Journal of Advances and Scholarly Researches in Allied Education
Vol. XV, Issue No. 1, April-2018, ISSN 2230-7540

A Security Challenges in Multi-Tenant Cloud Computing

Dipti Prava Sahu[1]* Dr. B. L. Raina[2]

[1] Research Scholar, Computer Science & Engineering, Glocal University, Uttar Pradesh

[2] Professor, Computer Science & Engineering, Glocal University, Uttar Pradesh

Abstract – Cloud computing facilitates multi-tenancy for optimal resource utilization through leveraging hardware and software resources through multiple clients. Multi-tenant programs operating on a cloud infrastructure are delivered over the network to clients as Software-as - a-Service (SaaS). Multi-tenancy presents additional challenges regarding its benefits, such as partitioning, extensibility and customizability during application development. In this article, we discussed the various types of multi-tenancy, multi-tenancy implementations, advantages, and drawbacks within different cloud-based service models. Many users are, however, reluctant to subscribe to cloud computing services due to security concerns.

Keywords – Multi-Tenancy, Multi-Tenancy Security Challenges, Multi-Tenancy Economics

----------X----------

INTRODUCTION

Cloud computing has a new architecture known as multi-tenancy. It explicitly describes Bezemer's "Multi-tenant framework let tenants (customers) share the same hardware resources, including mutual apps and database instances that enable users to fit their needs as they do in a dedicated setting." Multi-Tenancy has the ability to share hardware services and offer a high degree of device configuration[1]. It has an infrastructure in which several tenants use a single database and an application instance in a single system, and a security breach can result in other tenants being exposed to data. The advantages of cloud computing entail changes to the model; one of the most challenging aspects of this is security. Information Security provides security for data and information systems against unauthorized entry, usage, release, disturbance, alteration, review, monitoring or degradation. Based on the Cloud Security Alliance (CSA) report, there are seven major threats that companies will encounter when implementing Cloud Computing. These include Violence and Nefarious Use of Cloud Computing, Insecure Application Programming Interfaces (APIs), Malicious Insiders, Remote Infrastructure Vulnerabilities, Information Loss / Leakage, Server, Network and Traffic Hijacking and Known Risk Profile. Multi-Tenancy is regarded as one of the main security and privacy aspects of cloud computing. Multi-Tenancy is a crucial part of cloud computing and a significant dimension of cloud security issues that require a vertical solution from the Software-as - a-Service (SaaS) to the Infrastructure-as-Service (IaaS) system. The multi-purpose function of cloud computing allows multiple people to concurrently access the same hardware and software services that are available in remote locations but with personalized requirements using the virtualization principle. As Multi-Tenancy has been described as a security issue in cloud computing, a detailed analysis of Multi-Tenancy is expected to deal effectively with it.

MULTI-TENANCY

The Main requirement of multitenancy is that the software provider gets many requests from customers with the customized needs. If a software product is implemented according to each customer needs separately and delivered, then the implementation takes more time to complete. The software cannot be maintained easily if there are different implementations of the product. The provider needs to spend more money to satisfy different customers. Here multi-tenancy comes into existence to provide solution for all the problems faced by provider to satisfy different customer with different needs. Multi-Tenancy allows single software to be served between the multiple customers by using customized settings option. The needs of each customer are stored in custom settings. The software provider serves the same product by implementing it seeing the customized requirements of each customer and makes it available only to the specific customer respectively. The tenants who share the software product cannot see each other's implementation of product. There is no contact between each customer's sharing the same software.

The software provider must be in contact with multiple customers to satisfy them[3].

Multi-Tenancy means sharing the application software between multiple users who have different needs. Allocating solitary instance of an application software i.e., cloud to multiple users is called as multitenancy. Each user is called as tenant. The users who need similar type of resources are allocated a solitary illustration of cloud, so that the cost is shared between the users to make the access of instance of cloud computing cost effective. Multi-Tenancy allows users to easily access, maintain, configure and manipulate the data stored in single database running on the same operating system[4]. The data storage mechanism remains same for all users who share the similar hardware and software resources. In multitenant architecture, user cannot distribute or observe every other's data, here the security and privacy is provided. To perform any type of services like IaaS, SaaS and PaaS in public cloud and private clouds the key technique is Multi-tenancy. If the people discuss about the clouds they many speak about the IaaS Services. Both cloud architectures like private and public clouds go beyond the special features like Virtualization and the concept of IT-as-a-Service through payments or billing back in the event of private clouds based on metered usage. An IaaS service has an advanced features such as Service Level Agreements (SLAs), Identity and Access Management for Security Access)(IDAM), fault tolerance, disaster recovery, dynamic resource allocation and many other important properties. By Injecting all these key services at the level of infrastructure, the clouds become multitenant to a degree. In the case of IaaS multi-tenancy go beyond the layer to merge the PaaS layer and at the end SaaS layer or application layer. IaaS layer contains Servers, Storages and networking components, PaaS layer Consists of Platform for Applications like Java Virtual Machines like Java Compilers, Application Servers and SaaS Layer Consists of applications like business logic, work flow, data bases and user interfaces.

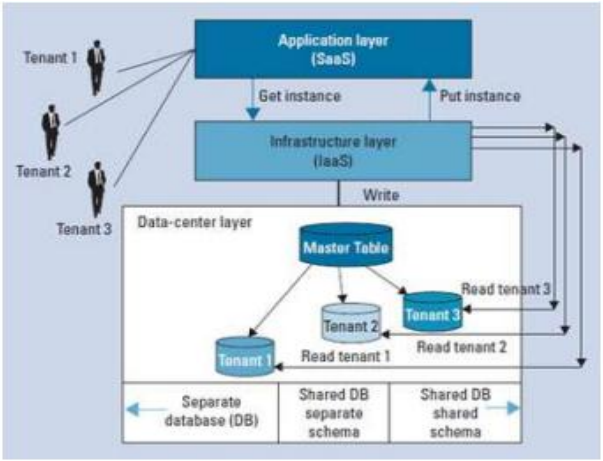

Fig1: Architecture of Multitenancy

Tenant can like the full stream of applications that are widely used from the network application cloud services to the user interface, depending on the degree of multi-tenancy provided by the provider. Cloud computing multi-tenancy is used for most, if not all, of the SaaS systems, since computational services are flexible and the distribution of these resources is determined by real usage.

There are different types of SaaS services that the clients can access by using internet, from low internet bases applications to a very big software applications that contains a very high security requirements depends on the type of information stored on the software vendors infrastructure outside the corporate network. There are basically two types of Multitenancy Techniques like:

Virtual Multi-Tenancy: This Computing and Space Capacity is shared by multiple users. Several tenants are supported by virtual machines that run simultaneously on top of the same computing and space tools.

Organic Multi-Tenancy: Throughout organic multi-tenancy, each component, i.e. hardware and software resources throughout the network architecture, is owned by several tenants. Internet multi-tenancy principles are introduced at three specific rates of consumer integration.

They are:

- Data centre layer
- Infrastructure layer
- Application layer

The infrastructure layer and application layer consumer integration levels are latest additions to the cloud computing model. This integration is used to diminish the cost and developing highly scalable SaaS applications, which they do by compromising on security and customer segregation requirements.

Data centre layer: This configuration provides the highest level of security requirements if implemented correctly, with firewalls and access controls to meet business requirements as well as defined security access to the physical location of the infrastructure providing the SaaS. Mostly data centre layer multitenancy acts as a service provider that that rents cages to companies that host their hardware, network, and software in the same building.

Infrastructure layer: In infrastructure layer multi-tenancy the software stacks are provided. Each customer or tenant is provided with a dedicated software stack. T his configuration saves costs compared to data centre-layer multi-tenancy, because stacks are deployed based on actual customer accounts. The high availability of hardware and software resources can be seen in this layer. In

this case, you can grow hardware requirements based on actual service use.

Application layer: Application-layer multi-tenancy requires architectural implementations at both the software layer and the infrastructure layer. Modifications are required for the existing software architecture, including multi-tenant patterns in the application layer. For example, multi-tenant applications require application methods and database tables to access and store data from different user accounts, which compromises on security. If done accurately, however, the benefit is cost savings.

Software as a Service provides a software model to deliver software based applications to provide remote access to the customers. A key feature of cloud multitenancy is the provision of SaaS services to multiple tenants at the same time as a single application instance at the top of the shared infrastructure.

Nowadays, SaaS applications are being built with centralization through a single instance of multi-tenant architecture to provide an advanced rich experience compared to on-site models. The benefit of multi-tenancy is that operating costs are minimized by splitting equipment, sharing computing resources among various tenants, and simplifying maintenance and management efforts. All of these advantages of a multi-tenancy impact of rising implementation costs in order to provide maximum benefits for small and medium-sized organizations. Multi-Tenancy System Standards for Cloud Services Providers include tenant data insulation, tenant environment insulation, tenant execution insulation, Tenant-aware protection, surveillance, maintenance, reporting and self-service administration, separation of tenant customizations and business logic extensions, tenant-aware version control, Tenant-aware error tracking and recovery. The degree of flexibility of the framework is specified as the amount of base application or the SaaS layer is built to be shared number tenants. The maximum degree of flexibility enables the database schema to be exchanged and facilitates configuration of business logic, process and user experience levels. Personal clouds are accessible at the lowest multi-tenancy rates and are more suitable to unique large business clients.

MULTI-TENANCY SECURITY CHALLENGES

What is unique about Multi-Tenancy in Cloud Computing is that both the attacker and the victim are sharing the same server (i.e. physical machine (PM)). Such a setup cannot be mitigated by traditional security techniques and measures, simply because it is not designed to penetrate inside servers and their monitoring techniques are limited to the network layer [5-6].

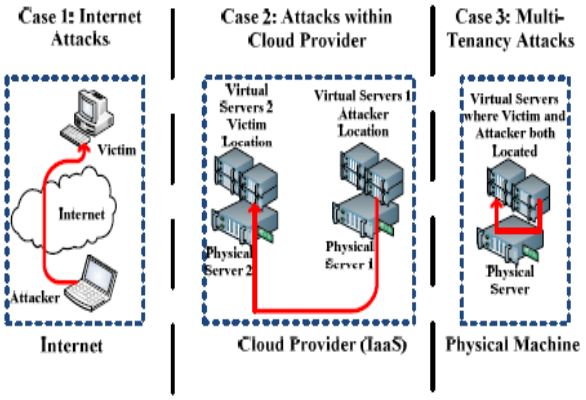

Figure 2: Difference between Multi-Tenancy and Traditional Cases.

To illustrate, Fig. 2 shows the different cases of attacker and victim locations and the networking between them. In case one, the attacker and the victim both are regular Internet users; in order to defend against such attacks, traditional network security techniques and devices are efficient. In case two, both attacker and victim are customers in the same Cloud provider but each one of them is located on a separate server. This kind of setup is due to the utilization of the virtualization layer in the Cloud Computing Model; to secure such a setup, virtual network security devices and techniques must be implemented by Cloud providers [6].

Case three describes the problem that we intend to address in future work, where both the attacker and the victim are customers in the same Cloud and are sharing the same server. Such a situation is due to Multi-Tenancy; securing such a setup is not an easy task as network communication between the attacker's VM and the victim's VM is limited within the physical machine (PM). Therefore, traffic will not leave the physical machine, which is harder to be mitigated by virtual network security defenses as opposed to case two.

In order to secure such vulnerability, we must first answer the following question: how is Multi-Tenancy exploited? An answer can be found in [7], where an attack is generated over the Amazon EC2 Cloud to investigate data leakage. In order to carry out the attack, network probing is performed; following this, a brute force attack is generated to take advantage of the Multi-Tenancy effect by allocating the attacker's VM beside the victim's VM. The results show that Figure 3: Proposed System Model. by spending just a few dollars, an attacker has a 40% chance to allocate his VM beside the victim's VM. After achieving Multi-Tenancy, a side channel attack any attack takes advantage of the system characteristics is generated to extract the data of the victims. Obviously, any tenant can attack its neighbor because the type of attack that could be utilized, such as side channels, cannot be detected by the hypervisor or even the

operating system. There is no way, however, to remove the Multi-Tenancy impact in order to retain its advantages, yet the effect could be reduced and what this paper is attempting to demonstrate. Multi-Tenancy cannot be avoided, yet a clever resource allocation strategy can minimize the risk of multi-Tenancy; in other terms, a resource allocation methodology would improve the complexity of obtaining multi-Tenancy for users while being easily managed by cloud providers.

What is interesting of Multi-Tenancy is that in order to achieve it for targeted victims, the attacker needs to invest an effort, time and cost. So, by making Multi-Tenancy difficult to be achieved by customers, we are restricting the number of potential attackers.

MULTI-TENANCY ECONONMICS

- Cost savings:- Multi-Tenancy provides cost savings over and above the simple economies of scale that can be obtained from the convergence of IT services into a single operation[8]. Systems typically incur a certain amount of overhead memory and processing that can be significant when compounded by many customers, particularly when customers are low. Multi-Tenancy wipes out this cost by extending it to numerous customers. More cost savings can emerge out of permitting costs for the basic programming, (for example, working systems and database the executives systems). Put roughly, in the event that you can work everything on a solitary programming case, you just need to acquire one programming permit. Cost savings can be eclipsed by the trouble of scaling a solitary occurrence as request increments expanding the exhibition of a case on a solitary server must be accomplished by acquiring quicker equipment, for example, quick CPUs, more memory, and quicker disk systems, and these costs typically increment quicker than if the heap was part between multiple servers with generally a similar total capacity[9]. Furthermore, the improvement of multi-tenant systems is progressively intricate and security testing is increasingly thorough because of the blend of multiple client data.

- Data aggregation/data mining:- One of the most convincing purposes behind sellers/ISVs to utilize multi-tenancy is the natural points of interest of data aggregation. Rather than social event data from multiple data sources with conceivably extraordinary database outlines, all data for all customers is contained in a typical database mapping. In this way, running inquiries through customers, mining data, and pattern looking is much simpler. This is probably going to be overhyped as one of the key multi-tenancy necessities is the need to prevent Service Provider from getting to client (tenant) data. Furthermore, it is normal to distinguish the operating database from the mining database (usually due to different workload characteristics), thereby undermining the claim even more.

- Complexity:- Due to the additional complexities of configuration and the need to retain per-tenant metadata, multi-tenant applications require greater development effort. Considerations such as vector-based data sequence encrypting effective algorithm systems and virtualized control interfaces must be taken into account. [10].

- Release management:- Multi-Tenancy simplifies the process of handling updates. In the conventional release management process, bundles comprising application and database improvements are delivered to client desktops and/or storage machines; in the case of a single instance, For each device, that would be one host PC. These things will be installed on any given computer at that stage. The software usually just has to be installed on a single server for a multi-tenant environment. This greatly increases the process of discharge power, and the size will never again be subject to consumer quantities.

At a similar period, multi-tenancy expands the dangers and effects engaged with the usage of the new discharge rendition. Since there is a solitary programming occasion serving multiple tenants, an update to this case may cause personal time for all tenants, regardless of whether the update is mentioned and is valuable for just one tenant. Likewise, certain glitches and issues that emerged from the execution of the most recent variant might be communicated in the modified understanding of the item by specific tenants. As per potential personal time, the time of discharge may be abbreviated dependent on the time pattern of utilization by more than one tenant.

BENEFITS OF MULTI-TENANCY

- Lower cost of ownership:- Since all customers have access to their apps from the same technology platform, it is much easier to access regular and periodic notifications. You no longer need to compensate for data customizations or add new features.

- Worry free capacity:- Multi-Tenancy allows businesses of all types to operate in the same network and data center.

- API Integration scalability:- Web API functionality is possible in single cases, but in the multi-tenancy world, unique configuration requirements will now be included in our product roadmap, so as they

become usable, they will be pushed out to all customers.

- Access to the latest releases:- Previously, anytime we decided to roll out a new version, it was a long process because we had to code the improvement individually for each customer instance and make sure it was consistent with their customizations, conduct QA, and then bring the change into development. It was a time-consuming job for our support team with more than 100 clients. Now with our multi-tenant environment, because every consumer instance has the same basic code, the roll-out of new releases will be very smooth and will provide faster access to innovative features to reduce IT and connectivity costs.

- Configurable to your own needs:- It helps our clients to meet their requirements and contact preferences in order to manage both IT and communication costs[11].

DISADVANTAGES OF MULTI-TENANCY:

- A multi-tenant system has less ability to create low-level requirements than a single-tenant device. This may not be a concern for you, but if your design requires a lot of flexibility for each new tenant, it may not be the best solution.

- The multi-tenant system is more complicated than the comparable single-tenant device, the design of which can stay largely unchanged. You do not need any code in a single-holder application to detect that tenant a web request is designed to prevent your clients from contamination of data among tenants. File types are easier because logs are segregated by a different program instance for each device.

- Since a multi-tenant program is backed up by a single database operating on a single server, there are less locations that are prone to failure, but those failure points may prove to be much more catastrophic. Both residents are feeling lack of operation. When a database for a multi-tenant app is unavailable, unlike when a single-tenant application breaks. It's taking down a single occupant. Other instances remain unaffected.

CONCLUSION

Cloud computing facilitates multi-tenancy for optimal resource utilization through leveraging hardware and software resources through multiple clients. At present, both systems are introduced utilizing multi-tenancy methods and these technologies are used in most business applications. Multi-Tenancy is consider as one of the main security and privacy aspects of cloud computing. In this article, we addressed the different types of multitenancy, implementations, advantages and disadvantages of multitenancy within specific cloud-based service models such as SaaS, PaaS and IaaS.

REFERENCES

[1]. S. Subashini, and V. Kavitha (2011). “A Survey on security issues in service delivery models of cloud computing,” Journal of Network and Computer Applications.

[2]. Mladen A. Vouk (2008). “Cloud computing – issues, research and implementations,” Journal of Computing and Information Technology.

[3]. M.Saraswathi, and T.Bhuvaneswari (2013). “Multitenancy in Cloud Software as a Service Application,” International Journal of Advanced Research in Computer Science and Software Engineering, Vol 3, Issue 11, 2013 pp.

[4]. Hussain Al-Jahdali, Abdulaziz Albatli, Peter Garraghan, Paul Townend, Lydia Lau, and Jie Xu (2014). “Multi-tenancy in cloud computing,” In proceedings of the 8th IEEE International symposium on service-oriented system engineering, pp. 159-162.

[5]. Ruoyu Wu, Gail-Joon Ahn, Hongxin Hu, and Mukesh Singhal (2010). “Information flow control in cloud computing,” (9-12 Oct. 2010).

[6]. Augusto Ciuffoletti (2010). “Monitoring a virtual network infrastructure,” (October 2010).

[7]. Prasad Saripalli, and Ben Walters (2010). “QUIRC: a quantitative impactand risk assessment framework for cloud security,” IEEE 2rd International Conference on Cloud Computing.

[8]. "Web-to-Print Technology, Recuce Costs, Increase Sales, Integration with Salesforce and Metrix". Presscentric.com. Retrieved 20 January 2014.

[9]. "Building SaaS App with Codeigniter MVC". Computer Technology News Blog. Retrieved 5 May 2016.

[10]. Aulbach, S. (2011). "Extensibility and data sharing in evolving multi-tenant databases".

2011 IEEE 27th International Conference on Data Engineering.

[11]. Ahmed E. Youssef (2012). Exploding Cloud Computing Services and Applications Journal of Emerging Trends in Computing and Information Sciences VOL. 3, NO. 6, ISSN 2079-8407

Corresponding Author

Dipti Prava Sahu*

Research Scholar, Computer Science & Engineering, Glocal University, Uttar Pradesh

diptiprava29@gmail.com