A Novel Attribute Based Access Control Model With Application in Iaas Cloud

[1]DILAWAR SINGH, [1]SHWETA SINHA, [2]VIKAS THADA
[1]ASET Amity University, Gurugram, Haryana, INDIA
[2]DCSE, MITRC, Alwar, Rajasthan, INDIA

Abstract: Cloud computing is viewed as one of the most dominant ideal models in the Information Technology industry nowadays. It offers new savvy administrations on-request like Software as a Service, Infrastructure as a Service, and Platform as a Service. Nonetheless, with these administrations promising offices and advantages, there are yet various difficulties related to using cloud computing, for example, data security, maltreatment of cloud administrations, malicious insiders, and cyber-attacks. Among all security necessities of cloud computing, access control is one of the fundamental prerequisites to keep away from unapproved access to frameworks and safeguard association's resources. Albeit different access control models and policies have been grown for various conditions, these models may not satisfy the cloud's access control necessities. It used a portion of the PM's parts alongside a proof-of-idea execution to implement ABAC augmentation for OpenStack while keeping OpenStack's present RBAC design set up. This gives the advantages of upgrading access control flexibility with help of client attributes while limiting the upward of changing the current OpenStack access control structure. The use cases are presented to portray added advantages of the proposed model and show authorization results. At this point, it assesses the exhibition of the proposed ABAC augmentation and examine its applicability and conceivable execution upgrades.

Keywords: Attribute, Based, Access, Control, Model, IAAS, Cloud

Received: June 26, 2021. Revised: June 9, 2022. Accepted: July 15, 2022. Published: September 26, 2022.

1. Introduction

Attribute Cloud-based IaaS access control

In general, barriers are an important and unimaginable tool to advertise higher levels of approach safety. For example, an organization's goals may establish higher-level policies to limit the powers of delegated authorities, such as: "Computer programmer" and "analyzer" professions for the same project. Ultimately, that foundation prevents the worker from working at the same time creating and testing code for the same project. In this proposal, the inclusion of requirement details in attribute-based access control (ABAC) and cloud system as a service (Cloud SaaS) (IaaS), ABAC, for the most part, aggressively monitors customer consent or the subject to access the resources of the framework. based on binding approval standards associated with a specific authorization.

Virtualization technology is one of the most important building blocks in delivering IaaS. Therefore, the competence of the access control framework is affected by the hypervisor. The hypervisor introduces a security risk related to controlling access to virtual machines (VMs) due to the single point of access. A trusted virtual machine running on an untrusted hypervisor has a higher risk of failure than an untrusted virtual machine running on a trusted hypervisor. For example, based on observational research in a real-time virtual machine (LVMM) migration process, it was recommended to review the current access control framework to prevent unauthorized access during the LVMM process. The interaction between LVMM and the virtual machine is considered one of the fundamental cycles in IaaS and occurs immediately after the virtual machine provisioning process. Additionally, access control security solutions in IaaS, such as firewalls and security clusters, cannot support security-aware policy creation or mechanisms. Because sensitive data can be exposed to unauthorized substances, much attention is paid to improving the trade-off between data flow security and IaaS flexibility. In this context, a well-thought-out IaaS access control system is mandatory.

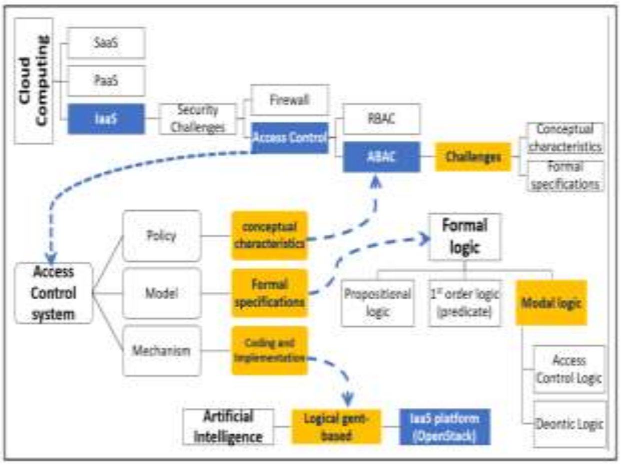


Fig. 1. General components of the investigation

According to the report, security considerations are key to convincing customers to use cloud computing administrations. For example, according to an IDC survey, 87% of customers cite satisfaction with the level of security and protection as their top reason for using cloud computing management. In particular, the access control perspective is a key security concern in the IaaS cloud environment. Unlike the traditional computing environment, the IaaS cloud has explicit features such as adaptability, multi-tenant, configurability, and dynamics, among others. As a result, traditional access control models face flexibility challenges and granular limitations when it comes to running and configuring them in IaaS.

2. Related Work

It addressed access control issues in Infrastructure as a Service. Updating IaaS access control as shown in Figure 2 is believed to be the recommended solution for some security issues. Because IaaS is a multi-tenant environment, it must meet a wide range of customer access needs. Therefore, the IaaS access control framework must be designed to allow fine-grained policy implementation.

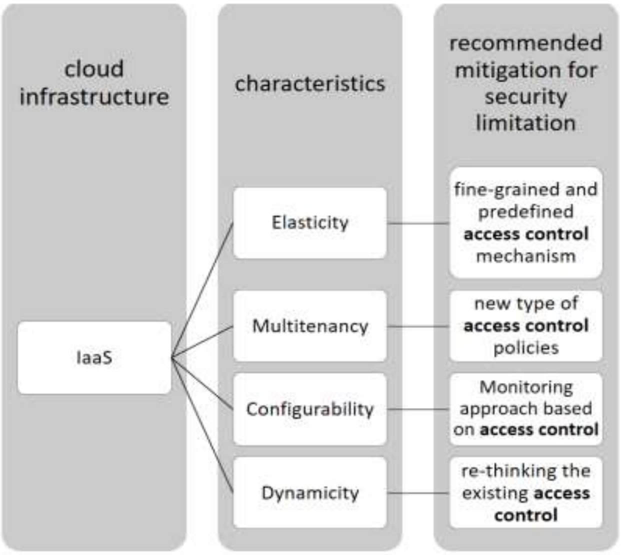

cloud infrastructure
characteristics
recommended mitigation for security limitation
IaaS
Elasticity
Multitenancy
Configurability
Dynamicity
fine-grained and predefined **access control** mechanism
new type of **access control** policies
Monitoring approach based on **access control**
re-thinking the existing **access control**

Fig. 2. Recommended mitigation for specific IaaS security challenges

2.1 ABAC in the IaaS cloud

The ABAC model concept is aware in the sense that you can use the concept of attributes in your selection of permissions. However, ABAC does not display pop-up highlighting in this scenario. Under the same circumstances, the designer created ABAC offers some configuration suggestions designed to merge advanced RBAC models. In addition to highlighting framework conditions, protection, and trust, Smari et al. developed ABAC to include areas where the environment is linked to an access control approach along with issues and protests. Neither the evaluation of the complexity of the framework nor the formalization of the strategic language have been outlined in detail so far. HGABAC is a model proposed by Servos and Osborn in which the ABAC model considers hierarchical components instead of the abacus. Servos and Osborn incorporated some environmental awareness into their design by considering the association between organization and environmental features. However, they did not take into account the concept of destination and the emphasis on delegation.

2.2 Formal logic in ABAC

The focus language used in HGABAC was developed based on the considerations of Kleene K3. The conventional dialects used in ABAC and HGABAC are themselves a kind of propositional logic. During a strategy update or strategy research, you are faced with the problem of NP-complete satisfaction. Whichever conventional language based on the logic of the first requirement is used to express the ABAC technique, it will face an undecidable computational task regardless of the language used. Wang et al. developed the programming language for requirements reasoning and used it to implement the meaning-based framework for ABAC, which is considered the first commercial application of the framework. Bijon and his colleagues in their paper (ABCL) developed a crucial language for the details of attribute-based constraints. Thus, the crucial language is explicit, as it focuses on the specified required benefits of the capabilities contained in the ABAC model, as opposed to the implicit language. ABCL, on the other hand, was unable to assess the benefits of the functionality of the authorization policy rules.

2. 3 An ABAC Extension for Openstack

It proposed a role based ABAC model for OpenStack by extending the proven OSAC mod.

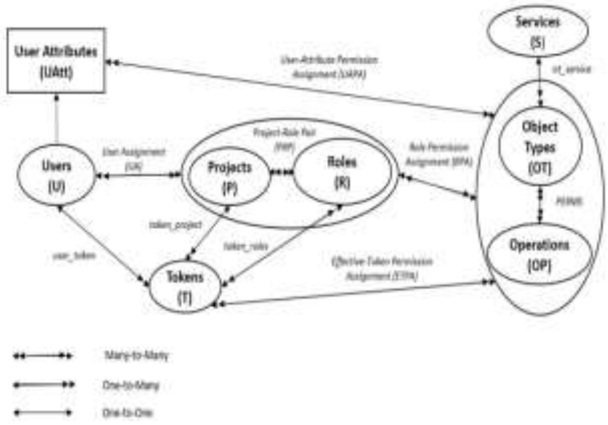


Figure 3: OSAC with extended user attribute on a single tenant

This model is called the Extended OSAC Model with User Attributes and is shown in Figure 3 as the Extended OSAC Model with User Attributes. It contains all the key and specified components of the extended OSAC model, as well as some recently added elements and relationships. The client attributes are added to the user table and a new UAPA connection is provided to determine the value of the client attributes and perform authorization tasks. Because it integrates with the current OpenStack RBAC system and retains all of its benefits, this model is known as the ABAC role-based model. It also offers the flexibility of an ABAC model by displaying customer characteristics. The most restrictive permissions are determined by the client's responsibilities and are further limited by the client's attribute authorization activities.

Attribute functions are functions that take the client and return a scope-specific value, where the scope of an attribute is a restricted set of atomic properties specific to each attribute function. In general, there are two types of predicted attributes: estimated atomic attributes return only a value of their scope, while defined predicted attributes return a subset of the values of their scope. Customer attributes are features or characteristics of the customer; For example, Department, Authorization, and Specialization are sample models. OSAC's extended user attribute paradigm only allows clients to have attributes that are evaluated atomically. UAPA is a set of permissions associated with customer attributes and features assigned to them. The ETPA was amended to include approvals for customer properties, regardless of whether or not the customer has given approval.

2.4 Cloud Services

SaaS is one of the main benefits of the upcoming cloud (software as a service) IaaS (infrastructure as a service) PaaS (platform as a service) (infrastructure as a service) Each of these types of cloud services includes the purchase of some type of administration, such as B. Programming, cycle planning, etc., by customers. This can be free or pay as you go if you are using a pay as you go model.

software as a service

Software as a Service (SaaS) is a type of cloud management in which software programs are distributed and accessed over the Internet. It is accessible to the end user through software that acts as a point of interaction. This arrangement allows for a technique where there is no need to install any software on client computers. This relieves customers of the obligation to purchase, update, or maintain the software they use, as support providers take care of it. You can find Google Apps, Negotiation and other examples of SaaS.

platform as a service

Stage as a Service provides the client with the necessary environment to develop software applications on the Internet. They can be used in the cloud and are accessible through a web application. There are several benefits for professionals who use PaaS to build their applications, including a reduction in infrastructure costs that would be incurred during application development. In addition, it integrates administrations such as business intelligence, databases, middleware, etc. on a single platform. Learn is an example of PaaS.

Infrastructure as a Service

Archiving, mounting, and management are provided to end users as part of the infrastructure-as-a-service model. They are available upon request. It provides the infrastructure that allows them to send and run their programs over a network. This reduces the upward pressure to maintain infrastructure for the benefit of customers. For example, Amazon Web Administration and Google Compute Engine are two examples of IaaS. It describes the many administrations provided by different cloud administrations, while the others describe who is responsible for managing these assets in different types of support, regardless of whether it is the service provider or the customer.

OBJECTIVES

1. Study attribute Cloud-based IaaS access control
2. Study of access control for cloud computing

3. Research Method

Access Control for Cloud Computing (AC3)

The proposed model is compatible with the role and race criteria already proposed in the previous work. Customers are organized in the model according to their actual situation. This way, clients are placed in a safe space that matches their job description and responsibilities. Each position within the model is assigned a set of the most relevant and necessary tasks to perform to prepare for that role. Each affected activity is assigned a security group to access data or resources and the exact permissions required to perform the activity. With a betting machine it is possible to control the random, dynamic and unpredictable behaviour of the customer; Request credits from consumers based on the methods they use to access the game.

In addition, a security label engine is used to apply security labels in semi-trusted or untrusted environments and through cycles. You can restrict access to data or resources by specifying data or resources with security names in the model. Any attempt to access the data must ensure that the properties of the activity take precedence over the security names of the data or resources. In our concept we apply security labels in specific situations based on the level of trust and security established within the ecosystem. As shown in Figure 4, the proposed security token would include the client's role, command, privileges, current region, specified time, and an irregular exception number, which will be irregular in nature.

AC3 includes the following basic elements:

- Customers (U) and refers to a group of customers.
- Roles(R)e is a collection of different roles.
- Tasks (T): This is a set of tasks.
- Sessions (S)e is a collection of individual sessions.
- Permissions (P) e is a collection of different permissions.
- The data (D) e is a collection of information.
- Customer Assignment (UA) e is a subset of convergence between U and R. It is a subset of convergence between U and R.
- A subset of convergence between R and T, role assignment (RA), and is a subset of convergence between R and T.
- PA e is a subset of convergence between P and T.
- The permission assignment (PA) e is a subset of convergence between P and T.
- e is a set of constraints used in the framework, such as B. the division of responsibilities and the delegation of powers.
- In the model, categories (Cla) are a set of security classifications used to logically organize activities.
- Confidentiality Tokens (SL) are a set of confidentiality names that are used to restrict access to data based on the sensitivity of the data.
- A collection of security labels (ST) is called a security label collection.
- However, there is one exception to this rule: the links between users and sessions (a user can only have one session at a time) and between activities and classifications (each activity has a classification), which are unique in the model.

$\forall u \in U \rightarrow s_i \in S.$. where ur cannot start outside of itself, but can still involve different roles, as in Fig. 3.

$\forall t \in T \rightarrow cla_i \in Cla.$

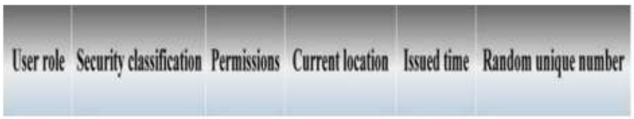
$\forall r \in R$ approved and activated an extreme number of clients at the same time.

$UA \subseteq U \times R.$ Many-to-many scheduling of client-to-role tasks, as in Fig. 4, where ((X and Y = U, R, or T) and (Z = UA, RA, or PA)).

$RA \subseteq R \times T.$ · A many-to-many mapping of roles in business

$PA \subseteq P \times T.$ A many-to-many mapping of activity authorization activities

There are some caveats to this paradigm, which are as follows: The least respectful policy is the concept of granting a subject the principal permissions p necessary to carry out its effort t, even if the subject has more permissions than are strictly necessary to complete the work in progress.



User role	Security classification	Permissions	Current location	Issued time	Random unique number

Figure 4

Model Security Levels

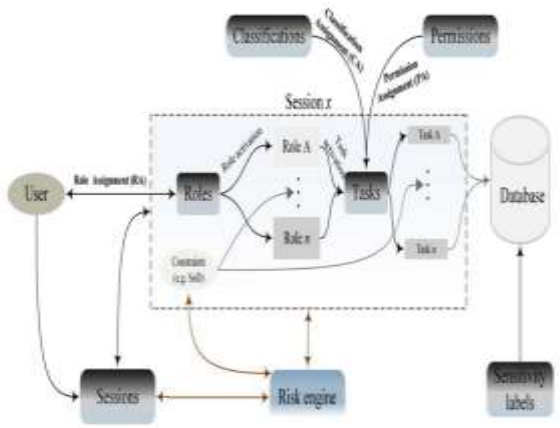
The framework can provide three different levels of security based on environmental security and process reliability:

1. A safe environment and reliable processes

It is not necessary to use security labels in this situation. Figure 4 illustrates how the rules and permissions are followed for each of the assigned tasks. Task attributes and privileges can be assigned to processes used by tasks to perform their tasks more efficiently. For example, an assignment could propagate its security group to any process that is currently involved in its activity.

2. Semi-safe environment or reliable process

As shown in Figure 4, security labels are used at this level and approval schedules are also established. Its application is limited to approval periods and cannot be transferred to procedures. When processes want to access data, they use some of the information provided by an execution that uses them, such as: B. General layout and configuration.



Coward. 5 Access Control for Cloud Computing (AC3) (Level 2 and 3)

3. Unsafe environment and unreliable processes

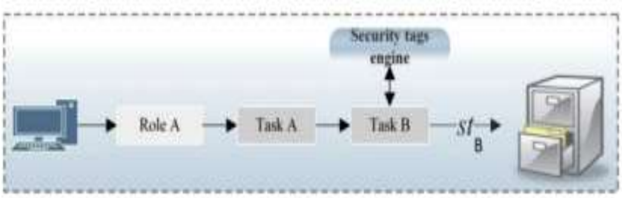
It uses a standard similar to that of Fig. 5, but each access to data or resources is assigned a security label to avoid the possibility of reuse of the security label or fraud by activities or processes. For example, to work, you need to create a specific security token and use it only once for each access point or process associated with it.

DATA ANALYSIS

Analyse

Although the proposed model is based on the T-RBAC model, whose configuration is advantageous due to its simplicity and adaptability, it must be developed and made available for distributed computing to support the assignment of dynamic and arbitrary standards. Customer Behaviour and Local and Global Access. Information has different levels of general knowledge, and this must be taken into account when designing an access control framework. The most widely used access control framework, MAC, includes information responsiveness as a component to grant or deny access based on information responsiveness. Even so, shipping is an expensive and time-consuming process. Apart from that, there is a big gap between web applications used in the application tiers and the lower tiers, because the framework components and loops in the lower tiers are treated as if they are fully trusted. At the end of the day, you shouldn't completely trust them. Because the connections between clients and assets are constantly changing, it is inevitably difficult to achieve the dynamics of distributed computing. It is very likely that cooperatives and specialized clients are in different security zones. Furthermore, because clients are not constrained by time or space constraints, the dynamic and arbitrary behaviours in which they behave are an important issue for designers of access control systems.

The proposed framework, as shown in Figure 6, can address each of the above problems with accompanying concepts:



Coward. 6 An activity accesses data through another activity.

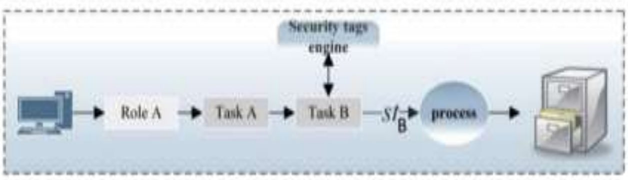
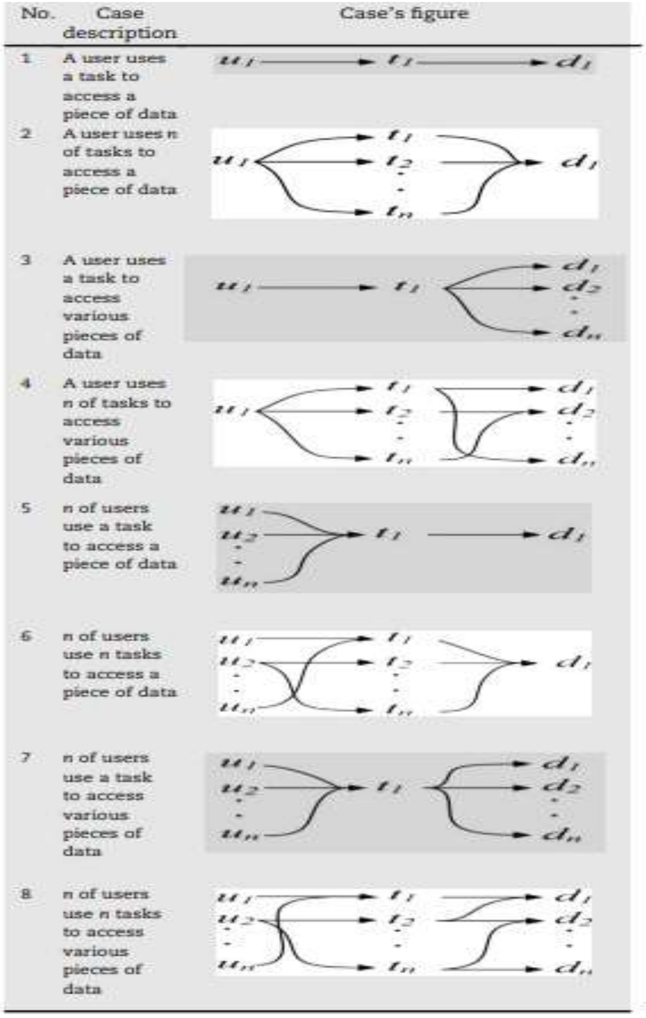


Fig. 7 A process received a security label from an activity used by another activity.

Table 1 Cases of data access in AC3.



No.	Case description	Case's figure
1	A user uses a task to access a piece of data	$u_1 \rightarrow t_1 \rightarrow d_1$
2	A user uses n of tasks to access a piece of data	$u_1 \rightarrow t_1, t_2, \ldots, t_n \rightarrow d_1$
3	A user uses a task to access various pieces of data	$u_1 \rightarrow t_1 \rightarrow d_1, d_2, \ldots, d_n$
4	A user uses n of tasks to access various pieces of data	$u_1 \rightarrow t_1, t_2, \ldots, t_n \rightarrow d_1, d_2, \ldots, d_n$
5	n of users use a task to access a piece of data	$u_1, u_2, \ldots, u_n \rightarrow t_1 \rightarrow d_1$
6	n of users use n tasks to access a piece of data	$u_1, u_2, \ldots, u_n \rightarrow t_1, t_2, \ldots, t_n \rightarrow d_1$
7	n of users use a task to access various pieces of data	$u_1, u_2, \ldots, u_n \rightarrow t_1 \rightarrow d_1, d_2, \ldots, d_n$
8	n of users use n tasks to access various pieces of data	$u_1, u_2, \ldots, u_n \rightarrow t_1, t_2, \ldots, t_n \rightarrow d_1, d_2, \ldots, d_n$

1. In groups and companies, paper is considered the standard means of regulating access to resources and other resources. The guilt of a subject is more important than the identity of the subject. Several general job skills combine to create jobs like accountant, secretary, and cook. Sandhu and his colleagues define a job as follows: "A job is a task skill or job title within association with semantics related to the authority and responsibility conferred on a person through the job."
2. Using jobs allows companies to enforce their requirements while maintaining full flexibility in terms of adding or removing activities for their customers based on their actual activities and calls.
3. Activity is another term used in AC3 to store permissions and access rights to work on a specific activity. Each client within a framework is assigned a task, and jobs are given assignments with associated permissions. In each job position within the framework there is a compilation of the most important and necessary missions to practice the specific profession. These permissions are limited to work related to your business, and the permissions granted change dynamically in response to the immediately adjacent assignment. Accreditation determines who is authorized to perform what tasks, with whom, and under what conditions.
4. The current or business cycle status determines whether model rights are enabled or disabled. These authorizations are assigned to specific executions, and the authorizations granted are dynamically modified based on the behaviour of individual clients.
5. The model is supported by a number of imperatives, including period mastery requirements, the rule of least honor, separation of duties into static and dynamic categories, and skill naming. However, the model contains no restrictions on the conditions that can be met.
6. Access to resources and data streams is controlled, among other things, through the use of security labels and classifications. Sensitivity names are used to label information based on its sensitivity and importance, which can be classified as highly confidential, secret, private, or not, among others. The model allows systems to use their own sensitivity flags. Any shop or bike used by a race requires a warrant to access resources as there should be no access to an asset without a characterization equal to or greater than the sensible names of the asset as access is not possible should give a good without a characterization that equals or exceeds the activity sensitivity designations.
7. AC3 includes a game engine to address a number of security issues. Stay in control and credit customers based on their past and current behaviour patterns. The board may be responsible for building relationships between customers, utilities, and other parties. Try your luck with awareness. Manage unpredictable and erratic behaviour patterns.
8. Security label engines are used to generate security labels for mappings, which can then be passed to procedures that need them. Security labels can be used in untrusted environments to restrict access to framework resources by higher priority applications or lower priority loops.

Case studies

Customers and specialized organizations can benefit from the higher level of security that AC3 offers across the various cloud tiers (including software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS)) both for buyers and specialized companies. Figure 1 show how it enables system administrators to efficiently and securely monitor access and authentication to their systems. We recognize that a cooperative cloud-based framework like Amazon or Google has been developed. The framework is designed using the three unique managements that distributed computing offers (SaaS, PaaS, and IaaS).

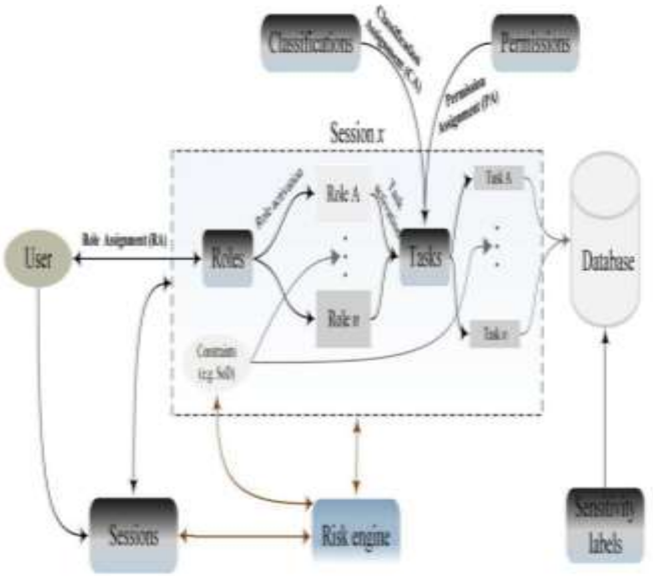


Fig. 8 Phase 1 in the AC3 model

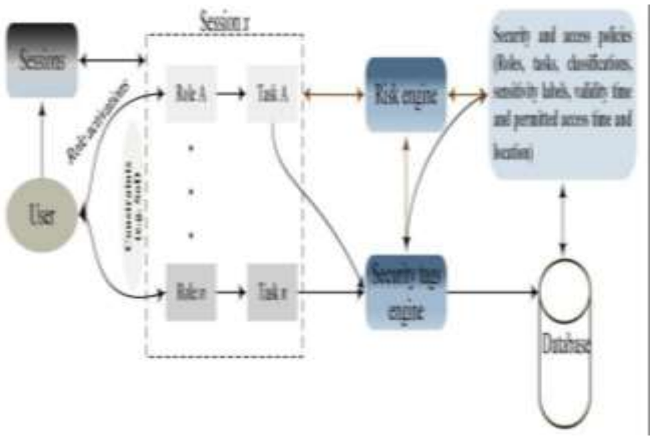

Sessions
Session x
Role activation
Role A
Task A
Role n
Task n
User
Risk engine
Security tags engine
Security and access policies (Roles, tasks, classifications, sensitivity labels, validity time and permitted access time and location)
Database

Fig. 9 the block diagram of AC3.

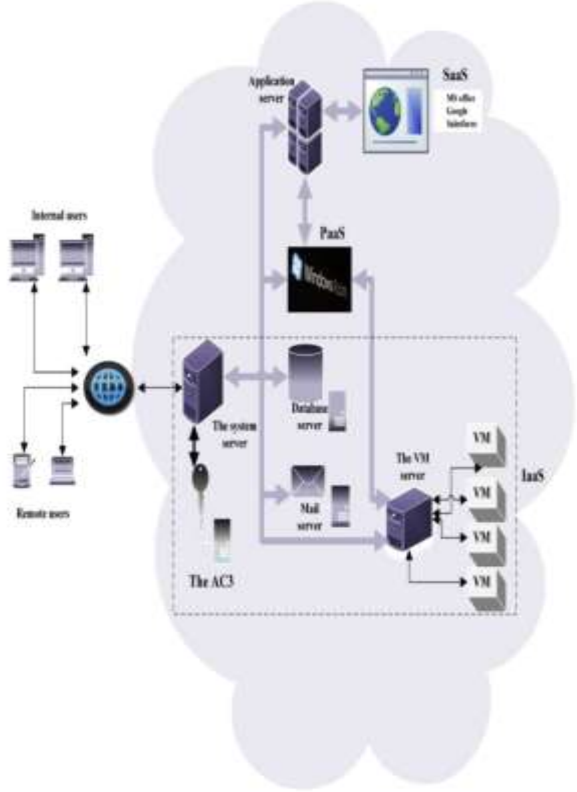


Fig. 10 AC3 in the cloud.

The concept has the potential to have a significant impact on IaaS access management. AC3 access methods describe the work and activities required for each task. Each project receives the necessary permits to carry out its mission and a security order to protect its assets. The security group assigned to a task is determined by the nature of the work and the resources trying to reach the site. In addition, it offers a fantastic component to control access at levels 2 and 3 through the use of security labels. All that is needed to generate these labels is a security label engine. They are also used when it is possible for someone else to gain access or a designated opportunity to complete an identical career. Each new client who logs in to AC3 will be assigned a job or a series of jobs based on their actual job and the client will be able to start working according to the established security and access measures. When a client is evaluating a job, the job will work with the client to create an actionable to-do list. For example, to access their mail through the mail server, a user uses a mapping to reach the mail server and that mapping must have a provision similar to or better than the security level of the mail server, eg:. B. a password generator. In this situation, the paths offered in SaaS can be used. As a result, access to the mail server can be granted based on assignments made or any verified client requesting it.

Discussion

When we start thinking about a cloud-based access control model for distributed computing, we are faced with two options: develop another access control model on the fly or evaluate one that is already available and can be used in the cloud. It has been established that offering a cryptic model is not best practice for the following reasons: First, there are access control models that contain a variety of components that should not be ignored. For co-op and specialist cloud customers to have confidence in the level of control they will be given, co-op and specialist cloud customers can make changes instead of using a different input control model that takes time to test and use in the cloud. for these cloud functions. Apart from that, the RBAC model is a well-known model of thinking that is used by various companies and organizations. Therefore, it should be used as the basis for all new proposed models. However, sending directly to the cloud is not possible. In addition, a small number of stores (associations or companies) use it in their internal organizations; Therefore, a shift to distributed computing may encourage the trend to move to cloud computing. To determine if the model is acceptable or not, we compare it with standard access control models. Table 2 contains the results of the review. In addition, some of the proposed access control methods for distributed IT systems were compared and compared. The evaluation depends on the security features offered by AC3 or alternative variants. Almost all access control systems proposed for distributed computing were examined before proposing our approach, which we then present. It is estimated that the vast majority of them have not been licensed or implemented in a true distributed computing environment. Another aspect of the planned plans is that some of the information will be retested and made available through the cloud (Wan et al., 2012; Yu et al., 2010). Others modify traditional access control models and their extensions for use in distributed computing environments (Andal Jayaprakash and Hadi Günes, 2011; Tsai and Shao, 2011).

1. Principles

The model can adhere to a number of important standards, such as the rule of least honor, the naming of capabilities, the distribution of obligations (static and dynamic), and ephemeral imperatives (space and time). To our knowledge, only one plan has come close to these standards (Andal Jayaprakash and Hadi Günes, 2011).

2. Support for passive and active workflows

AC3 supports separate and dynamic workflows, where orders are uninvolved workflows and commissions are dynamic workflows. There is only one methodology that offers both work processes and this objective model of health care systems (Andal Jayaprakash and Hadi Günes, 2011).

3. Review

The model provides a unique and innovative access control methodology that uses data review and logging to monitor and credit customers based on their past and current behaviour. This is complemented by the game engine in the proposed model.

4. Policy management (adds, delete, modify, import, export)

There is great interest in legal means that can be used to establish relationships between customers, utilities, and other external parties. AC3's betting engine offers a new way of looking at the leaders. There was little evidence that any of the models studied (Andal Jayaprakash and Hadi Günes, 2011; Tsai and Shao, 2011; Wan et al., 2012; Yu et al., 2010) manipulated board strategy or referred to all modes.

5. Manage heterogeneity

Heterogeneity may appear in access control systems using many types of tools, locations, and methods. The strategy table of the model depends on the betting machine, which must be able to adapt to the heterogeneity caused by the security devices. Consequently, it relies on this model. Regarding the difficulties of heterogeneity, the cosmological rule of Sun et al. (2012) and Tsai and Shao (2011) but require extensive metaphysical calculations of change to examine the comparability of multiple cosmologies. These plans require the creation of a new mapping of the back-end registries to facilitate the implementation of O-RBAC.

As a basis for property representation, the authors of Iqbal and Noll (2012) have advocated the Uniform Resource Identifier (URI) as a standard. They have integrated it into the Resource Description Framework (RDF) to manage the risks of heterogeneity that arise from using different functions within the framework. In any case, the URI and RDF means of transmitting information must be thoroughly tested, as they have a significant impact on the credits used, and the language of the Semantic Web rules must comply with current Semantic Web standards and guidelines.

4. Conclusion

In this research, we propose a new access control mechanism for distributed computing that is efficient and effective. We believe that the proposed model can meet the input control requirements in distributed computing and should be implemented. It works in conjunction with activity and career requirements to make the award process incredibly quick and easy. It also uses the transition (big picture) and naming requirements and requirements. Customers are assigned security zones that are appropriate for their legitimate use and location within our facilities. Each job within the model is assigned important tasks that allow it to perform its functions in the model. A continuous stream of information for the model can be accessed by marking the information with security markers that represent the sensitivity of the information. Each run comes with a security order detailing how to access the information or resources and the exact permissions needed to complete the mission. Consequently, any execution or loop that attempts to access the information must be in an order that overrides the information security flags of the selected resource or equivalent. A game engine is used to control the behaviour of clients that have variable and arbitrary behaviour patterns. Give credit to customers based on their behaviour in certain situations. A security label engine can also be used to deliver security labels in semi or unreliable situations and loops, depending on the situation. Security labels are used in specific circumstances and the image is communicated to reflect the level of trust and security in the area. The proposed security tag contains only the information necessary to secure access while being lightweight. We will develop a validation tool capable of handling large amounts of time and complex spatial arrangements in connection with this project. We will also start up the betting machine and its components, which are responsible for controlling the dynamic modes of action. After completing the validation component and the random engine, the model is deployed and evaluated.

References

[1] Smriti Bhatt, Farhan Patwa and Ravi Sandhu (2017) “An Attribute-Based Access Control Extension for OpenStack and its Enforcement Utilizing the Policy Machine” 2016 IEEE 2nd International Conference on Collaboration and Internet Computing

[2] Shadha Mohamed Sulaiyam AL Amri (2018) “IaaS-Cloud Security Enhancement: An Intelligent Attribute-based Access Control framework”

[3] Younis A. Younis*, Kashif Kifayat, Madjid Merabti (2017) “An access control model for cloud computing” School of Computing and Mathematical Sciences, Liverpool John Moores University

[4] Sushmita Ruj (2017)"Attribute-Based Access Control in Clouds: A Survey" R.C. Bose Center for Cryptology and Security, Indian Statistical Institute, Kolkata

[5] J. Bringer, B. Gallego, G. Karame, M. Kohler, P. Louridas, M. Onen, ¨ H. Ritzdorf, A. Sorniotti, D. Vallejo, TREDISEC: Trust-Aware REliable and Distributed Information SEcurity in the Cloud, in: EDemocracy Citizen Rights in the World of the New Computing Paradigms, Springer International Publishing, 2015, pp. 193–197.

[6] D. Nguyen, Provenance-based access control models, Ph.D. thesis, The University of Texas at San Antonio (2014).

[7] B. Tang, Multi-tenant access control for cloud services, Phd, The University of Texas at San Antonio (2014).

[8] Y. A. Younis, K. Kifayat, M. Merabti, An access control model for cloud computing, Journal of Information Security and Applications 19 (1) (2014) 45–60.

[9] B. Anggorojati, N. R. Prasad, R. Prasad, Secure capability-based access control in the M2M local cloud platform, in 2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronics Systems (VITAE), IEEE, 2014, pp. 1–5.

[10] B. Li, J. Li, L. Liu, C. Zhou, Toward a flexible and fine-grained access control framework for infrastructure as a service cloud, Security and Communication Networks (2015)

[11] F. Li, Context-Aware Attribute-Based Techniques for Data Security and Access Control in Mobile Cloud Environment (Apr 2015).

[12] Y. Zhang, F. Patwa, R. Sandhu, B. Tang, Hierarchical Secure Information and Resource Sharing in OpenStack Community Cloud, in 2015 IEEE International Conference on Information Reuse and Integration, IEEE, 2015, pp. 419–426.

[13] K. Bijon, R. Krishnan, R. Sandhu, Virtual Resource Orchestration Constraints in Cloud Infrastructure as a Service, in: Proceedings of the 5th ACM Conference on Data and Application Security and Privacy - CODASPY ’15, ACM Press, New York, New York, USA, 2015, pp. 183–194.

[14] C. Ngo, Y. Demchenko, C. de Laat, Multi-tenant attribute-based access control for cloud infrastructure services, Journal of Information Security and Applications 27 (2015) 65–84.