International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 4, Issue 7 (November 2012), PP. 01-10

Identification and Implementation of Suitable Solutions to Critical Security Issues in Cloud Computing

Dr.G.Manoj Someswar[1,] Smt.Hemalatha[2]
[1]*Anwarul-uloom College of Engineering & Technology, PRINCIPAL, Yennepally, Vikarabad - 501101, RR Dist., A.P*
[2] *H.K.E.S Sree Veerendra Patil Degree college, HOD CSE Sadashivnagar, Bangalore, Karnataka, India.*

Abstract:- Cloud computing security (sometimes referred to simply as "cloud security") is an evolving sub-domain of computer security, network security and more broadly, information security. It refers to a broad set of policies, technologies, and controls deployed to protect data, applications, and the associated infrastructure of cloud computing [7].

Cloud security is not to be confused with security software offerings that are "cloud-based" (a.k.a. security-as-a-service).

There are a number of security issues/concerns associated with cloud computing but these issues fall into two broad categories: Security issues faced by cloud providers (organizations providing Software, Platform or Infrastructure - as - a – service via the cloud) and security issues faced by their customers. In most cases, the provider must ensure that their infrastructure is secure and that their clients' data and applications are protected while the customer must ensure that the provider has taken the proper security measures to protect their information.

The extensive use of virtualization in implementing cloud infrastructure brings unique security concerns for customers or tenants of a public cloud service. Virtualization alters the relationship between the OS and underlying hardware - be it computing, storage or even networking. This introduces an additional layer - virtualization - that itself must be properly configured, managed and secured. Specific concerns include the potential to compromise the virtualization software, or "hypervisor". While these concerns are largely theoretical, they do exist.

I. INTRODUCTION

Cloud computing is an upcoming paradigm that offers tremendous advantages in economical aspects, such as reduced time to market, flexible computing capabilities, and limitless computing power. To use the full potential of cloud computing, data is transferred, processed and stored by external cloud providers. However, data owners are very skeptical to place their data outside their own control sphere. This thesis discusses to which degree this skepticism is justified, by presenting the Cloud Computing Confidentiality Framework (CCCF). The CCCF is a step-by-step framework that creates mapping from data sensitivity onto the most suitable cloud computing architecture. To achieve this, the CCCF determines first of all the security mechanisms required for each data sensitivity level, secondly which of these security controls may not be supported in certain computing environments, and finally which solutions can be used to cope with the identified security limitations of cloud computing. The most thorough security controls needed to protect the most sensitive data may not be guaranteed in public cloud computing architectures, while they can be realized in private cloud computing architectures. As the most promising cloud computing approach, this thesis suggests selective cloud bursting, which acts as a hybrid cloud model with selective data transfers between public and private clouds.

When the underlying components fail in the cloud, the effect of the failures to the mission logic needs to be known so that correct recovery measures can be performed. We propose an application-specific run-time monitoring and management tool. With this tool, the application logic can remain on the consumer's host computer. This allows the consumer to centrally monitor all aspects of the application as well as data flow. Since all outputs from underlying services are sent to the application logic, any data incompatibility between services is not an issue. The capabilities of the run-time monitoring and management tool are as follows: 1) Enable application user to determine the status of the cloud resources that may be used to run the application (across multiple clouds), 2) Enable application user to determine the real-time security posture and situational awareness of the application, 3) Provide the application user with the ability to move user's application (or part of it) to another site (other VM in same cloud or different cloud altogether), 4) Provide the application user with the ability to change the application logic on the fly, 5) Provide communicate capabilities with cloud providers. There are a few cloud vendors such as NimSoft [41] and Hyperic [42] that provide application-specific monitoring tools that provide some of the above functionality. These monitoring tools may be further enhanced or used in conjunction with other tools to provide the degree of monitoring required. However, any tool that is to be used for military purposes must also receive some type of accreditation and certification procedure [2].

These SLAs typically state the high level policies of the provider (e.g. Will maintain uptime of 98%) and do not allow cloud consumers to dictate their requirements to the provider. COI clouds in particular have specific security policy requirements that must be met by the provider, due to the nature of COIs and the missions they are used for. These requirements need to be communicated to the provider and the provider needs to provide some way of stating that the requirements can be met. Cloud consumers and providers need a standard way of representing their security requirements and capabilities. Consumers also need a way to verify that the provided infrastructure and its purported security mechanisms meet the requirements stated in the consumer's policy (proof of assertions). For example, if the consumer's policy requires isolation of VMs, the provider can create an assertion statement that says it uses cache separation to support VM isolation. A **service-level agreement (SLA)** is a part of a service contract where the level of service is formally defined. In practice, the term *SLA* is sometimes used to refer to the contracted delivery time (of the service) or performance. As an example, internet

service providers will commonly include service level agreements within the terms of their contracts with customers to define the level(s) of service being sold in plain language terms. In this case the SLA will typically have a technical definition in terms of *mean time between failures* (MTBF), *mean time to repair* or *mean time to recovery* (MTTR); various data rates; throughput; jitter; or similar measurable details.

Chiles and McMakin (1996) define trust as increasing one's vulnerability to the risk of opportunistic behavior of another whose behavior is not under one's control in a situation in which the costs of violating the trust are greater than the benefits of upholding the trust.

Trust here means mostly lack of accountability and verifiability.

While VPC providers argue that they provide superior isolation, the fact of the matter is that your data is not a separate physical system: your data is still stored on actual servers along with other consumers' data, but logically separated. If the actual server fails, your data and applications stored on it are lost. Also, there needs to be a high level of trust as to the degree of isolation provided.

Government and Military sectors: complicated procurement rules and stringent security requirements
Cloud-based categories:

- Cloud-based applications (SAAS)
- Cloud-based development (e.g. Google App Engine)
- Cloud-based infrastructure (e.g. Amazon's EC2)

Trust and tenancy issues as well as loss of control related to the management model Data mobility: the abiltiy to share data between cloud services
Where does data reside?
- out-of-state, out-of-country issues

Security Concerns for government in particular

- FISMA
- How to certify and accredit cloud computing providers under FISMA (e.g. ISO 27001)

Differing data semantics example: does a data item labeled secret in one cloud have the same semantics as another piece of data also labeled secret in a different cloud?

In cloud computing (as well as other systems), there are many possible layers of access control. For example, access to the cloud, access to servers, access to services, access to databases (direct and queries via web services), access to VMs, and access to objects within a VM. Depending on the deployment model used, some of these will be controlled by the provider and others by the consumer.

For example, Google Apps, a representative SaaS Cloud controls authentication and access to its applications, but users themselves can control access to their documents through the provided interface to the access control mechanism. In IaaS type approaches, the user can create accounts on its virtual machines and create access control lists for these users for services located on the VM.

Regardless of the deployment model, the provider needs to manage the user authentication and access control procedures (to the cloud). While some providers allow federated authentication – enabling the consumer-side to manage its users, the access control management burden still lies with the provider. This requires the user to place a large amount of trust on the provider in terms of security, management, and maintenance of access control policies. This can be burdensome when numerous users from different organizations with different access control policies, are involved. This proposal focuses on access control to the cloud. However, the concepts here could be applied to access control at any level, if deemed necessary. We propose a way for the consumer to manage the access control decision-making process to retain some control, requiring less trust of the provider.

Cloud computing infrastructures enable companies to cut costs by outsourcing computations on-demand. However, clients of cloud computing services currently have no means of verifying the confidentiality and integrity of their data and computation.

To address this problem we propose the design of a trusted cloud computing platform (TCCP). TCCP enables Infrastructure as a Service (IaaS) providers such as Amazon EC2 to provide a closed box execution environment that guarantees confidential execution of guest virtual machines. Moreover, it allows users to attest to the IaaS provider and determine whether or not the service is secure before they launch their virtual machines.

Companies can greatly reduce IT costs by offloading data and computation to cloud computing services. Still, many companies are reluctant to do so, mostly due to

Outstanding security concerns. A recent study surveyed more than 500 chief executives and IT managers in 17 countries, and found that despite the potential benefits, executives "trust existing internal systems over cloud-based systems due to fear about security threats and loss of control of data and systems". One of the most serious concerns is the possibility of confidentiality violations. Either maliciously or accidentally, cloud

Provider's employees can tamper with or leak a company's data. Such actions can severely damage the reputation or finances of a company.

In order to prevent confidentiality violations, cloud services' customers might resort to encryption. While encryption is effective in securing data before it is stored at the provider, it cannot be applied in services where data is to be computed, since the unencrypted data must reside in the memory of the host running the computation. In Infrastructure as a Service (IaaS) cloud services such as Amazon's EC2, the provider hosts virtual machines (VMs) on behalf of its customers,

who can do arbitrary computations. In these systems, anyone with privileged access to the host can read or manipulate a customer's data. Consequently, customers cannot protect Their VMs on their own.

Cloud service providers are making a substantial effort to secure their systems, in order to minimize the threat of insider attacks, and reinforce the confidence of customers. For example, they protect and restrict access to the hardware facilities, adopt stringent accountability and auditing procedures, and minimize the number of staff who have access to critical components of the infrastructure [8]. Nevertheless, insiders that administer the software systems at the provider backend ultimately still possess the technical means to access customers' VMs. Thus, there is a clear need for a technical solution that guarantees the confidentiality and integrity of computation, in a way that is verifiable by the customers of the service.

Traditional trusted computing platforms like Terra take a compelling approach to this problem. For example, Terra is able to prevent the owner of a physical host from inspecting and interfering with a computation. Terra also provides a remote attestation capability that enables a remote party to determine upfront whether the host can securely run the computation. This mechanism reliably detects whether or not the host is running a platform implementation that the remote party trusts. These platforms can effectively secure a VM running in a single host. However, many providers run data centers comprising several hundreds of machines, and a customer's VM can be dynamically scheduled to run on any one of them. This complexity and the opaqueness of the provider backend creates vulnerabilities that traditional trusted platforms cannot address.

This RESEARCH THESIS proposes a trusted cloud computing platform (TCCP) for ensuring the confidentiality and integrity of computations that are outsourced to IaaS.

The TCCP provides the abstraction of a closed box execution environment for a customer's VM, guaranteeing that no cloud provider's privileged administrator can inspect or tamper with its content. Moreover, before requesting the service to launch a VM, the TCCP allows a customer to reliably and remotely determine whether the service backend is running a trusted TCCP implementation. This capability extends the notion of attestation to the entire service, and thus allows a customer to verify if its computation will run securely [1].

In this thesis, we show how to leverage the advances of trusted computing technologies to design the TCCP. We introduce these technologies and describes the architecture of an IaaS service. Section 3 presents our design of TCCP. Although we do not yet have a working prototype of TCCP, the design is sufficiently detailed that we are confident that a solution to the problem under discussion is possible.

II. OBJECTIVES OF STUDY

- Present cloud issues/characteristics that create interesting security problems.
- Identify a few security issues within this framework.
- Propose some approaches to address these issues with preliminary ideas to think about.
- Identify and Implement suitable solutions to critical security issues in cloud computing.

III. RESEARCH METHODOLOGY

Research Design

In this THESIS, we detail the most relevant TCCP mechanisms. We describe the protocols that manage the set of nodes of the platform that are trusted and the protocols that secure the operations involving VM(Virtual Machine) management, namely launching and migrating VMs. In these protocols, we use the following notation for cryptographic operations. The pair hKp, KP represents the private-public keys of an asymmetric cryptography keypair. Notation {y}Kx indicates that data y is encrypted with key Kx. We use a specific notation for the following keys: EKx denote endorsement keys, TKx indicate trusted keys, and Kx denote session keys. Nonces n, unique numbers generated by x. x, help detect message replays.

Node management

The TC dynamically manages the set of trusted nodes that can host a VM by maintaining a directory.

Message exchange during VM launch or each node within the security perimeter, the public endorsement key EKP identifying the node's N TPM, and the expected measurement list MLN . The ETE makes some properties of the TC securely available to the public, namely the EKP , the MLTC, and TC the TKP (identifying the TC). Both the MLN and the TC MLTC express the canonical configurations that a remote party is expected to observe when attesting to the platform running on a node N or on the TC, respectively.

In order to be trusted, a node must register with the TC by complying to the designated protocol. The following steps are involved in the process:

Steps 1 and 2, N attests to the TC to avoid an impersonation of the TC by an attacker: N sends a challenge nN to the TC, and the TC replies with its bootstrap measurements. MLTC encrypted with EKto guarantee the authenticity of the TC. If the MTC matches the expected configuration, it means the TC is trusted. Reversely, the TC also attests to N by piggybacking a challenge nTC in message 2, and checking whether the node is authentic, and is running the expected configuration (step 3). The pnode generates a keypair hTK,TKP i, and sends its NN public key to the TC. If both peers mutually attest successfully, the TC adds TKP to its node database, and N sends message 4 to confirm that the node is trusted. Key TKN certifies that node N is trusted.

In the case that a trusted node reboots, the TCCP must guarantee that the node's configuration remains trusted, otherwise the node could compromise the security of the p TCCP. To ensure this, the node only keeps TKin mem-N. [4]

IV. LIMITATIONS

- Are local host machines part of the cloud infrastructure?
  - Outside the security perimeter
  - While cloud consumers worry about the security on the cloud provider's site, they may easily forget to harden their own machines
- The lack of security of local devices can
  - Provide a way for malicious services on the cloud to attack local networks through these terminal devices
  - Compromise the cloud and its resources for other users
- With mobile devices, the threat may be even stronger
  - Users misplace or have the device stolen from them
  - Security mechanisms on handheld gadgets are often times insufficient compared to say, a desktop computer
  - Provides a potential attacker an easy avenue into a cloud system.
  - If a user relies mainly on a mobile device to access cloud data, the threat to availability is also increased as mobile devices malfunction or are lost.
- Devices that access the cloud should have
  - Strong authentication mechanisms
  - Tamper-resistant mechanisms
  - Strong isolation between applications

.

V. LITERATURE SURVEY

Infrastructure as a Service

Today, myriads of cloud providers offer services at various layers of the software stack. At lower layers, Infrastructure as a Service (IaaS) providers such as Amazon, Flexiscale, and GoGrid allow their customers to have access to entire virtual machines (VMs) hosted by the provider. A customer, and user of the system, is responsible for providing the entire software stack running inside a VM. At higher layers, Software as a Service (SaaS) systems such as Google Apps offer complete online applications than can be directly executed by their users.

The difficulty in guaranteeing the confidentiality of computations increases for services sitting on higher layers of the software stack, because services themselves provide and run the software that directly manipulates customer's data (e.g., Google Docs). In this paper we focus on the lower layer IaaS cloud providers where securing a customer's VM is more manageable.

While very little detail is known about the internal organization of commercial IaaS services, we describe (and base our proposal on) Eucalyptus , an open source IaaS platform that offers an interface similar to EC2.

This system manages one or more clusters whose nodes run a virtual machine monitor (typically Xen) to host customers' VMs. Eucalyptus comprehends a set of components to manage the clusters. For simplicity, our description aggregates all these components in a single cloud manager (CM) that handles a single cluster; we refer the reader to for more details.

From the perspective of users, Eucalyptus provides a web service interface to launch, manage, and terminate VMs. A VM is launched from a virtual machine image (VMI) loaded from the CM. Once a VM is launched, users can log in to it using normal tools such as ssh. Aside from the interface to every user, the CM exports services that can be used to perform administrative tasks such as adding and removing VMIs or users. Xen supports live migration, allowing a VM to shift its physical host while still running, in a way that is transparent to the user. Migration can be useful for resource consolidation or load balancing within the cluster.

Attack model

A sysadmin of the cloud provider that has privileged control over the backend can perpetrate many attacks in order to access the memory of a customer's VM. With root privileges at each machine, the sysadmin can install or execute all sorts of software to perform an attack. For example, if Xen is used at the backend, Xenaccess [7] allows a sysadmin to run a user level process in Dom0 that directly accesses the content of a VM's memory at run time. Furthermore, with physical access to the machine, a sysadmin can perform more sophisticated attacks like cold boot attacks and even tamper with the hardware.

In current IaaS providers, we can reasonably consider that no single person accumulates all these privileges. Moreover, providers already deploy stringent security devices, restricted access control policies, and surveillance mechanisms to protect the physical integrity of the hardware. Thus, we assume that, by enforcing a security perimeter, the provider itself can prevent attacks that require physical access to the machines.

Nevertheless, sysadmins need privileged permissions at the cluster's machines in order to manage the software they run. Since we do not precisely know the praxis of current IaaS providers, we assume in our attack model that sysadmins can login remotely to any machine with root privileges, at any point in time. The only way a sysadmin would be able to gain physical access to a node running a costumer's VM is by diverting this VM to a machine under her control, located outside the IaaS's security perimeter. Therefore, the TCCP must be able to 1) confine the VM execution inside the perimeter, and 2) guarantee that at any point a sysadmin with root privileges remotely logged to a machine hosting a VM cannot access its memory [6].

Trusted Computing

The Trusted Computing Group (TCG) [10] proposed a set of hardware and software technologies to enable the construction of trusted platforms. In particular, the TCG proposed a standard for the design of the trusted platform module (TPM) chip that is now bundled with commodity hardware. The TPM contains an endorsement private key (EK) that uniquely identifies the TPM (thus, the physical host), and some cryptographic functions that cannot be modified. The respective manufacturers sign the corresponding public key to guarantee the correctness of the chip and validity of the key.

Trusted platforms leverage the features of TPM chips to enable remote attestation. This mechanism works as follows. At boot time, the host computes a measurement list MLconsisting of a sequence of hashes of the software involved in the boot sequence, namely the BIOS, the bootloader, and the software implementing the platform. The MLis securely stored inside the host's TPM. To attest to the platform, a remote party challenges the platform running at the host with a nonce nU . The platform asks the local TPM to create a message containing both the ML and the nU , encrypted with the TPM's private EK. The host sends the message back to the remote party who can decrypt it using the EK's corresponding public key, thereby authenticating the host. By checking that the nonces match and the MLcorresponds to a configuration it deems trusted, a remote party can reliably identify the platform on an untrusted host.

A trusted platform like Terra [4] implements a thin VMM that enforces a closed box execution environment, meaning that a guest VM running on top cannot be inspected or modified by a user with full privileges over the host. The VMM guarantees its own integrity until the machine reboots. Thus, a remote party can attest to the platform running at the host to verify that a trusted VMM implementation is running, and thus make sure that her computation running in a guest VM is secure.

Given that a traditional trusted platform can secure the computation on a single host, a natural approach to secure an IaaS service would be to deploy the platform at each node of the service's backend. However, this approach is insufficient: a sysadmin can divert a customer's VM to a node not running the platform, either when the VM is launched (by manipulating the CM), or during the VM execution (using migration). Consequently, the attestation mechanism of the platform does not guarantee that the measurement list obtained by the remote party corresponds to the actual configuration of the host where the VM has been running.

The components of the trusted cloud computing platform include a set of trusted nodes (N) and the trusted coordinator (TC). The untrusted cloud manager (CM) makes a set of services available to users. The TC is maintained by an external trusted entity (ETE). Therefore, the TCCP needs to provide a remote attestation that guarantees the immutability of the platform's security properties in the backend.

Trusted Cloud Computing Platform

The trusted cloud computing platform (TCCP) that provides a closed box execution environment by extending the concept of trusted platform to an entire IaaS backend. The TCCP guarantees the confidentiality and the integrity of a user's VM, and allows a user to determine up front whether or not the IaaS enforces these properties.

Overview

TCCP enhances today's IaaS backends to enable closed box semantics without substantially changing the architecture. The trusted computing base of the TCCP includes two components: a trusted virtual machine monitor (TVMM), and a trusted coordinator (TC).

Each node of the backend runs a TVMM that hosts customers' VMs, and prevents privileged users from inspecting or modifying them. The TVMM protects its own integrity over time, and complies with the TCCP protocols. Nodes embed a certified TPM chip and must go through a secure boot process to install the TVMM.

Due to space limitations we will not go into detail about the design of the TVMM, and for an architecture that can be leveraged to build a TVMM that enforces local closed box protection against a malicious sysadmin.

The TC manages the set of nodes that can run a customer's VM securely. We call these nodes trusted nodes. To be trusted, a node must be located within the security perimeter, and run the TVMM. To meet these conditions, the TC maintains a record of the nodes located in the security perimeter, and attests to the node's platform to verify that the node is running a trusted TVMM.

Message exchange during node registration

The TC can cope with the occurrence of events such as adding or removing nodes from a cluster, or shutting down nodes temporarily for maintenance or upgrades. A user can verify whether the IaaS service secures its computation by attesting to the TC.

To secure the VMs, each TVMM running at each node cooperates with the TC in order to 1) confine the execution of a VM to a trusted node, and to 2) protect the VM state against inspection or modification when it is in transit on the network. The critical moments that require such protections are the operations to launch, and migrate VMs. In order to secure these operations, the TCCP specifies several protocols. Due to space constraints, we do not address other critical operations such as suspend/resume allowed by Xen.

It is assumed as an external trusted entity (ETE) that hosts the TC, and securely updates the information provided to the TC about the set of nodes deployed within the IaaS perimeter, and the set of trusted configurations. Most importantly, sysadmins that manage the IaaS have no privileges inside the ETE, and therefore cannot tamper with the TC. We envision that the ETE should be maintained by a third party with little or no incentive to collude with the IaaS provider e.g., by independent companies analogous to today's certificate authorities like VeriSign.

Consumer-managed access control

Approach:

This approach requires the client and provider to have a pre-existing trust relationship, as well as a pre-negotiated standard way of describing resources, users, and access decisions between the cloud provider and consumer. It also needs to be able to guarantee that the provider will uphold the consumer-side's access decisions. Furthermore, we need to show that this approach is at least as secure as the traditional access control model. This approach requires the data owner to be involved in all requests. Therefore, frequent access scenarios should not use this method if traffic is a concern. However, many secure data outsourcing schemes require the user to grant keys/certificates to the query side, so that every time the user queries a database, the owner needs to be involved. Therefore, not much different than that so may not be a problem.

Differing data semantics example: does a data item labeled secret in one cloud have the same semantics as another piece of data also labeled secret in a different cloud?

PDP

Short for *Packet Data Protocol*, it is a network protocol used by packet switching external networks to communicate with GPRS (General Packet Radio Services) networks. The PDP data structure is present on both the SGSN (*Service GPRS Support Node)* and the GGSN (*gateway GPRS support node)* that contains the subscriber's (MS) session information during an active session.

PEP

A Policy Enforcement Point (PEP) is the trusted component in the XACML architecture that enforces the decisions made by a Policy Decision Point. A PEP is located and run closely to the application that hosts the protected resource. A PEP can be as simple as an IF statement in the application, or as advanced as an agent running on an application server or a filter in an XML-gateway that intercepts access requests, gathers necessary data (attributes) for access decisions, collects access decisions and implements these decisions.

Axiomatics develops and provides a range of PEPs. Currently, there are a number of ready to use PEPs for application servers such as J2EE and .NET ASP. Axiomatics .NET ASP also has a support for Microsoft Active Directory Federation Services 2.0 (earlier called Geneva) claims. The Geneva modules gather claims and translate them into XACML attributes which are then, together with access requests, submitted to Axiomatics Policy Server for access decisions.
Axiomatics also provides PEPs for a number of XML-gateways. For more information and inquiries regarding our PEPs contact Axiomatics online or look for a sales representative on the Contacts page.

XACML

XACML is an OASIS standard that describes both a policy language and an access control decision request/response language (both written in XML). The policy language is used to describe general access control requirements, and has standard extension points for defining new functions, data types, combining logic, etc. The request/response language lets you form a query to ask whether or not a given action should be allowed, and interpret the result. The response always includes an answer about whether the request should be allowed using one of four values: Permit, Deny, Indeterminate (an error occurred or some required value was missing, so a decision cannot be made) or Not Applicable (the request can't be answered by this service).

The typical setup is that someone wants to take some action on a resource. They will make a request to whatever actually protects that resource (like a filesystem or a web server), which is called a Policy Enforcement Point (PEP). The PEP will form a request based on the requester's attributes, the resource in question, the action, and other information pertaining to the request. The PEP will then send this request to a Policy Decision Point (PDP), which will look at the request and some policy that applies to the request, and come up with an answer about whether access should be granted. That answer is returned to the PEP, which can then allow or deny access to the requester. Note that the PEP and PDP might both be contained within a single application, or might be distributed across several servers. In addition to providing request/response and policy languages, XACML also provides the other pieces of this relationship, namely finding a policy that applies to a given request and evaluating the request against that policy to come up with a yes or no answer.

Cloud Computing Background

- Features
  - Use of internet-based services to support business process
  - Rent IT-services on a utility-like basis
- Attributes
  - Rapid deployment
  - Low startup costs/ capital investments
  - Costs based on usage or subscription
  - Multi-tenant sharing of services/ resources
- Essential characteristics
  - On demand self-service
  - Ubiquitous network access
  - Location independent resource pooling
  - Rapid elasticity
  - Measured service

- "Cloud computing is a compilation of existing techniques and technologies, packaged within a new infrastructure paradigm that offers improved scalability, elasticity, business agility, faster startup time, reduced management costs, and just-in-time availability of resources".

Cloud Models

- Delivery Models
  - SaaS
  - PaaS
  - IaaS
- Deployment Models
  - Private cloud
  - Community cloud
  - Public cloud
  - Hybrid cloud
- Can propose one more Model: Management Models (trust and tenancy issues)
  - Self-managed
  - 3rd party managed (e.g. public clouds and VPC)

Cloud Computing: A Massive Concentration of Resources

- Also a massive concentration of risk
  - expected loss from a single breach can be significantly larger
  - concentration of "users" represents a concentration of threats
- "Ultimately, you can outsource responsibility but you can't outsource accountability."

Cloud Computing: who should use it?

- Cloud computing definitely makes sense if your own security is weak, missing features, or below average.
- Ultimately, if
  - the cloud provider's security people are "better" than yours (and leveraged at least as efficiently),
  - the web-services interfaces don't introduce too many new vulnerabilities, and
  - the cloud provider aims at least as high as you do, at security goals,

  then cloud computing has better security.

Problems Associated with Cloud Computing

- Most security problems stem from:
  - Loss of control
  - Lack of trust (mechanisms)
  - Multi-tenancy
- These problems exist mainly in 3rd party management models
  - Self-managed clouds still have security issues, but not related to above

Loss of Control in the Cloud

- Consumer's loss of control
  - Data, applications, resources are located with provider
  - User identity management is handled by the cloud
  - User access control rules, security policies and enforcement are managed by the cloud provider
  - Consumer relies on provider to ensure
- Data security and privacy
- Resource availability
- Monitoring and repairing of services/resources

Lack of Trust in the Cloud

- A brief deviation from the talk
  - (But still related)
  - Trusting a third party requires taking risks
- Defining trust and risk
  - Opposite sides of the same coin (J. Camp)
  - People only trust when it pays (Economist's view)
  - Need for trust arises only in risky situations
- Defunct third party management schemes
  - Hard to balance trust and risk
  - e.g. Key Escrow (Clipper chip)
  - Is the cloud headed toward the same path?

Multi-tenancy Issues in the Cloud

- Conflict between tenants' opposing goals
  - Tenants share a pool of resources and have opposing goals

- How does multi-tenancy deal with conflict of interest?
– Can tenants get along together and 'play nicely'?
– If they can't, can we isolate them?
- How to provide separation between tenants?

Security Issues in the Cloud

- In theory, minimizing any of the issues would help:
– Loss of Control
- Take back control
– Data and apps may still need to be on the cloud
– But can they be managed in some way by the consumer?
– Lack of trust
- Increase trust (mechanisms)
– Technology
– Policy, regulation
– Contracts (incentives): topic of a future talk
– Multi-tenancy
- Private cloud
– Takes away the reasons to use a cloud in the first place
- VPC: it's still not a separate system
- Strong separation

Minimize Lack of Trust: Policy Language [5]

- Consumers have specific security needs but don't have a say-so in how they are handled
– What the heck is the provider doing for me?
– Currently consumers cannot dictate their requirements to the provider (SLAs are one-sided)
- Standard language to convey one's policies and expectations
– Agreed upon and upheld by both parties
– Standard language for representing SLAs
– Can be used in a intra-cloud environment to realize overarching security posture
- Create policy language with the following characteristics:
– Machine-understandable (or at least processable),
– Easy to combine/merge and compare
– Examples of policy statements are, "requires isolation between VMs", "requires geographical isolation between VMs", "requires physical separation between other communities/tenants that are in the same industry," etc.
– Need a validation tool to check that the policy created in the standard language correctly reflects the policy creator's intentions (i.e. that the policy language is semantically equivalent to the user's intentions).

Minimize Lack of Trust: Certification[5]

- Certification
– Some form of reputable, independent, comparable assessment and description of security features and assurance
– Sarbanes-Oxley, DIACAP, DISTCAP, etc (are they sufficient for a cloud environment?)
- Risk assessment
– Performed by certified third parties
– Provides consumers with additional assurance

Minimize Loss of Control in the Cloud

- Monitoring
- Utilizing different clouds
- Access control management

Minimize Loss of Control: Monitoring[9]

- Cloud consumer needs situational awareness for critical applications
– When underlying components fail, what is the effect of the failure to the mission logic
– What recovery measures can be taken (by provider and consumer)
- Requires an application-specific run-time monitoring and management tool for the consumer
– The cloud consumer and cloud provider have different views of the system
– Enable both the provider and tenants to monitor the the components in the cloud that are under their control
– Provide mechanisms that enable the provider to act on attacks he can handle.
- infrastructure remapping (create new or move existing fault domains)
- shutting down offending components or targets (and assisting tenants with porting if necessary
- Repairs
– Provide mechanisms that enable the consumer to act on attacks that he can handle (application-level monitoring).
- RAdAC (Risk-adaptable Access Control)
- VM porting with remote attestation of target physical host

- Provide ability to move the user's application to another cloud

Minimize Loss of Control: Utilize Different Clouds [9]

- The concept of 'Don't put all your eggs in one basket'
– Consumer may use services from different clouds through an intra-cloud or multi-cloud architecture
– Propose a multi-cloud or intra-cloud architecture in which consumers
- Spread the risk
- Increase redundancy (per-task or per-application)
- Increase chance of mission completion for critical applications
– Possible issues to consider:
- Policy incompatibility (combined, what is the overarching policy?)
- Data dependency between clouds
- Differing data semantics across clouds
- Knowing when to utilize the redundancy feature (monitoring technology)
- Is it worth it to spread your sensitive data across multiple clouds?
– Redundancy could increase risk of exposure

Minimize Loss of Control: Access Control [9]

- Many possible layers of access control
– E.g. access to the cloud, access to servers, access to services, access to databases (direct and queries via web services), access to VMs, and access to objects within a VM
– Depending on the deployment model used, some of these will be controlled by the provider and others by the consumer
- Regardless of deployment model, provider needs to manage the user authentication and access control procedures (to the cloud)
– Federated Identity Management: access control management burden still lies with the provider
– Requires user to place a large amount of trust on the provider in terms of security, management, and maintenance of access control policies. This can be burdensome when numerous users from different organizations with different access control policies, are involved [10]
- Consumer-managed access control
– Consumer retains decision-making process to retain some control, requiring less trust of the provider (i.e. PDP is in consumer's domain)
– Requires the client and provider to have a pre-existing trust relationship, as well as a pre-negotiated standard way of describing resources, users, and access decisions between the cloud provider and consumer. It also needs to be able to guarantee that the provider will uphold the consumer-side's access decisions.
– Should be at least as secure as the traditional access control model.
– Facebook and Google Apps do this to some degree, but not enough control
– Applicability to privacy of patient health records

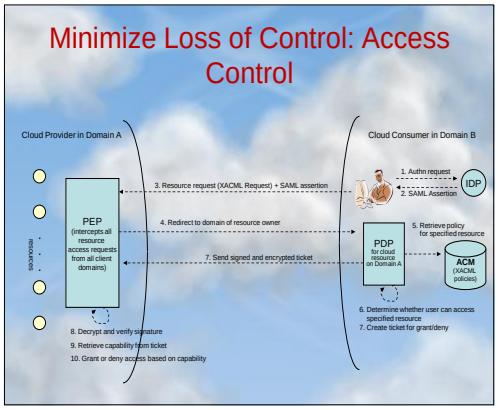

Fig1: Minimize Multi-tenancy in the Cloud

- Can't really force the provider to accept less tenants
– Can try to increase isolation between tenants
- Strong isolation techniques (VPC to some degree)
– C.f. VM Side channel attacks (T. Ristenpart et al.)
- QoS requirements need to be met
- Policy specification
– Can try to increase trust in the tenants
- Who's the insider, where's the security boundary? Who can I trust?
- Use SLAs to enforce trusted behavior

VI. CONCLUSION

- Cloud computing is sometimes viewed as a reincarnation of the classic mainframe client-server model
  – However, resources are ubiquitous, scalable, highly virtualized
  – Contains all the traditional threats, as well as new ones
- In developing solutions to cloud computing security issues it may be helpful to identify the problems and approaches in terms of
  – Loss of control
  – Lack of trust
  – Multi-tenancy problems

REFERENCES

[1.] M. Armbrust, *et al.*, "Above the Clouds: A Berkeley View of Cloud Computing," UC Berkeley Reliable Adaptive Distributed Systems Laboratory February 10 2009.
[2.] Cloud Security Alliance, "Security Guidance for Critical Areas of Focus in Cloud Computing, ver. 2.1," 2009.
[3.] M. Jensen, *et al.*, "On Technical Security Issues in Cloud Computing," presented at the 2009 IEEE International Conference on Cloud Computing, Bangalore, India 2009.
[4.] P. Mell and T. Grance, "Effectively and Securely Using the Cloud Computing Paradigm," ed: National Institute of Standards and Technology, Information Technology Laboratory, 2009.
[5.] N. Santos, *et al.*, "Towards Trusted Cloud Computing," in *Usenix 09 Hot Cloud Workshop*, San Diego, CA, 2009.
[6.] R. G. Lennon, *et al.*, "Best practices in cloud computing: designing for the cloud," presented at the Proceeding of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications, Orlando, Florida, USA, 2009.
[7.] P. Mell and T. Grance, "The NIST Definition of Cloud Computing (ver. 15)," National Institute of Standards and Technology, Information Technology LaboratoryOctober 7 2009.
[8.] C. Cachin, *et al.*, "Trusting the cloud," *SIGACT News*, vol. 40, pp. 81-86, 2009.
[9.] J. Heiser and M. Nicolett, "Assessing the Security Risks of Cloud Computing," Gartner 2008.
[10.] A. Joch. (2009, June 18) Cloud Computing: Is It Secure Enough? *Federal Computer Week*.

AUTHORS INFORMATION:

Dr.G.Manoj Someswar, B.Tech., M.S.(USA), M.C.A., Ph.D. is having 20+ years of relevant work experience in Academics, Teaching, Industry, Research and Software Development. At present, he is working as PRINCIPAL and Professor CSE Department Anwarul-uloom College of Engineering & Technology, Yennepally, Vikarabad - 501101, RR Dist., A.P., and utilizing his teaching skills, knowledge, experience and talent to achieve the goals and objectives of the Engineering College in the fullest perspective. He has attended more than 100 national and international conferences, seminars and workshops. He has more than 10 publications to his credit both in national and international journals. He is also having to his credit more than 50 research articles and paper presentations which are accepted in national and international conference proceedings both in India and Abroad. He can be contacted

Smt. Hemalatha Working as HOD in Computer Science Department in H.K.E.S. Sree Veerendra Patil Degree College, Sadashiva Nagar, Banglore, Karnataka, India. Her Areas of interests are Cloud Computing and Data Mining. She can be contacted