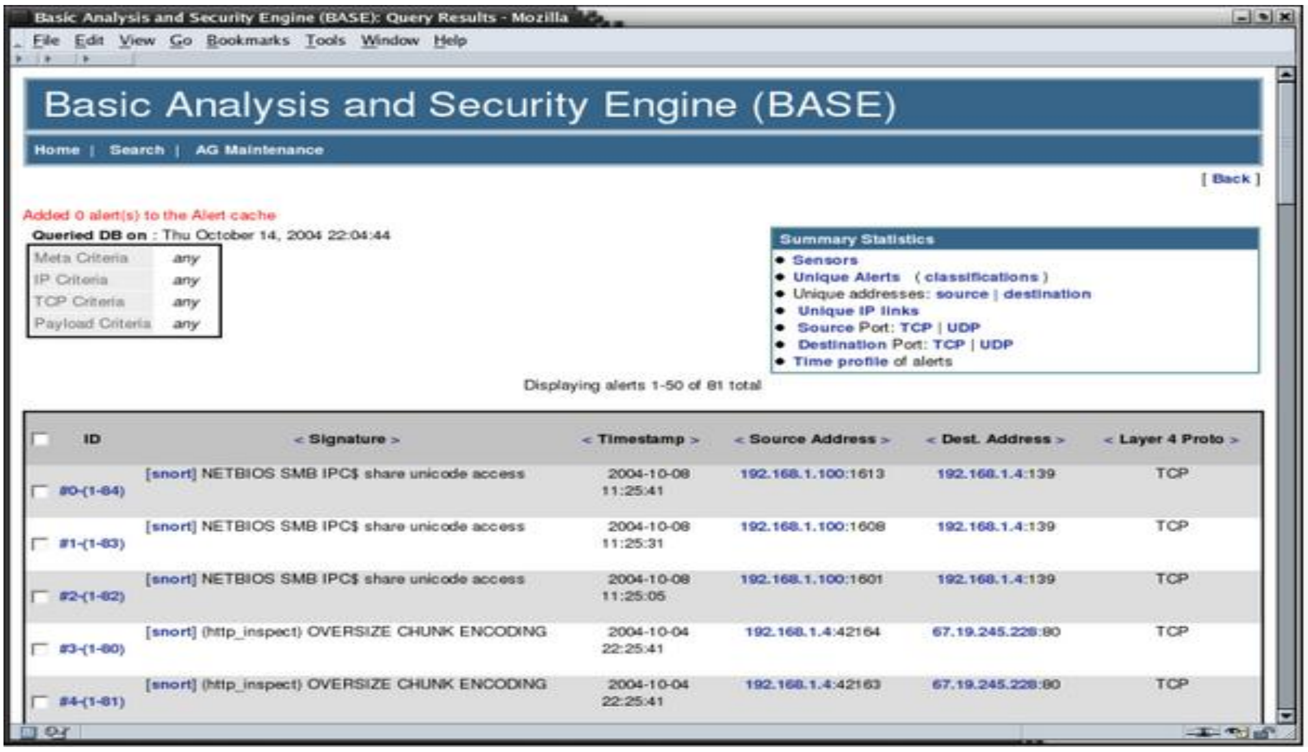
Table 1.Mechanisms Implemented By Trust Component[4]

Trust Component	**Mechanisms Adopted**
Security	Physical Security, Firewalls, Intrusion Detection System , Remote Attestation using public key infrastructure.
Privacy	Role-based access, Multi-factor authentication, VM isolation, Key Rotation, data and VM encryption.
Accountability	System logs which mainly focus on the server status and overall network status report.
Auditability	Internal and External Audits.

For achieving trustworthiness in cloud all the four components i.e. security, privacy,accountability and auditability must be ensured completely.

2.2 Existing Security Controls for Trust
In order to increase trust in cloud computing, there are both preventive and detective measures. The mechanisms through which security and privacy are achieved are classified under preventive measures whereas accountability and auditability are classified under detective measures. Preventive controls for privacy and security measures are actively being researched, but there is still little focus on detective controls related to cloud accountability and auditability .Many CSP focus on preventive measures(eg.better firewalls strong encryption,etc) and detective measures (eg.logging,audit trails, report for cloud forensics,etc) are neglected .The complexity resulting from sheer amount of virtualization and data distribution carried out in current clouds has also revealed an urgent need for research in cloud accountability[3].The customers are now more concerned about their data's integrity and confidentiality rather than health and utilization of servers.
Despite accountability being such a crucial component of improving trust, current prominent providers are still not providing full transparency and capabilities for the tracking and auditing of the file access history and data provenance [6] of both the physical and virtual servers utilized [3].Currently, users can at best monitor the virtual hardware performance metrics and the system event logs of the services they engage[3].Logs generated mainly focuses on the overall system health indicators i.e. uptimes, processor usage, events, etc. Especially in IAAS environment where users have control on virtual machine, it becomes difficult to implement accountability. Current systems at most can monitor the time stamping information about when the VM's where started and when the VM's where killed, which are not enough to track user activities.
Log information generated in cloud environment are highly sensitive and user's privacy issue are directly related to it. Previous studies do not provide a secure way of revealing the logs while maintaining user privacy.Moreover,it is vital to ensure that logs are not tampered before exposing it to investigators. Currently, to collect logs from cloud, investigators and user's are dependent on the CSP.Investigator needs to issue a subpoena to the CSP to acquire logs of a particular user [5].However they need to believe the CSPs blindly, as there is no way to verify whether the CSPs are providing valid logs or not.Moreover, if the adversary shuts down the virtual machine (VM) she/he is using.There is no way to collect the logs from terminated VM.

2.3 Approaches for Accountability and Secure Logging
Existing accountability techniques used in traditional client server architecture cannot be directly implemented in cloud environment. Cloud's are basically general purpose platforms and it should provide accountability for every services that user wants to run over it. Thus application specific technique like Repeat and Compare[10] is ruled out. The application independent techniques like Peer Review[11] requires that behavior of software should be deterministic which cannot be guaranteed in cloud environment.
Other tools like snort are used for monitoring packet in network. Fig 2 shown the logs generated by snort through BASE. BASE is a php script used to read the logs generated by Snort through a graphical interface. From the figure we can see that snort gives only the network information which alone is not enough to track user activities on cloud.
With the rise of virtualization technology ,tools like HyTrust Appliance[12] are becoming more prominent, but these tools does not provide full transparency of the user activities on cloud.Figure 2 gives the snapshot of the log generated by the Hytrust appliance.
Other tools like CloudKick[13] only focuses on server health and performance. Figure 3 shows a dashboard provided by cloudkick.Cloudkick dashboard provides an overview of infrastructure status; and graphs that help in visualizing bandwidth allowances and other metrics, as well as sends email alerts when things go wrong.



Added 0 alert(s) to the Alert cache

Queried DB on : Thu October 14, 2004 22:04:44

Meta Criteria	any
IP Criteria	any
TCP Criteria	any
Payload Criteria	any

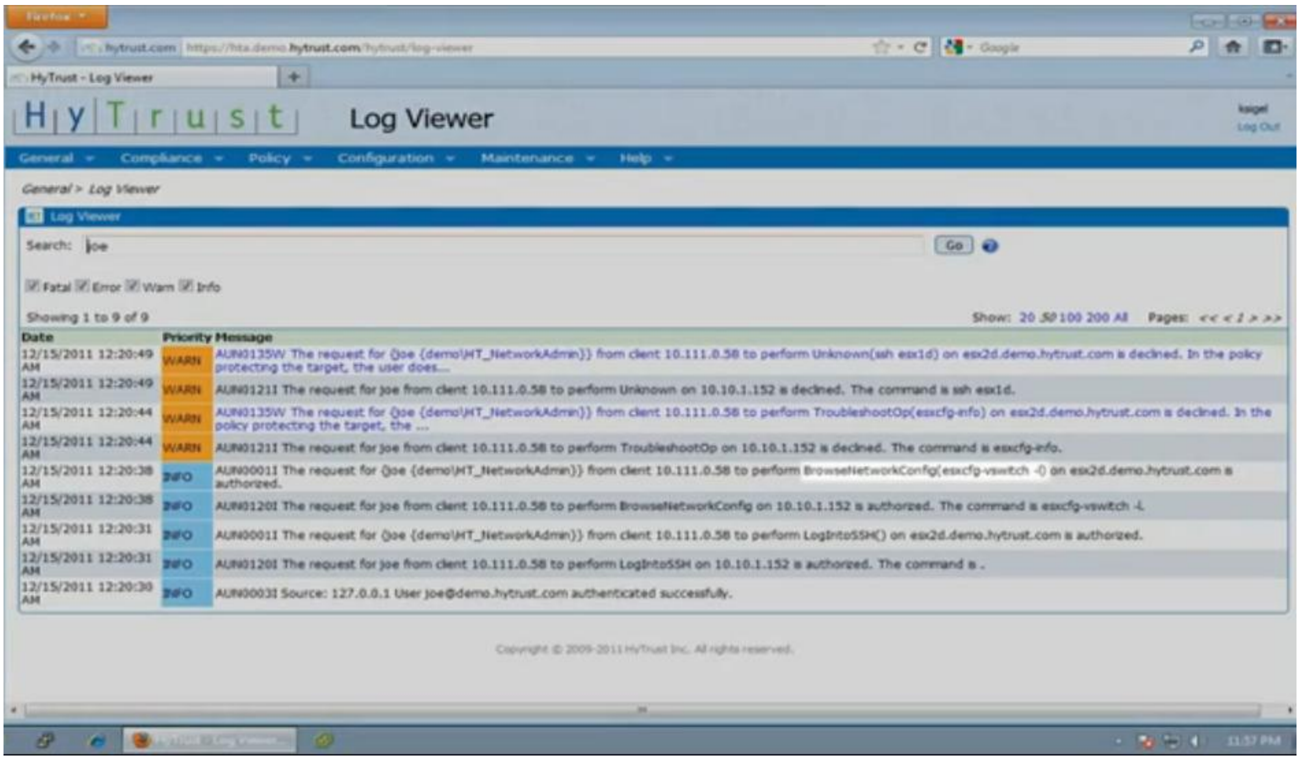
Summary Statistics
- Sensors
- Unique Alerts (classifications)
- Unique addresses: source | destination
- Unique IP links
- Source Port: TCP | UDP
- Destination Port: TCP | UDP
- Time profile of alerts

Displaying alerts 1-50 of 81 total

ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(1-84)	[snort] NETBIOS SMB IPC$ share unicode access	2004-10-08 11:25:41	192.168.1.100:1613	192.168.1.4:139	TCP
#1-(1-83)	[snort] NETBIOS SMB IPC$ share unicode access	2004-10-08 11:25:31	192.168.1.100:1608	192.168.1.4:139	TCP
#2-(1-82)	[snort] NETBIOS SMB IPC$ share unicode access	2004-10-08 11:25:05	192.168.1.100:1601	192.168.1.4:139	TCP
#3-(1-80)	[snort] (http_inspect) OVERSIZE CHUNK ENCODING	2004-10-04 22:25:41	192.168.1.4:42164	67.19.245.228:80	TCP
#4-(1-81)	[snort] (http_inspect) OVERSIZE CHUNK ENCODING	2004-10-04 22:25:41	192.168.1.4:42163	67.19.245.228:80	TCP

Basic Analysis and Security Engine (BASE): Query Results - Mozilla
File Edit View Go Bookmarks Tools Window Help
Basic Analysis and Security Engine (BASE)
Home | Search | AG Maintenance
[Back]

Figure 1:Snapshot Of Snort Logs Through BASE.



Search: joe

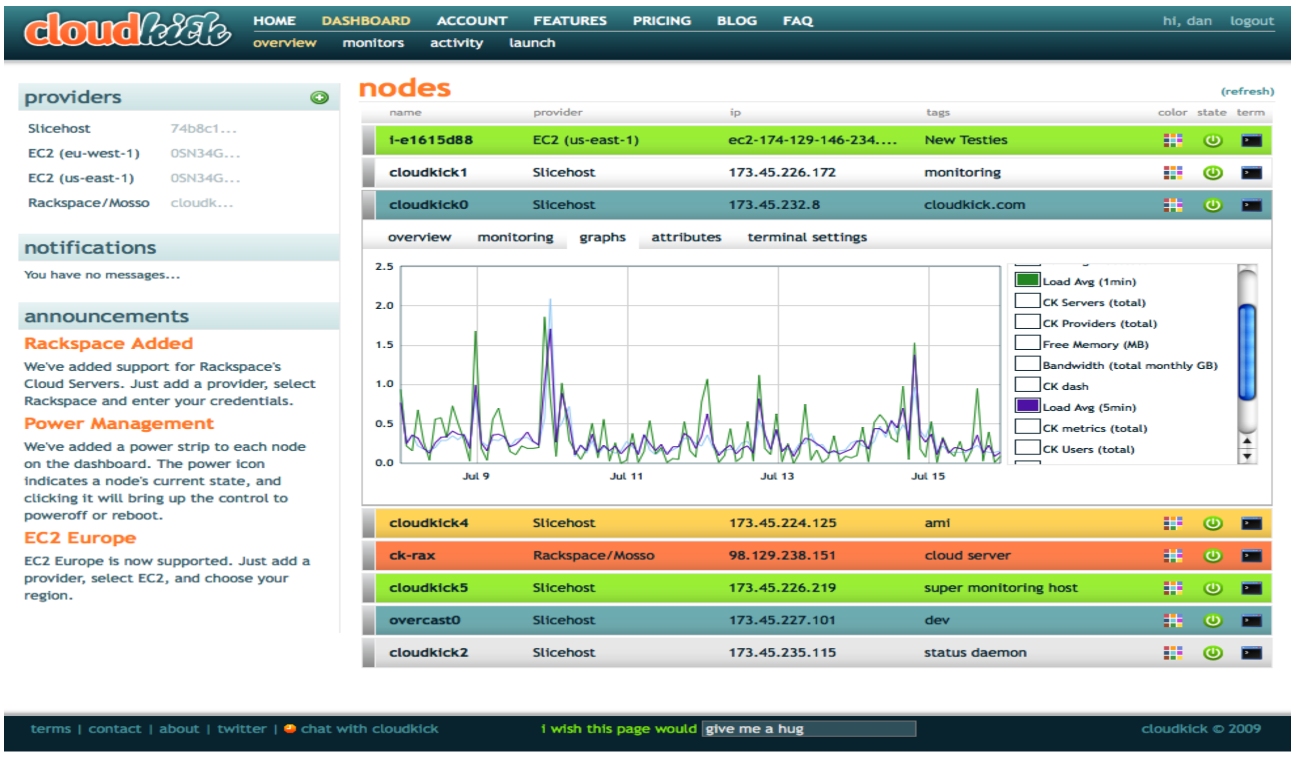
✓ Fatal ✓ Error ✓ Warn ✓ Info

Showing 1 to 9 of 9

Date	Priority	Message
12/15/2011 12:20:49 AM	WARN	AUN0135W The request for (joe (demo\HT_NetworkAdmin)) from client 10.111.0.58 to perform Unknown(ssh esx1d) on esx2d.demo.hytrust.com is declined. In the policy protecting the target, the user does...
12/15/2011 12:20:49 AM	WARN	AUN0121I The request for joe from client 10.111.0.58 to perform Unknown on 10.10.1.152 is declined. The command is ssh esx1d.
12/15/2011 12:20:44 AM	WARN	AUN0135W The request for (joe (demo\HT_NetworkAdmin)) from client 10.111.0.58 to perform TroubleshootOp(esxcfg-info) on esx2d.demo.hytrust.com is declined. In the policy protecting the target, the ...
12/15/2011 12:20:44 AM	WARN	AUN0121I The request for joe from client 10.111.0.58 to perform TroubleshootOp on 10.10.1.152 is declined. The command is esxcfg-info.
12/15/2011 12:20:38 AM	INFO	AUN0001I The request for (joe (demo\HT_NetworkAdmin)) from client 10.111.0.58 to perform BrowseNetworkConfig(esxcfg-vswitch -l) on esx2d.demo.hytrust.com is authorized.
12/15/2011 12:20:38 AM	INFO	AUN0120I The request for joe from client 10.111.0.58 to perform BrowseNetworkConfig on 10.10.1.152 is authorized. The command is esxcfg-vswitch -l.
12/15/2011 12:20:31 AM	INFO	AUN0001I The request for (joe (demo\HT_NetworkAdmin)) from client 10.111.0.58 to perform LogIntoSSH() on esx2d.demo.hytrust.com is authorized.
12/15/2011 12:20:31 AM	INFO	AUN0120I The request for joe from client 10.111.0.58 to perform LogIntoSSH on 10.10.1.152 is authorized. The command is .
12/15/2011 12:20:30 AM	INFO	AUN0003I Source: 127.0.0.1 User joe@demo.hytrust.com authenticated successfully.

HyTrust - Log Viewer
HyTrust
Log Viewer
General Compliance Policy Configuration Maintenance Help
General > Log Viewer
Log Viewer
Go
Show: 20 50 100 200 All
Pages: << < 1 > >>
Copyright © 2009-2011 HyTrust Inc. All rights reserved.

Figure 2:Snapshots of Hytrust Logs[12]



providers

Slicehost	74b8c1...
EC2 (eu-west-1)	0SN34G...
EC2 (us-east-1)	0SN34G...
Rackspace/Mosso	cloudk...

notifications

You have no messages...

announcements

Rackspace Added

We've added support for Rackspace's Cloud Servers. Just add a provider, select Rackspace and enter your credentials.

Power Management

We've added a power strip to each node on the dashboard. The power icon indicates a node's current state, and clicking it will bring up the control to poweroff or reboot.

EC2 Europe

EC2 Europe is now supported. Just add a provider, select EC2, and choose your region.

nodes

name	provider	ip	tags
i-e1615d88	EC2 (us-east-1)	ec2-174-129-146-234....	New Testies
cloudkick1	Slicehost	173.45.226.172	monitoring
cloudkick0	Slicehost	173.45.232.8	cloudkick.com
cloudkick4	Slicehost	173.45.224.125	ami
ck-rax	Rackspace/Mosso	98.129.238.151	cloud server
cloudkick5	Slicehost	173.45.226.219	super monitoring host
overcast0	Slicehost	173.45.227.101	dev
cloudkick2	Slicehost	173.45.235.115	status daemon

terms | contact | about | twitter | chat with cloudkick   i wish this page would give me a hug   cloudkick © 2009

Figure 3:Snapshot of CloudKick Dashboard[13]

3. ISSUES RELATED TO CLOUD ACCOUNTABILITY

Cloud computing empowered with virtualization introduces lot of challenges for achieving accountability and security of log's.

3.1 Tracking of Virtual to Physical Mapping and Viceversa [3]

Virtualization allows CSPs to use their server resources more efficiently. However, the addition of virtualized layers also means that accountability might require the identification not only of the virtual server in which an event takes place, but also the physical server.

3.2 Tracking Multiple Operating System Environments [3]

In IAAS model, virtual machines can have many different operating system, this potentially introduces the need to manage the logging of machines in the cloud which use a large number of different operating systems.

3.3 Operating System versus File-centric Logging [3]

Current tools focus on operating systems and system health monitoring (e.g. cloudstatus.com, etc), but few emphasize the file-centric perspective. By the file-centric perspective, we mean that we need to trace data and files from the time they are created to the time they are destroyed. When we log from a file-centric perspective, we view data and information independent from the environmental constraint.

3.4 Scope, Scale and Size of Logging [3]

Since logging feature is provided for every single activity done by user, amount of log records generated will be huge.Thus,an efficient mechanism is needed to manage such exponentially increasing log size. Detailed logs may reveal information that is private or sensitive, thus adequate controls on *who* gets access to this information, and for *what* purposes are needed [5]. Thus the scope and scale of logging may need to be limited for reasons of security and privacy as well as for manageability.

3.5 Reduced Level of Control and Dependence on the CSP [5]

In cloud environment users have a very limited control over their data and resources.Thus,for accessing log's and other information they extensively depend on the CSPs. From the figure 4,it can be observed that cloud users have highest control in IaaS and least control in SaaS In SaaS, customers do not get any logs of their system, unless the CSP provides the logs. In PaaS, it is only possible to get the application log from the customers. To get the network log, database log, or operating system log we need to depend on the CSP.In IaaS, customers do not have the network or process logs.Thus, we need to depend on the cloud service providers for acquiring log's, which in turn brings the honesty issue of the CSP's employee. CSPs can always tamper the logs.

3.6 Multi-Tenancy [5]

In cloud computing, multiple virtual machines (VM) can share the same physical infrastructure, i.e., log for multiple customers may be co-located. The nature of this infrastructure is different from the traditional single owner computer system. Hence, while collecting logs for one user, other users' data can be mingled with the log evidence. An alleged user can claim that the log contains information of other users, not her. The investigator then needs to prove that the provided logs indeed belong to the malicious user. Moreover, we need to preserve the privacy of the other tenants. For both of these issues, collecting and providing logs to the investigator is challenging in cloud paradigm.

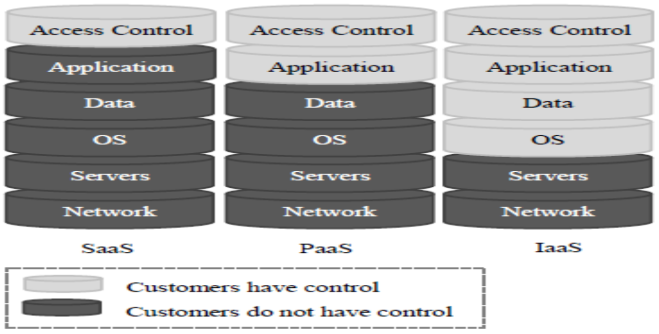


Figure 4:Customer's control over different layers in different service model[5]

4. PROPOSED FRAMEWORK

The entire framework is divided into two parts:
1)Logging mechanism.
2)Secure Logging scheme.

4.1 Architecture of Proposed Framework

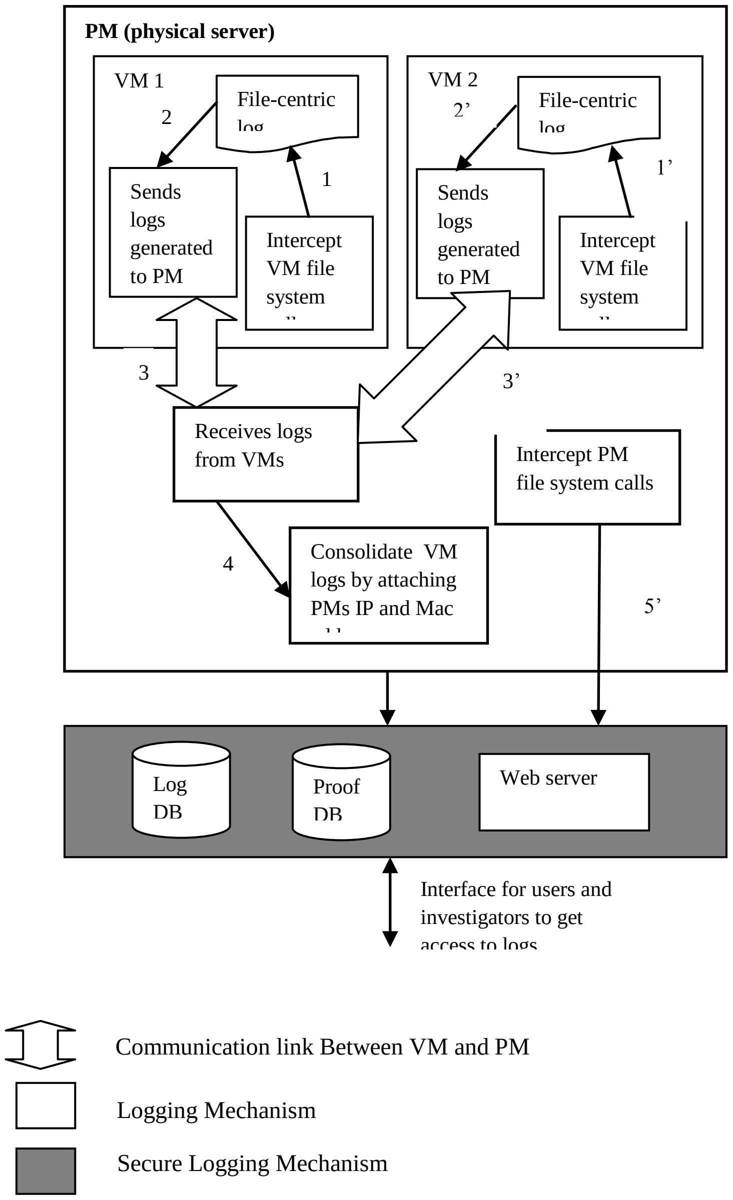

PM (physical server)
VM 1
File-centric log
2
1
Sends logs generated to PM
Intercept VM file system
VM 2
File-centric log
2'
1'
Sends logs generated to PM
Intercept VM file system
3
3'
Receives logs from VMs
Intercept PM file system calls
4
Consolidate VM logs by attaching PMs IP and Mac
5'
Log DB
Proof DB
Web server
Interface for users and investigators to get access to logs
Communication link Between VM and PM
Logging Mechanism
Secure Logging Mechanism

Figure 5:Proposed System

4.2 Logging Mechanism

For every user this mechanism records file-centric accesses, network access happening within the VM's, thus providing full transparency of entire user activities in the cloud. Basically two types of logs are maintained per user i.e. file access logs and network logs.

The following information is captured for every file access done on VM[7]:

- VM accessed file name and full path.
- VM file access date time.
- VM IP address.
- VM MAC address.
- Machine type VM/PM.
- UID of file owner of the accessed file.
- GID of file owner of the accessed file.
- UID of process owner who accessed the file.
- GID of process owner who accessed the file.
- Action done to accessed file e.g. create,read,write,socket(send message), socket(receive message),delete.

Apart from the file access logs, network logs i.e. information about VM to VM communication and communication of VM with outside world is also captured. The network logs will stored the following information:

- UID of the user .
- GID of user.
- From IP.
- To IP.
- Port number.
- Timestamp information i.e. date and time.

After the file access logs and network logs are captured it is sent to the underline ohysical machine (PM) where it is consolidated with PM's IP address and PM's MAC address. The consolidated logs are further sent for the encryption purpose.

Further, logs generated are crucial to the entire cloud environment, hence they should be safeguarded. Since database in which the logs are stored comes under the CSPs architecture, there are chances that administrator or some malicious cloud employee can delete some log entries or can reorder the log entries or even add some fake entries to it. Thus a mechanism for preventing such tampering of logs is needed. Next section presents a secure logging mechanism which guarantees that logs provided to the user or the investigator are not tampered

4.3 Secure Logging Mechanism [5]

4.3.1 *Overview*

This scheme ensures the integrity and confidentiality of the logs. After saving a log entry in the log database, the system will additionally store the proof of this entry in the proof database. When an investigator wants logs of a particular user to investigate an incident, he can get the necessary logs by an API call. In order to prove that logs as not tampered, proof of logs is provided along with logs.

4.3.2 *Schematic Description*

Consider network logs,log record LR for network log is defined as follows:

$$LR = \left\langle UID; GID; FromIP; ToIP; Port; T_L \right\rangle \qquad (1)$$

To ensure the confidentiality of users' log, some information of the LR can be encrypted using a common public key of the security agencies. The Encrypted Log Record ELR is prepared as follows:

$$ELR = \left\langle E_{PUa}\left(UID; GID; ToIP; Port\right); FromIP; T_L \right\rangle \qquad (2)$$

Where PUa is common public key of agencies. To preserve the correct order of every log record, a hash-chain scheme is used

$$LC = \left\langle H\left(ELR; LC_{Prev}\right)\right\rangle \qquad (3)$$

where LC_{Prev} is the Log Chain LC of the previous entry of the persistent storage. Each entry for the persistent log database DBE is constituted of ELR and LC,

$$DBE = \left\langle ELR; LC \right\rangle \qquad (4)$$

The proof of this DBE will be inserted into an accumulator. We denote this as Accumulator Entry AC. At the end of each day, CSP retrieves the AC_D of that day and generates the Proof of Past Log PPL as follows:

$$PPL = \left\langle H\left(AC_D\right); S_{PRc}\left(AC_D\right); t \right\rangle \qquad (5)$$

where $H(AC_D)$ is the hash of AC_D, t represents the proof generation time, and S_{PRc} (AC_D) is the signature over AC_D using the private key of the CSP, PRc.

After computing the PPL, the CSP will publish the PPL and its public key.

4.3.3 *Verification*

Whenever an investigators requests for logs of particular user,along with the collected logs ,proof of the logs are also provided.

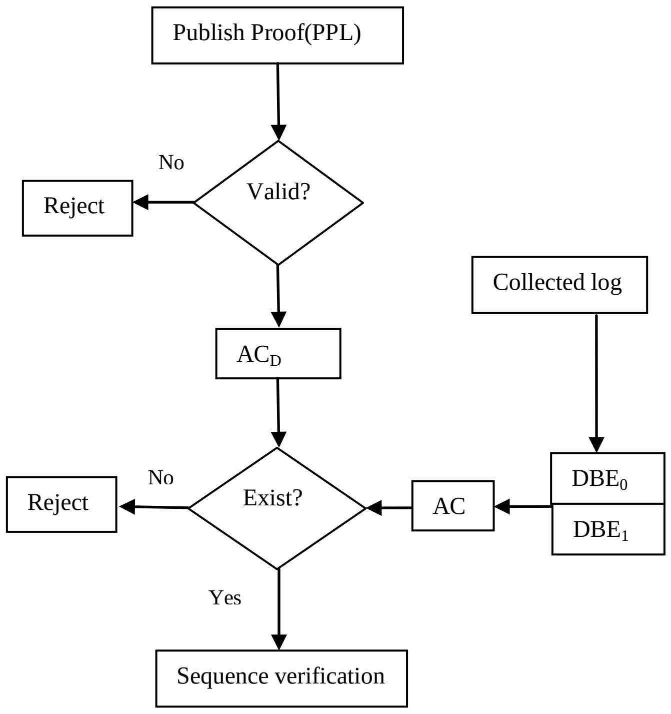


Figure 3:Log verification process Flow

The verification process starts from checking the validity of the published Proof of Past Log PPL. To do so, first, the investigator decrypts the $S_{PRc}(AC_D)$ using the public key of

the CSP and he will get the AC_D. Then he generates the hash value from the dycrypted AC_D. If the generated hash and the $H(AC_D)$ of the PPL matches, then the investigator accepts the PPL as a valid proof of log, otherwise he rejects the verification process.
In the next step, the investigator generates the Accumulator Entry AC for each DBE. Then, he will check whether the calculated AC exists in the AC_D. If exists, then the he proceeds towards log order verification process, otherwise he rejects the provided log information.

4.3.4 Sequence Verification
Figure 4 illustrates the log order verification process, where we verify whether the current log (DBE1) is actually after the previous log (DBE0) in the original sequence of log generation. In the figure 4, ELR0 denotes the Encrypted Log Record of the first log and ELR1 represents the same for the second log. To verify the correct order, the auditor calculates the Log Chain LCa from the first Log Chain LC0 and the second Encrypted Log ELR1 according to the following equation.

$$LCa = \left\langle H\left(ELR_1; LC_0\right)\right\rangle \quad (6)$$

If LCa matches with the 2nd Log Chain LC1 then the auditor accepts the logs, otherwise he rejects it.

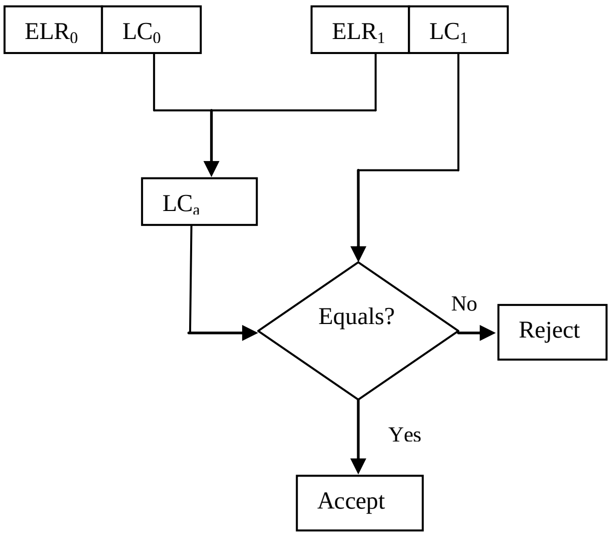


Figure 4:Log order verification process

4.3.5 Security Analysis
Logs are always generated and maintained by the CSP Thus,while presenting the logs to the user or the investigator the CSP can tamper the logs .Such attempts for violating the integrity and confidentiality of logs can be easily detected by the above mentioned scheme.

Following presents the possible attacks on the logs and how the above discussed scheme defends these attacks.

4.3.5.1 Removal of crucial log information
If any log entries are deleted ,it can be easily detected at verification stage.Let DBE_0, DBE_1, DBE_2,are the log entries which are present and there proof of log is already being published.Suppose CSP removes DBE_1 and provides only DBE_0 ,DBE_2 to the user.Such removal can be easily detected as log chain (LC) are linked,$H(ELR_2,LC_0)$ will not be equal to LC_2.

4.3.5.2 Reordering of logs
Suppose if CSP provides the log records in the order DBE_0, DBE_2, DBE_1,then by above technique it is easy to show that $DBLE_2$ is out of sequence.Further the CSP can change the original LC_2,so that order breaking is not detected.But this can be caught during individual log verification process ,as fake DBE_2 will not be present in PPL.

4.3.5.3 Inserting false log information:
Suppose,DBE_F is a fake log while verification let AC_F is the accumulator entry generated for this log record.If it is fake then AC_F will not be present in AC_D of PPL and hence it will be rejected as incorrect log.

5. CONCLUSIONS
A current system requires that the user should completely trust the CSPs, which is not acceptable to many organizations. Thus, there is a need to develop mechanisms which will allow the organizations to monitor and assess the trustworthiness of CSP.
In this paper we propose a framework which helps to increase trust in CSP. Users are made accountable for every activity done by them on cloud.Further the accountability information generated is vital for conducting auditing and cloud forensics. Thus a secure logging mechanism is proposed by which, it is possible to store and provide logs to the user along with the proof that the logs are not tampered by anyone. Accountability combined with secure logging will make clouds more reliable and trustworthy.

6. REFERENCES
[1] A. Haeverlen, and, "A Case for the Accountable Cloud",ACM SIGOPS Operating System Review,vol.44,pp.52-57,2010.

[2] C. Cachin, I. Keidar, A. Shaer, "Trusting the Cloud", ACM SIGACT News, vol.40, No.2, June 2009.

[3] K. L. Ryan Ko, P. Jagadpramana, M. Mowbray, S. perasons, M. Kirchberg, Q. Liang, B. S. Lee "TrustCloud-A freamework for accountability and trust in cloud computing",IEEE 2nd Cloud Forum for Practitioners,IEEE Computer Society, Washington DC, USA, 7-8 July 2011.

[4] J. K. Muppala,D. Shukla,S. K. Patil, "Establishing Trust in Public Clouds", J Inform Tech Softw Eng,2012.

[5] S. Zawoad, A. K. Dutta and R. Hasan, "SecLaas:Secure Logging-as-a-Service for Cloud Forensics", ACM Symposium on Information, Computer and Communications Security (ASIACCS) Feburary 25,2013.

[6] O. Q. Zhang, K. L. Ryan Ko, M. Kirchberg and B. S. Lee, "How to Track Your Data:The Case for Cloud Computing Provenance",January 21, 2012.

[7] K. L. Ryan Ko, P. Jagadpramana and B. S. Lee "Flogger:A File-centric Logger For Monitoring File Access and Transfer within Cloud Computing Environments",In Proc.IEEE 10th International Conference on Trust,Security and Privacy in Computing and Communication,pp.765-771 August 6,2011.

[8] S. Pearson and A. Charlesworth and, "Accountability as a Way Forward for Privacy Protection in the Cloud", Springer LNCS 5931, pp. 131–144, December 2009.

[9] S. Pearson, “Towards Accountability in the Cloud”,IEEE Internet Computing, IEEE Computer Society, vol.15, no.4, pp. 64-69, July/August 2011.

[10] N. Michalakis,R. Soule and R. Grimm,” Ensuring Content Integrity For Untrusted Peer to Peer Content Distribution Network”,In Proc. NSDI,April 2009.

[11] A. Haeberlen, P. Kuznetsov, P. Druschel,”PeerReview:Practical Accountability for Distributed Systems”,In Proc.SOSP,October 2007.

[12] HyTrust,”HyTrust Appliance”,2010;

http:/www.hytrust.com/product/overview/.

[13] CloudKick,”CloudKick-Cloud Monitoring and Management”,2011 http:/www.cloudkick.com.

[14] D. Ma, G. Tsudik,”A New Approach to secure Logging”,Trans.Storage,5(1):2:1,March 2009.

[15] R. Marty,”Cloud Application Logging For Forensics”,In Proc. ACM Symposium on Applied Computing,pp 178-184,2011.