KTU NOTES

CLOUD COMPUTING- MODULE 4

Syllabus:

Part 1: Basic terms and concepts in security- Threat Agents, Cloud Security threats/risks, Trust.
Part 2: Operating System security- Virtual machine security-Security of virtualization-Security risks posed by shared images, Security risks posed by management OS.
Part 3: Infrastructure security - Network level security, Host level security, Application level security, Security of the physical systems. Identity & Access Management- Access Control.

Part 1

# Basic terms and concepts in security

Confidentiality

Confidentiality is the characteristic of something being made accessible only to authorized parties. Within cloud environments, confidentiality primarily pertains to restricting access to data in transit and storage.

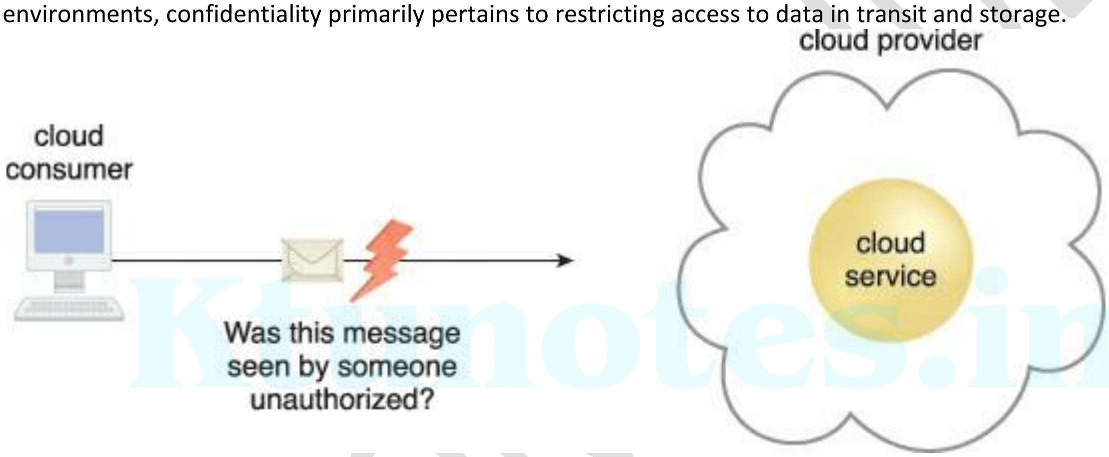


Figure: The message issued by the cloud consumer to the cloud service is considered confidential only if it is not accessed or read by an unauthorized party.

Integrity

Integrity is the characteristic of not having been altered by an unauthorized party. An important issue that concerns data integrity in the cloud is whether a cloud consumer can be guaranteed that the data it transmits to a cloud service matches the data received by that cloud service. Integrity can extend to how data is stored, processed, and retrieved by cloud services and cloud-based IT resources.

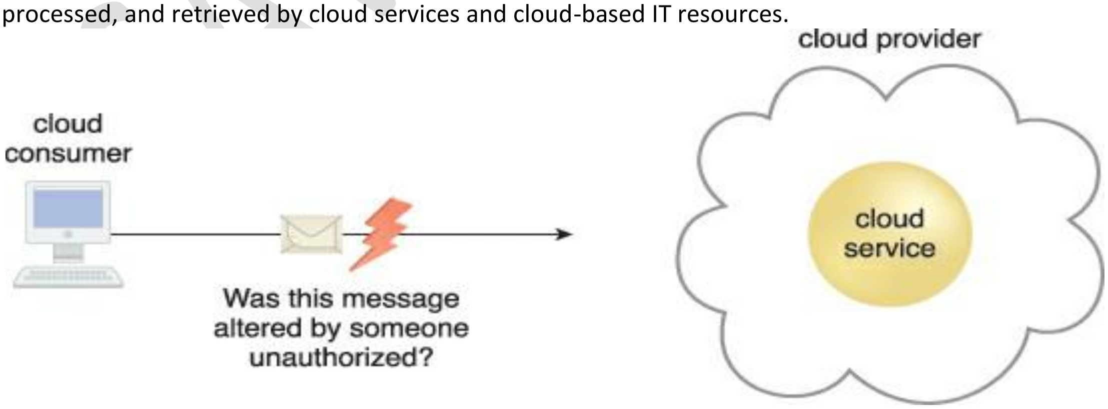


Figure: The message issued by the cloud consumer to the cloud service is considered to have integrity if it has not been altered.

Authenticity

Authenticity is the characteristic of something having been provided by an authorized source. This concept encompasses non-repudiation, which is the inability of a party to deny or challenge the authentication of an interaction. Authentication in non-repudiable interactions provides proof that these interactions are uniquely linked to an authorized source. For example, a user may not be able to access a non-repudiable file after its receipt without also generating a record of this access.

Availability

Availability is the characteristic of being accessible and usable during a specified time period. In typical cloud environments, the availability of cloud services can be a responsibility that is shared by the cloud provider and the cloud carrier. The availability of a cloud-based solution that extends to cloud service consumers is further shared by the cloud consumer.

### Threat

A *threat* is a potential security violation that can challenge defenses in an attempt to breach privacy and/or cause harm. Both manually and automatically instigated threats are designed to exploit known weaknesses, also referred to as vulnerabilities. A threat that is carried out results in an *attack*.

Vulnerability

A *vulnerability* is a weakness that can be exploited either because it is protected by insufficient security controls, or because existing security controls are overcome by an attack. IT resource vulnerabilities can have a range of causes, including configuration deficiencies, security policy weaknesses, user errors, hardware or firmware flaws, software bugs, and poor security architecture.

Risk

Risk is the possibility of loss or harm arising from performing an activity. Risk is typically measured according to its threat level and the number of possible or known vulnerabilities. Two metrics that can be used to determine risk for an IT resource are:

- the probability of a threat occurring to exploit vulnerabilities in the IT resource
- the expectation of loss upon the IT resource being compromised

Security Controls

Security controls are countermeasures used to prevent or respond to security threats and to reduce or avoid risk. Details on how to use security countermeasures are typically outlined in the security policy, which contains a set of rules and practices specifying how to implement a system, service, or security plan for maximum protection of sensitive and critical IT resources.

Security Mechanisms

Countermeasures are typically described in terms of security mechanisms, which are components comprising a defensive framework that protects IT resources, information, and services.

Security Policies

A security policy establishes a set of security rules and regulations. Often, security policies will further define how these rules and regulations are implemented and enforced. For example, the positioning and usage of security controls and mechanisms can be determined by security policies.

# Threat Agents

A *threat agent* is an entity that poses a threat because it is capable of carrying out an attack. Cloud security threats can originate either internally or externally, from humans or software programs. Corresponding threat agents are described in the upcoming sections. Figure illustrates the role a threat agent assumes in relation to vulnerabilities, threats, and risks, and the safeguards established by security policies and security mechanisms.

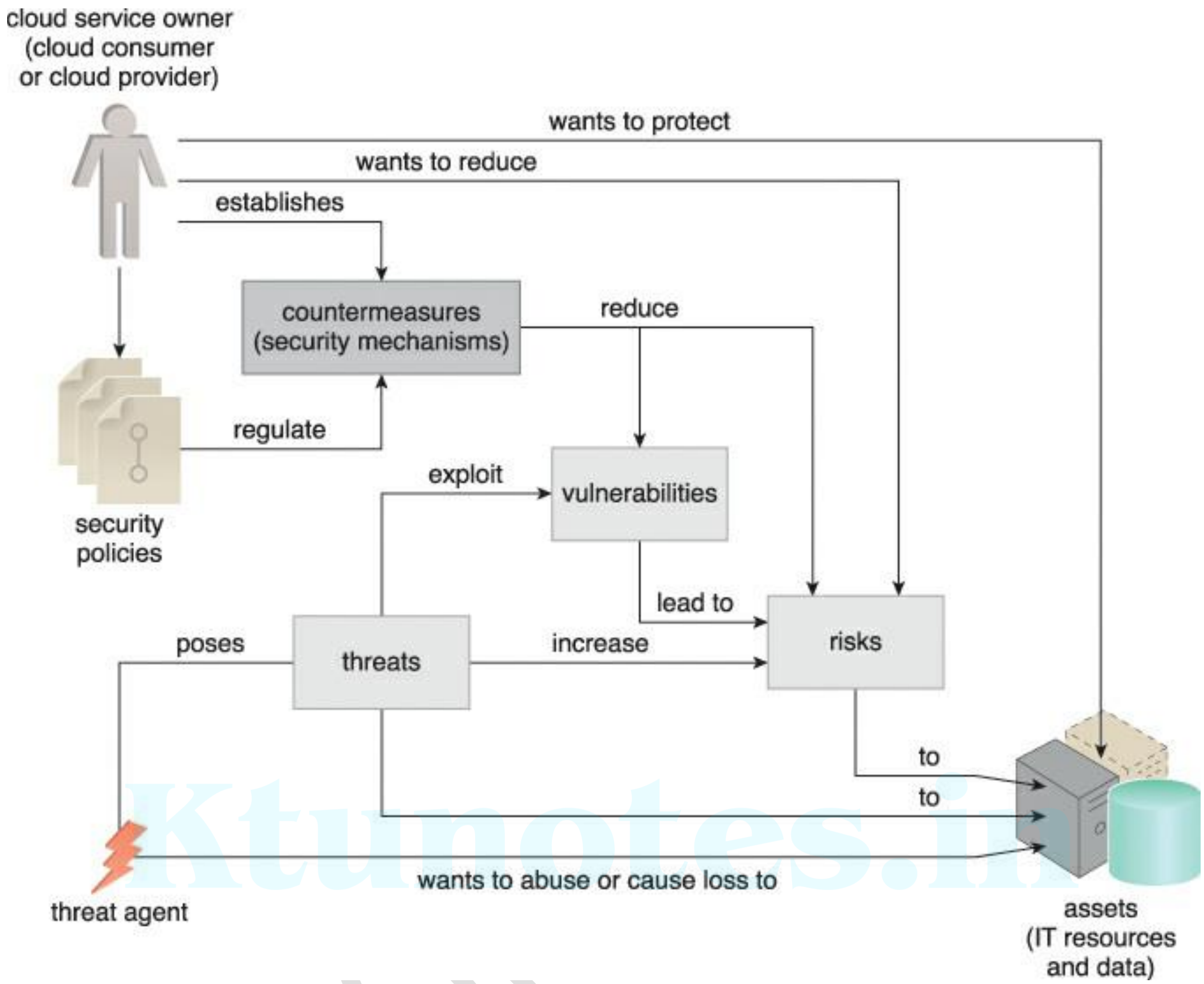

cloud service owner (cloud consumer or cloud provider)
wants to protect
wants to reduce
establishes
countermeasures (security mechanisms)
reduce
regulate
security policies
exploit
vulnerabilities
lead to
risks
poses
threats
increase
to
to
threat agent
wants to abuse or cause loss to
assets (IT resources and data)

Figure: How security policies and security mechanisms are used to counter threats, vulnerabilities, and risks caused by threat agents.

Anonymous Attacker

An *anonymous attacker* is a non-trusted cloud service consumer without permissions in the cloud (Figure 6.4). It typically exists as an external software program that launches network-level attacks through public networks. When anonymous attackers have limited information on security policies and defenses, it can inhibit their ability to formulate effective attacks. Therefore, anonymous attackers often resort to committing acts like bypassing user accounts or stealing user credentials, while using methods that either ensure anonymity or require substantial resources for prosecution.

Figure: The notation used for an anonymous attacker.

Malicious Service Agent

A *malicious service agent* is able to intercept and forward the network traffic that flows within a cloud. It typically exists as a service agent (or a program pretending to be a service agent) with compromised or malicious logic. It may also exist as an external program able to remotely intercept and potentially corrupt message contents.

Figure: The notation used for a malicious service agent.

Trusted Attacker

A *trusted attacker* shares IT resources in the same cloud environment as the cloud consumer and attempts to exploit legitimate credentials to target cloud providers and the cloud tenants with whom they share IT resources. Unlike anonymous attackers (which are non-trusted), trusted attackers usually launch their attacks from within a cloud's trust boundaries by abusing legitimate credentials or via the appropriation of sensitive and confidential information.

Trusted attackers (also known as *malicious tenants*) can use cloud-based IT resources for a wide range of exploitations, including the hacking of weak authentication processes, the breaking of encryption, the spamming of e-mail accounts, or to launch common attacks, such as denial of service campaigns.

Figure: The notation that is used for a trusted attacker.

Malicious Insider

Malicious insiders are human threat agents acting on behalf of or in relation to the cloud provider. They are typically current or former employees or third parties with access to the cloud provider's premises. This type of threat agent carries tremendous damage potential, as the malicious insider may have administrative privileges for accessing cloud consumer IT resources.

# Cloud Security Threats

Traffic Eavesdropping

Traffic eavesdropping occurs when data being transferred to or within a cloud (usually from the cloud consumer to the cloud provider) is passively intercepted by a malicious service agent for illegitimate information gathering purposes. The aim of this attack is to directly compromise the confidentiality of the data and, possibly, the confidentiality of the relationship between the cloud consumer and cloud provider. Because of the passive nature of the attack, it can more easily go undetected for extended periods of time.

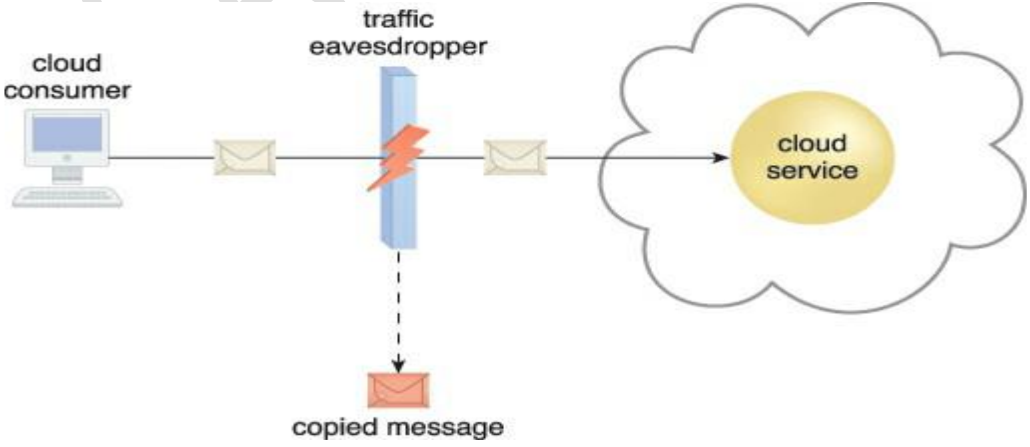


Figure: An externally positioned malicious service agent carries out a traffic eavesdropping attack by intercepting a message sent by the cloud service consumer to the cloud service. The service agent makes an unauthorized copy of the message before it is sent along its original path to the cloud service.

Malicious Intermediary

The *malicious intermediary* threat arises when messages are intercepted and altered by a malicious service agent, thereby potentially compromising the message's confidentiality and/or integrity. It may also insert harmful data into the message before forwarding it to its destination. Figure illustrates a common example of the malicious intermediary attack.

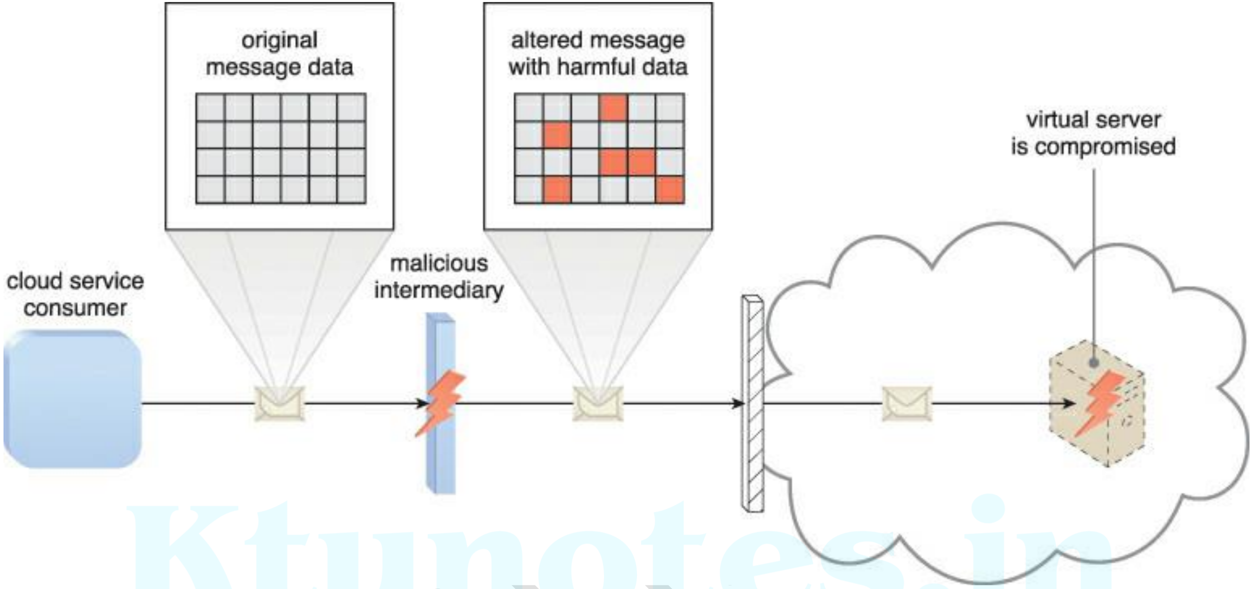


Figure: The malicious service agent intercepts and modifies a message sent by a cloud service consumer to a cloud service (not shown) being hosted on a virtual server. Because harmful data is packaged into the message, the virtual server is compromised.

Denial of Service

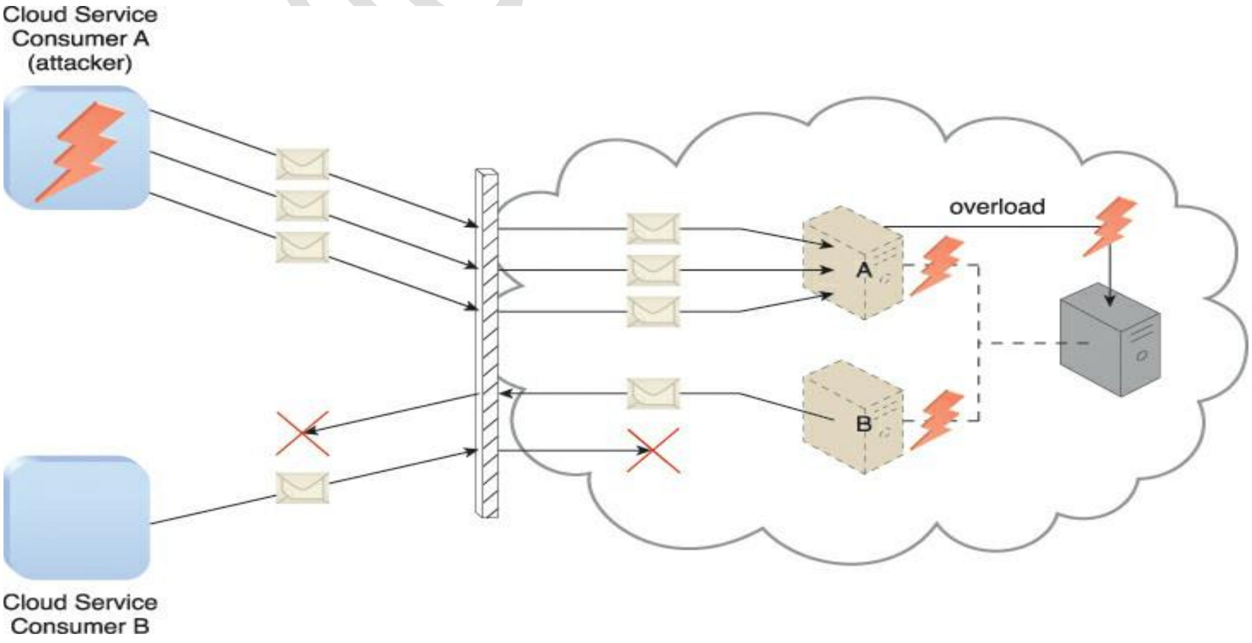


Figure: Cloud Service Consumer A sends multiple messages to a cloud service (not shown) hosted on Virtual Server A. This overloads the capacity of the underlying physical server, which causes outages with Virtual Servers A and B. As a result, legitimate cloud service consumers, such as Cloud Service Consumer B, become unable to communicate with any cloud services hosted on Virtual Servers A and B.

The objective of the denial of service (DoS) attack is to overload IT resources to the point where they cannot function properly. This form of attack is commonly launched in one of the following ways:

- The workload on cloud services is artificially increased with imitation messages or repeated communication requests.
- The network is overloaded with traffic to reduce its responsiveness and cripple its performance.
- Multiple cloud service requests are sent, each of which is designed to consume excessive memory and processing resources.

Insufficient Authorization

The insufficient authorization attack occurs when access is granted to an attacker erroneously or too broadly, resulting in the attacker getting access to IT resources that are normally protected. This is often a result of the attacker gaining direct access to IT resources that were implemented under the assumption that they would only be accessed by trusted consumer programs

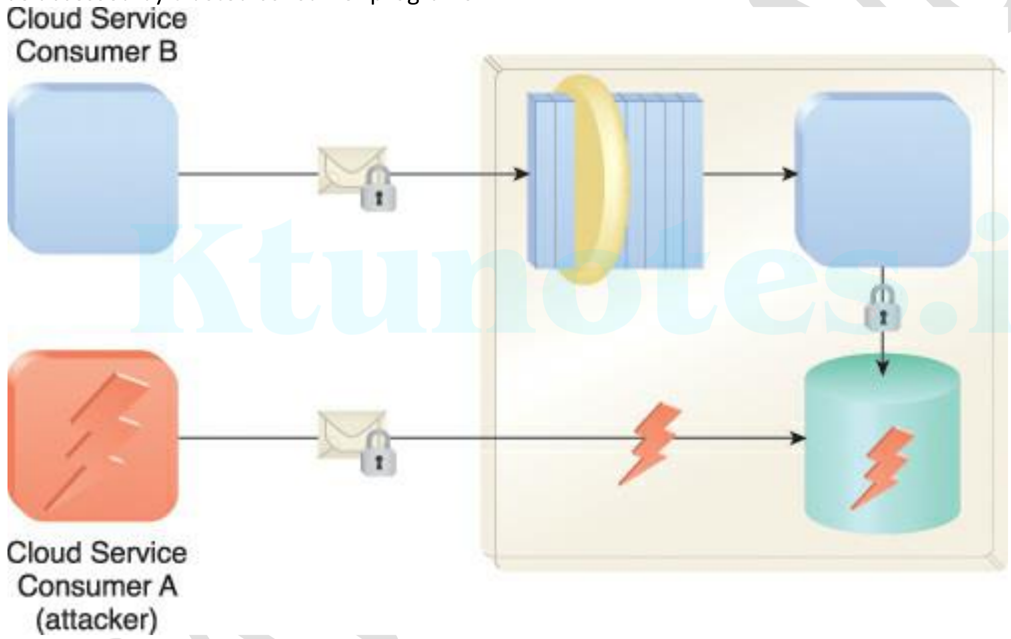


Figure: Cloud Service Consumer A gains access to a database that was implemented under the assumption that it would only be accessed through a Web service with a published service contract (as per Cloud Service Consumer B).

A variation of this attack, known as *weak authentication*, can result when weak passwords or shared accounts are used to protect IT resources. Within cloud environments, these types of attacks can lead to significant impacts depending on the range of IT resources and the range of access to those IT resources the attacker gains

Virtualization Attack

A *virtualization attack* exploits vulnerabilities in the virtualization platform to jeopardize its confidentiality, integrity, and/or availability. This threat is illustrated in Figure, where a trusted attacker successfully accesses a virtual server to compromise its underlying physical server. With public clouds, where a single physical IT resource may be providing virtualized IT resources to multiple cloud consumers, such an attack can have significant repercussions.

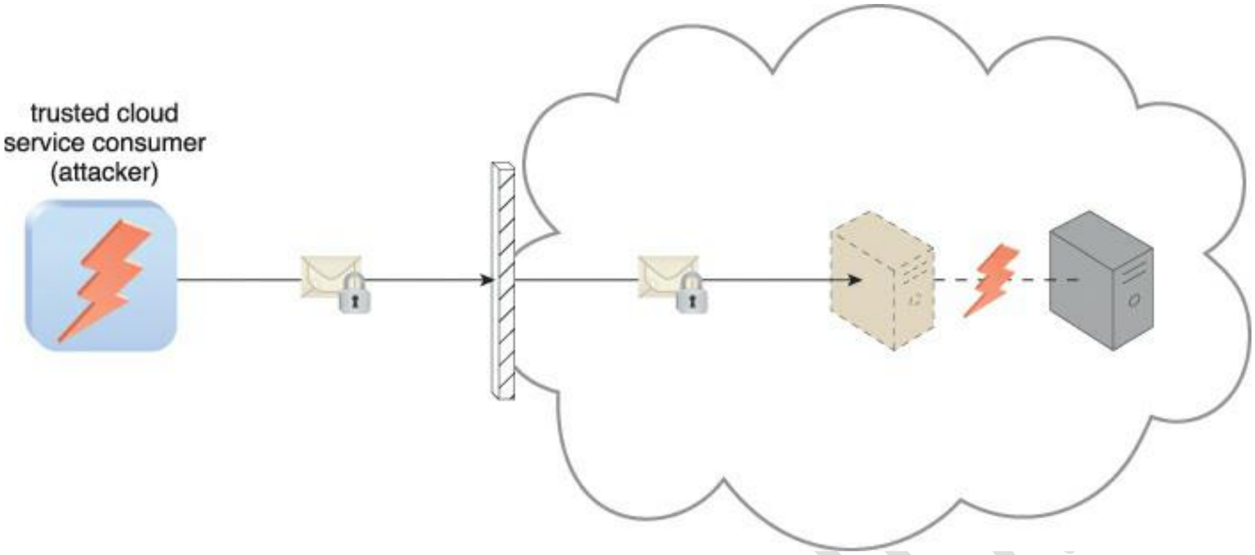


Figure: An authorized cloud service consumer carries out a virtualization attack by abusing its administrative access to a virtual server to exploit the underlying hardware.

Overlapping Trust Boundaries

If physical IT resources within a cloud are shared by different cloud service consumers, these cloud service consumers have overlapping trust boundaries. Malicious cloud service consumers can target shared IT resources with the intention of compromising cloud consumers or other IT resources that share the same trust boundary. The consequence is that some or all of the other cloud service consumers could be impacted by the attack and/or the attacker could use virtual IT resources against others that happen to also share the same trust boundary. Figure illustrates an example in which two cloud service consumers share virtual servers hosted by the same physical server and, resultantly, their respective trust boundaries overlap.

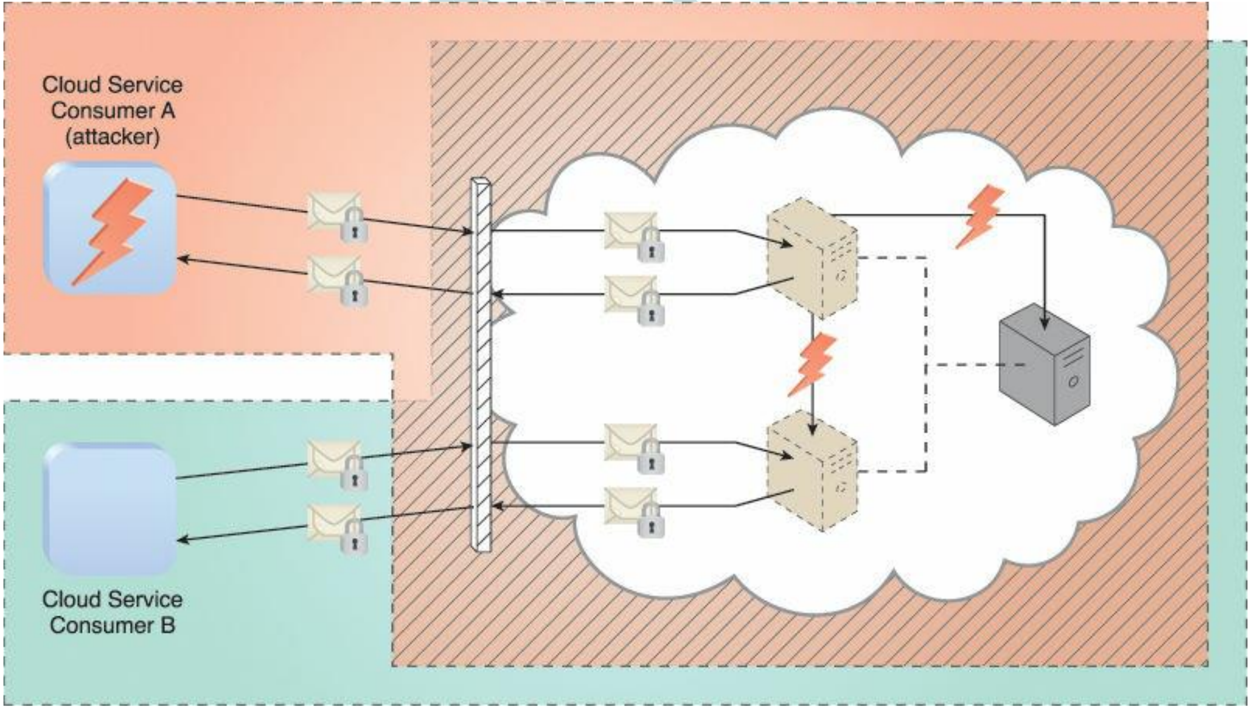


Figure: Cloud Service Consumer A is trusted by the cloud and therefore gains access to a virtual server, which it then attacks with the intention of attacking the underlying physical server and the virtual server used by Cloud Service Consumer B.

# Trust

According to the Merriam-Webster dictionary trust means “assured reliance on the character, ability, strength, or truth of someone or something.” Trust is a complex phenomenon, it enables cooperative behavior, promotes adaptive organizational forms, reduces harmful conflict, decreases transaction costs, facilitates formulation of ad hoc work groups, and promotes effective responses to crisis.

Two conditions must exist for trust to develop. The first is *risk*, the perceived probability of loss. Indeed, trust would not be necessary if there is no risk involved, if there is a certainty that an action can succeed. The second is *interdependence*, the interests of one entity cannot be archived without reliance on other entities.

A trust relationship goes though three phases:
1. Building phase, when trust is formed.
2. Stability phase, when trust exists.
3. Dissolution phase, when trust declines.

There are different reasons and forms of trust. Utilitarian reasons could be based on the belief that the costly penalties for breach of trust exceed any potential benefits from opportunistic behavior. This is the essence of *deterrence-based* trust. Another reason is the belief that the action involving the other party is in the self-interest of that party. This is the so-called *calculus-based* trust. After a long sequence of interactions *relational trust* between entities can developed based on the accumulated experience of dependability and reliance on each other.

Persistent trust is based on the long term behavior of an entity, while *dynamic trust* is based on a specific context, e.g., state of the system or the effect of technological developments.

Internet trust “obscures or lacks entirely the dimensions of character and personality, nature of relationship, and institutional character” of the traditional trust [360]. The missing identity, personal characteristics, and role definitions are elements we have to deal with in the context of online trust.

Policies and *reputation* are two ways of determining trust. Policies reveal the conditions to obtain trust, and the actions when some of the conditions are met. Policies require the verification of credentials. Reputation is a quality attributed to an entity based on a relatively long history of interactions or possibly observations of the entity.

PART-2

# Operating System Security

An operating system allows multiple applications to share the hardware resources of a physical system subject to a set of policies. A critical function of an OS is to protect applications against a wide range of malicious attacks such as unauthorized access to privileged information, tampering with executable code, and spoofing.

Access control, authentication usage, and cryptographic usage policies are all elements of the mandatory OS security. Access control policies specify how OS controls access to different system objects, authentication usage defines the authentication mechanisms used by the OS to authenticate a principal, and cryptographic usage policies specify the cryptographic mechanisms used to protect the data.

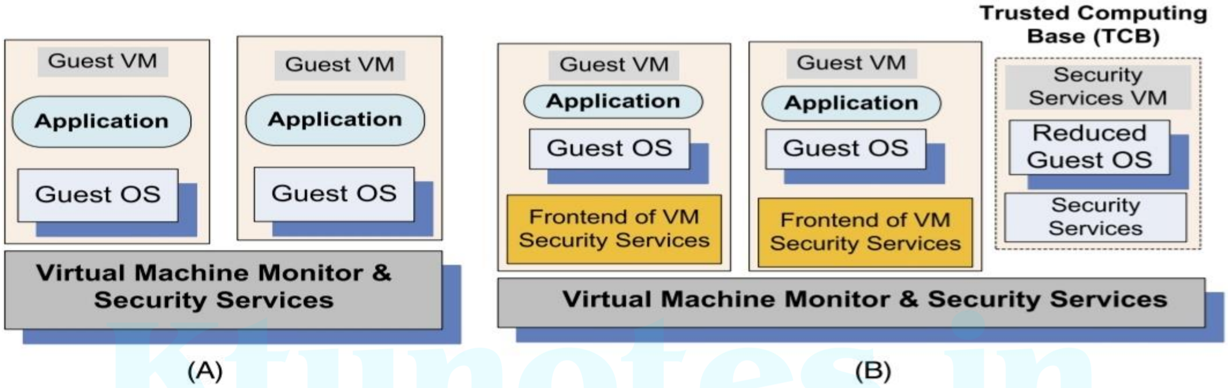
Applications with special privileges performing security-related functions are called *trusted applications.* Such applications should only be allowed the lowest level of privileges required to perform their functions. Commercial operating systems do not support multi-layered security. They only distinguish between a completely privileged security domain and a completely unprivileged one.

A highly secure operating system is necessary but not sufficient. Application-specific security is also necessary. Sometimes, security implemented above the operating system is better, e.g., electronic commerce requires a digital signature on each transaction.

An OS is a complex software system consisting of millions of lines of code and it is vulnerable to a wide range of malicious attacks. An OS poorly isolates one application from another; once an application is compromised, the entire physical platform and all applications running on it can be affected. Operating systems provide only weak mechanisms for applications to authenticate one another and do not have a trusted path between users and applications. These shortcomings add to the challenges of providing security in a distributed computing environment.

# Virtual Machine Security

Virtual security services are typically provided by the hypervisor as shown in Figure A; another alternative is to have a dedicated VM providing security service as in Figure B. A secure TCB (Trusted Computing Base) is a necessary condition for security in a VM environment. When the TCB is compromised then the security of the entire system is affected.


Trusted Computing Base (TCB)
Guest VM
Application
Guest OS
Virtual Machine Monitor & Security Services
(A)
Frontend of VM Security Services
Security Services VM
Reduced Guest OS
Security Services
(B)

Hypervisors are considerably less complex and better structured than traditional operating systems thus, in a better position to respond to security attacks. A major challenge is that a hypervisor sees only raw data regarding the state of a guest OS while security services typically operate at a higher logical level, e.g., at the level of a file rather than a disk block. A guest OS runs on simulated hardware and the hypervisor has access to the state of all VMs operating on the same hardware. The state of a guest VM can be saved, restored, cloned, and encrypted by the hypervisor.

We expect to pay some price for the better security provided by virtualization. This price includes: (i) higher hardware costs because a virtual system requires more resources such as CPU cycles, memory, disk, and network bandwidth; (ii) the cost of developing hypervisors and modifying the host operating systems in case of paravirtualization; and (iii) the overhead of virtualization as the hypervisor is involved in privileged operations.

NIST security group distinguishes two groups of threats, hypervisor-based and VM-based.

There are several types of hypervisor-based threats:

1. Starvation of resources and denial of service for some VMs. Probable causes: (a) badly configured resource limits for some VMs; (b) a rogue VM with the capability to bypass resource limits set in hypervisor.
2. VM side-channel attacks: malicious attack on one or more VMs by a rogue VM under the same hypervisor. Probable causes: (a) lack of proper isolation of inter-VM traffic due to misconfiguration of the virtual network residing in the hypervisor; (b) limitation of packet inspection devices to handle high speed traffic, e.g., video traffic; (c) presence of VM instances built from insecure VM images, e.g., a VM image having a guest OS without the latest patches.
3. Buffer overflow attacks.

There are also several types of VM-based threats:

1. Deployment of rogue or insecure VM; unauthorized users may create insecure instances from images or may perform unauthorized administrative actions on existing VMs. Probable cause: improper configuration of access controls on VM administrative tasks such as instance creation, launching, suspension, re-activation and so on.

2. Presence of insecure and tampered VM images in the VM image repository. Probable causes: (a) lack of access control to the VM image repository; (b) lack of mechanisms to verify the integrity of the images, e.g., digitally signed image.

# Security of Virtualization

The complete state of an operating system running under a VM is captured by the VM. The VM state can be saved in a file and then the file can be copied and shared. There are several useful implications of this important virtue of virtualization:

- Supports the IaaS delivery model. An IaaS user selects an image matching the local application environment and then uploads and runs the application on the cloud using this image.
- Increased reliability. An operating system with all the applications running under it can be replicated and switched to a hot standby in case of a system failure. Recall that a hot standby is a method to achieve redundancy. The primary and the backup systems, run simultaneously and have identical state information.
- Straightforward mechanisms for implementing resource management policies. An OS and the applications running under it can be moved to another server to balance the load of a system.
- Improved intrusion detection. In a virtual environment a clone can look for known patterns in system activity and detect intrusion. The operator can switch a server to hot standby when suspicious events are detected.
- Secure logging and intrusion protection. When implemented at the OS level intrusion detection can be disabled and logging can be modified by an intruder. When implemented at the hypervisor layer, the services cannot be disabled or modified. In addition, the hypervisor may be able to log only events of interest for a post-attack analysis.
- More efficient and flexible software maintenance and testing. Virtualization allows the multitude of OS instances to share a small number of physical systems, instead of a large number of dedicated systems running under different operating systems, different versions of each OS, and different patches for each version.

Undesirable effects of virtualization lead to a diminished ability of an organization to manage its systems and track their status. These undesirable effects are:

- The number of physical systems in the inventory of an organization is limited by cost, space, energy consumption, and human support. The explosion of the number of VMs is a fact of life; to create a VM one simply copies a file. The only limitation for the number of VMs is the amount of storage space available.
- There is also a qualitative side to the explosion of the number of VMs. Traditionally, organizations install and maintain the same version of system software. In a virtual environment such a uniformity cannot be enforced, the number of different operating systems, their versions, and the patch status of each version will be diverse and the diversity will tax the support team.
- One of the most critical problems posed by virtualization is related to the software lifecycle. The traditional assumption is that the software lifecycle is a straight line, hence the patch management is based on a monotonic forward progress. The virtual execution model *maps to a tree structure* rather than a line. Indeed, at any point in time multiple VM instances can be created and then each one of them can be updated, different patches installed, and so on. This problem has serious implication on security as we shall see shortly.

A more general observation is that in a traditional computing environment a steady state can be reached. In this steady state all systems are brought up to a “desirable” state, whereas “undesirable” states, states when some of the systems are either infected by a virus or display an undesirable pattern of behavior, are only transient. The desirable state is reached by installing the latest system software version and then applying the latest patches to all systems.

A virtual environment may never reach such a steady state due to the lack of control. In a non-virtual environment the security can be compromised when an infected laptop is connected to the network protected by a firewall, or when a virus is brought in on a removable media. But, unlike a virtual environment, the system can still reach a steady state.

# Security Risks Posed By Shared Images

Image sharing is critical for the IaaS cloud delivery model. For example, an AWS user has the option to choose between Amazon Machine Images (AMIs) accessible through the Quick Start or the Community AMI menus of the EC2 service. To use an image, a user has to specify the resources, provide the credentials for login, a firewall configuration, and specify the region. Many analyzed images allowed a user to *undelete* files, recover credentials, private keys, or other types of sensitive information with little effort, using standard tools.

A study was able to audit some 5 303 images out of the 8 448 Linux AMIs and 1 202 Windows AMIs at Amazon sites in the US, Europe and Asia. The audit covered software vulnerabilities and security and privacy risks.
The *software vulnerability* audit revealed that 98% of the Windows AMIs (249 out of 2 53) and 58% (2 005 out of 3 432) Linux AMIs audited had critical vulnerabilities. The audit reported only vulnerabilities of the highest severity level, e.g., remote code execution. Three types of *security risks* are analyzed: (1) backdoors and leftover credentials, (2) unsolicited connections, and (3) malware. An astounding finding is that about 22% of the scanned Linux AMIs contained credentials allowing an intruder to remotely login to the system.

An attacker can impersonate the agents at both ends of a communication channel in the *man-in-the middle* attack and makes them believe that they communicate through a secure channel. For example, if B sends her public key to A, but C is able to intercept it, such an attack proceeds as follows: C sends a forged message to A claiming to be from B, but instead includes C's public key. Then A encrypts her message with C's key, believing that she is using B's key, and sends the encrypted message to B. The intruder, C, intercepts, deciphers the message using her private key, possibly alters the message, and re-encrypts with the public key B originally sent to A. When B receives the newly encrypted message, she believes it came from A.

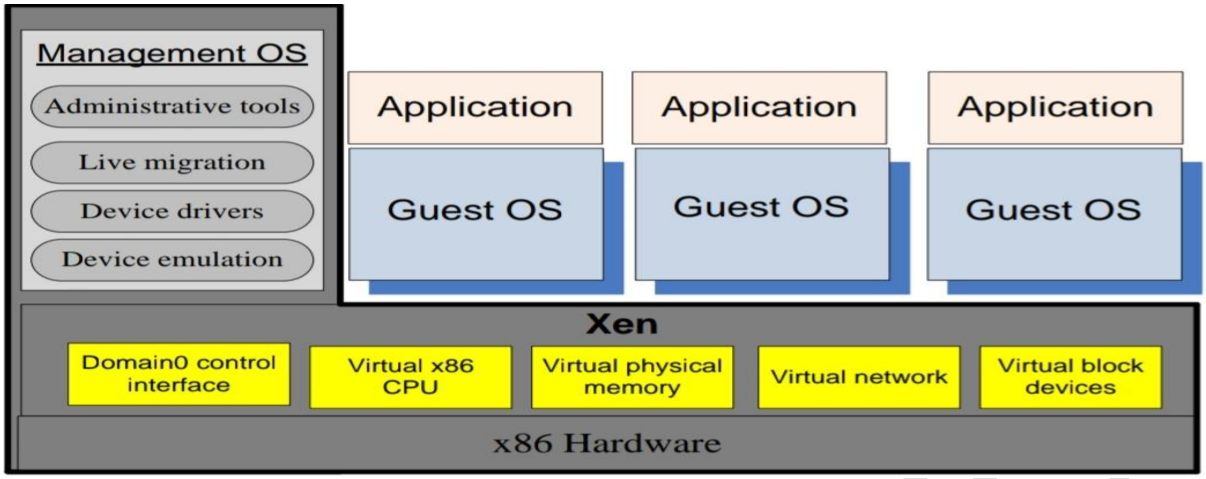
Recovery of deleted files containing sensitive information poses another risk for the provider of an image. When the sectors on the disk containing sensitive information are actually overwritten by another file, recovery of sensitive information is much harder. To be safe, the creator of the image effort should use utilities such as shred, scrub, zerofree or wipe to make recovery of sensitive information next to impossible. If the image is created with the block-level tool discussed at the beginning of this section the image will contain blocks of the file system marked as free; such blocks may contain information from deleted files. The audit process was able to recover files from 98% of the AMIs using the *exundelete* utility. The number of files recovered from an AMI were as low as 6 and as high as 40 000. We conclude that the users of published AMIs as well as the providers of images may be vulnerable to a wide range of security risks and must be fully aware of the dangers posed by image sharing.

# Security Risks Posed By Management OS

A VM monitor or hypervisor is considerably smaller than an operating system. For example, the Xen hypervisor has approximately 60 000 lines of code. The Trusted Computer Base (TCB)6 of a cloud computing environment includes not only the hypervisor but also the management OS. The management OS supports administrative tools, live migration, device drivers, and device emulators.
Xen management operating system runs in Dom0; it manages the building of all user domains, a process consisting of several steps:

- Allocate memory in the Dom0 address space and load the kernel of the guest OS from secondary storage.
- Allocate memory for the new VM and use foreign mapping to load the kernel to the new VM. The foreign mapping mechanism of Xen is used by Dom0 to map arbitrary memory frames of a VM into its page tables.
- Set up the initial page tables for the new VM.
- Release the foreign mapping on the new VM memory, set up the virtual CPU registers, and launch the new VM.

Figure below describes The trusted computing base of a Xen-based environment includes the hardware, Xen, and the management operating system running in Dom0. The management OS supports administrative tools, live migration, device drivers, and device emulators. A guest OS and applications running under it reside in a DomU.



A malicious Dom0 can play several nasty tricks at the time when it creates a DomU

- Refuse to carry out the steps necessary to start the new VM, an action that can be considered a *denial-of-service* attack.
- Modify the kernel of the guest OS in ways that will allow a third party to monitor and control the execution of applications running under the new VM.
- Undermine the integrity of the new VM by setting the wrong page tables and/or setup wrong virtual CPU registers.
- Refuse to release the foreign mapping and access the memory while the new VM is running.

How to deal with run time vulnerability of DomO?

To implement a secure run-time system we have to intercept and control the hypercalls used for communication between a Dom0 that cannot be trusted and a DomU we want to protect.

New hypercalls are necessary to protect:

- The privacy and integrity of the virtual CPU of a VM. When Dom0 wants to save the state of the VM the hypercall should be intercepted and the contents of the virtual CPU registers should be encrypted. The virtual CPU context should be decrypted and then an integrity check should be carried out when DomU is restored.
- The privacy and integrity of the VM virtual memory. The *page table update* hypercall should be intercepted and the page should be encrypted so that Dome handles only encrypted pages of the VM. The hypervisor should calculate a hash of all the memory pages before they are saved by Dom0 to guarantee the integrity of the system. An address translation is necessary because a restored DomU may be allocated a different memory region.
- The freshness of the virtual CPU and the memory of the VM. The solution is to add to the hash a version number.

PART- 3

# Infrastructure security

Infrastructure security describes the issues related with controlling access to physical resources which support the cloud infrastructure. Infrastructure security can be classified into three categories like network level, host level and service level.

Network Level Security

- The network-level security risks exist for all the cloud computing services (e.g., SaaS, PaaS or IaaS). It is actually not the service being used but rather the cloud deployment type (public, private or hybrid) that determine the level of risk.
- Ensuring data confidentiality, integrity and availability are the responsibilities of network level infrastructure security arrangement. Data confidentiality risk is generally reduced by using techniques like

encryption and digital signatures but data availability problem at the network level causes more difficulty and needs more attention to manage.

- If an organization can afford on-premises private cloud to meet their business needs, their network level security risks naturally decreases. Here it should be noted that the network-level security of private cloud deployed as on-premises or at some provider's facility depends on the potential of the infrastructure architect, either be it developed by some third party or the enterprise itself.
- Most of the network-level security challenges are not new to cloud; rather, these have existed since the early days of Internet. Advanced techniques are always evolving to tackle these issues.

Host Level Security

At cloud service provider's end, the 'host' refers to the physical machines. weak implementation of access control mechanism to the hypervisor may create trouble for physical hosts. VM escape problem may also cause damage to physical hosts as virtual machines are a little prone to this particular security threat as associated to virtualization technology.

The responsibilities of the host-level security management are:

- *For SaaS and PaaS consumers*: Service providers would not publicly share details regarding their host platforms like operating systems or security management mechanisms to secure the hosts. Otherwise, hackers may exploit those details to break the security.
- One difference between PaaS and SaaS consumers arises from the difference in access right to the abstraction layer that covers the OS on which applications they run. This abstraction layer is not accessible by the SaaS consumers but is accessible by the developers who are actually the PaaS consumers. But, the PaaS users cannot access this abstraction layer directly; rather they are given indirect access to the layer through the application program interface (API) of PaaS application.
- In general, the security responsibility of hosts in PaaS and SaaS services largely depends on the service providers.
- *For IaaS consumers*: Unlike PaaS and SaaS, IaaS consumers have the shares of responsibility in securing the host. Service providers use to take care of the security of physical resources through abstraction. But IaaS consumers must take care that no malicious application could try to break it.

Application Level Security

Both the consumer and service providers have their share of responsibilities of security management at this level so that no application can harm to the infrastructure.

- **IaaS Application Security:** At IaaS level, the users are largely accountable for managing and securing the virtual servers they work with, along with the providers. At this level, the virtual servers (which are delivered by IaaS service providers) are owned by customers, and the IaaS providers blindly serve the applications running over those virtual servers with full trust without verifying any threats. Therefore, the major responsibility of security management of virtual resources at this layer is task of consumers as well.
- **PaaS Application Security:** The security issues can be divided into two stages at the PaaS application level: Security of the PaaS platform itself and Security of consumers' applications deployed on a PaaS application. PaaS service providers are responsible for securing the platform software stack on which consumers deploy or develop their applications. Security management of these applications deployed on PaaS is consumer's prime responsibility, although PaaS providers take care of any kind of dependencies.
- **SaaS Application Security:** In SaaS model, it is the responsibility of the provider to manage the complete set of applications they deliver to consumers. Therefore, the SaaS providers must take suitable measures to make their offering secure so that consumers with ill intention cannot cause harm to them. From the consumer's viewpoint, the use of SaaS reduces lots of tensions.

Safety and Security of the Physical Systems

Apart from the above issues, the IaaS service providers are responsible to take care of some other issues to ensure reliable cloud services. Any general, physical or technical problems may cause the physical servers to go down and hence the cloud service loses its pace as well. This would be a loss of availability of service which harms the business. The following issues should be taken care of to ensure uninterrupted availability of cloud services:

■ Facility of uninterruptible power supply (UPS).

■ Proper safety measures against fire to minimize the loss in case of disaster.

■ Adequate cooling and ventilation facility.

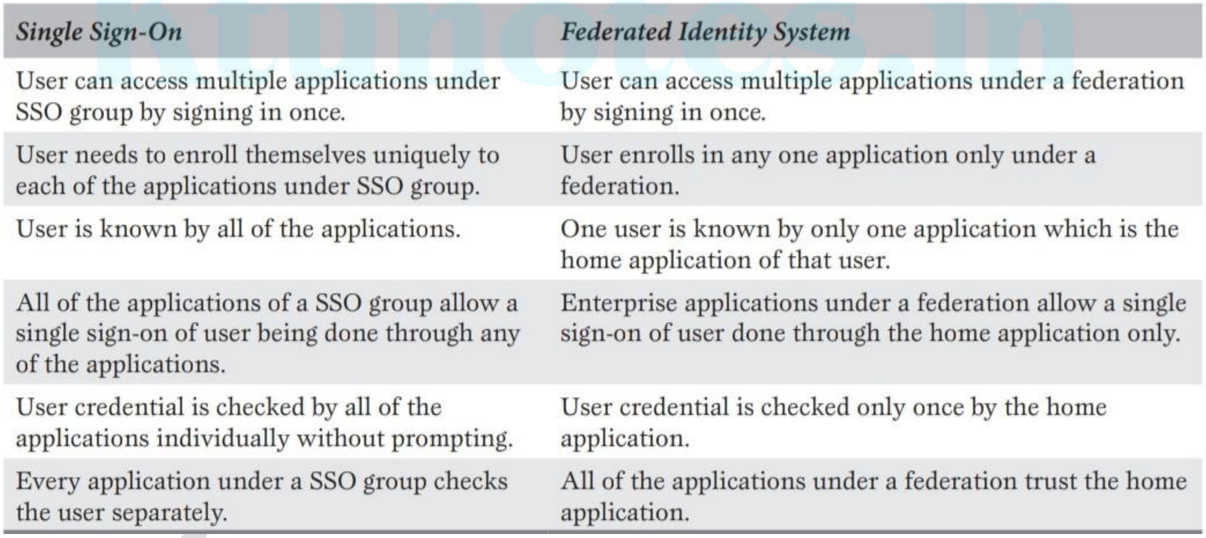
■ Stringent restriction on physical access to the servers. Unauthorized persons must not have any access to the area.
■ The physical protections listed above should also be maintained for all of the network related devices (such as routers) and cables.

# IDENTITY MANAGEMENT AND ACCESS CONTROL

Identity management and *access control* (often termed as *identification and access management* or IAM) are primary functionalities needed for any secure computing system. The benefits of identification and access management are:
■ Proper execution of IAM technique improves a system's operational efficiency through automation of user verification process.
■ It protects a system and enhances the security of its application and information against harmful attacks.

Service providers must provide utmost effort towards implementing identity management and access control mechanisms to protect their cloud computing environment from any malicious activities. Specially in public cloud environment this becomes very critical as the entire computing environment resides at some remote place outside the *network boundary* of consumer organization. From consumer organization's end, this loss of network control can be compensated by the implementation of proper user access control techniques, like authentication and authorization.

Maintaining multiple usernames and passwords to access different applications, especially in the enterprise applications, reduce productivity and hampers application adoption. SSO and federated identity solve this problem by integrating applications and eliminating the need for multiple usernames and passwords without compromising security.



Single Sign-On	*Federated Identity System*
User can access multiple applications under SSO group by signing in once.	User can access multiple applications under a federation by signing in once.
User needs to enroll themselves uniquely to each of the applications under SSO group.	User enrolls in any one application only under a federation.
User is known by all of the applications.	One user is known by only one application which is the home application of that user.
All of the applications of a SSO group allow a single sign-on of user being done through any of the applications.	Enterprise applications under a federation allow a single sign-on of user done through the home application only.
User credential is checked by all of the applications individually without prompting.	User credential is checked only once by the home application.
Every application under a SSO group checks the user separately.	All of the applications under a federation trust the home application.

Access control is basically a procedure or policy that allows, disallows or limits access of users to a system. Access control is inherently attached with identity management and is necessary to defend the confidentiality, integrity and availability of data.

Several access control models are there in practice like *mandatory access control* (MAC), *discretionary access control* (DAC) and *non-discretionary access control*. These models are known as identity-based access control models where the users (and sometimes processes) are called subjects, resources are called objects and they are identified uniquely (generally by unique Ids).

Mandatory Access Control

In mandatory access control (MAC), access policies are controlled by a central authority. Here the system (and not the users) specifies the access rule. Subjects are assigned into some classification levels (like management, administrator, official and friendly) and objects are assigned into some protection levels (like top-secret, private, moderate and friendly). In this example, the classification and protection levels are mentioned in descending order of grade in terms of security. A subject can be a group, department, project or process.

When the system makes an access control decision, it tries to match the classification level of the subject with the protection level of the object. If a user has a lower classification level than the protection level of the object he is trying to access, the access will be denied.

Adoption of MAC ensures a safe and secure access control environment but that comes with some cost as well. Since it is controlled from the system level, planning of the access rules requires a considerable amount effort. The MAC model is usually used in environments where the confidentiality is of utmost importance.

Discretionary Access Control

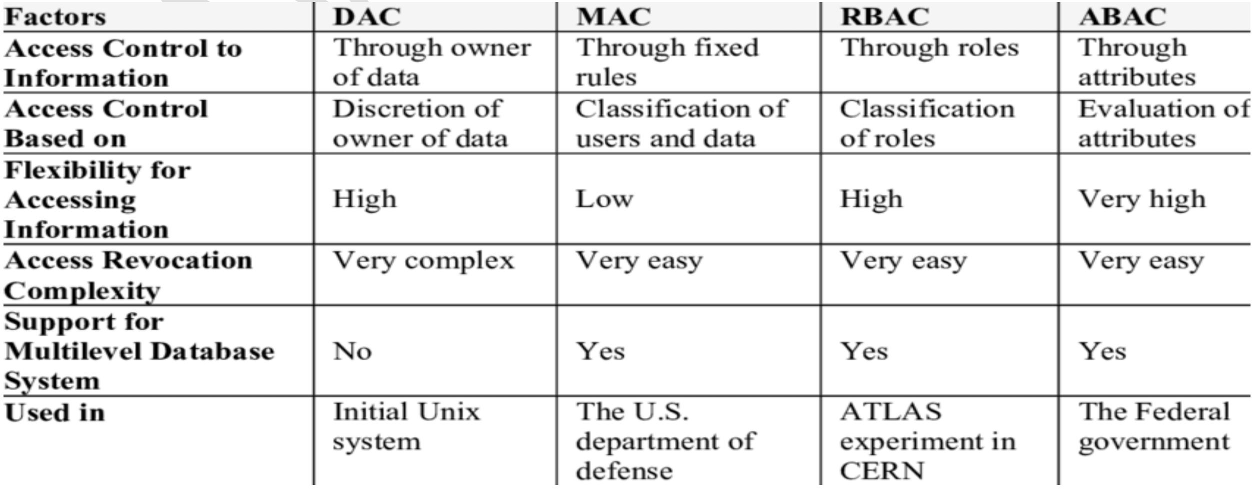
Discretionary Access Control (DAC) is not so restrictive like MAC as it is more open. In DAC, the access policies are controlled by owner of object. There are no central rules to control access, rather object owners can decide which subjects will be given access of an object. The model is called discretionary as the access permission of objects depends on the discretion of owners. Owner of object has to determine what kind of privileges he/she is going to provide to a user. If appropriate privilege is given, a user in turn will be able to pass on access rights to other users.

Unlike the protection levels of MAC system, DAC-based system maintains a list containing identity of objects and their access permissions. This list is maintained for each and every object separately and is called as *access control list* (ACL). Although DAC is more flexible than MAC, but this approach has some risk associated with it if not planned properly because the users have the privileges of granting access permissions to the objects.

Non-discretionary Access Control

In non-discretionary access control mechanism, the access policies are determined based on user’s role. At first, roles are defined according to different possible job profiles or responsibilities in an organization. These roles assign access privileges. Later with each user as added into the system, they are assigned with appropriate roles. Access privilege of user is defined with this role assignment. Here, the assignments of access controls to subjects do not depend on discretionary decisions of the owner of an object, rather it complies with the organization’s guidelines.

These access control policies are non-discretionary in the sense that they are imposed on all of the users based on their roles. Among those popular access control methods often been heard in enterprise applications, the *Role-Based Access Control* (RBAC) is actually nondiscretionary access control mechanism. Roles in non-DAC differ from groups in MAC in the sense that while a user may belong to multiple groups, he/she can be assigned only a single role within an organization.



Factors	DAC	MAC	RBAC	ABAC
Access Control to Information	Through owner of data	Through fixed rules	Through roles	Through attributes
Access Control Based on	Discretion of owner of data	Classification of users and data	Classification of roles	Evaluation of attributes
Flexibility for Accessing Information	High	Low	High	Very high
Access Revocation Complexity	Very complex	Very easy	Very easy	Very easy
Support for Multilevel Database System	No	Yes	Yes	Yes
Used in	Initial Unix system	The U.S. department of defense	ATLAS experiment in CERN	The Federal government