Secure Multi-tenant Application in Software as a Service

M. D. Samrajesh1,∗ and N. P. Gopalan2

1Research and Development Centre, Bharathiar University, Coimbatore, Tamil Nadu 641 046, India.
2Professor, National Institute of Technology, Tiruchirappalli, Tamil Nadu 620 015, India.
e-mail: samrajesh.md@gmail.com; npgopalan@nitt.edu

Abstract. A multi-tenant application in Software as a Service (SaaS) is accessed by multiple organizations called tenants who have several users attached to them. Multi-tendency enhances resource utilization by effectively sharing resources and reducing cost, increasing productivity and online collaboration. However, providing a secure multi-tenant access to various tenants is still a challenge. Traditional security methods of applications are not effective in a multi-tenant application model due to multiple tenants’ access on a single application instance at runtime. This paper focuses on providing an integrated security model for SaaS application. The tenant specific security specifications are defined using a pluggable component that can be easily integrated to the SaaS application. It works effectively in enforcing security controls, and monitoring SaaS application security. The pluggable security definitions can be easily interleaved with the application at runtime without any interference from the provider. Thereby the multitenant SaaS provider focuses more on the application functionality rather on application security. The proposed Secure Multi-tenant Application (SMA) model provides security isolation among tenants at various levels during runtime, reduces security risks and protects sensitive tenant data. Our evaluation and discussions show the effectiveness of the proposed model in securing SaaS multitenant application.

Keywords: Application security, Cloud computing, Encryption, Multi-tenancy, SaaS.

1. Introduction

Cloud Computing based Software as a Service (SaaS) multi-tenant application allow SaaS providers offer a single application that runs on top of shared hardware resources as shown in figure 1 to several users from multiple organizations called tenants [9]. In a SaaS multitenant environment operational overheads and maintenance is considerably low due to the sharing of hardware and software resources among multiple tenants. Moreover, upgrades need to be applied only to the shared application instance giving all tenants access the most recent version of the application. Maximum resource utilization and efficiency is achieved by application-level multi-tenancy where the infrastructure, data storage, operating system, and the application are shared among multiple tenants [2].

SaaS offers tremendous benefits, however concerns of security is one of the primary obstacle that makes organizations avoid its adoption [1]. Traditional model of application security that is fixed and hard coded by service providers are not well suited for multitenant applications that run on single shared instance. Moreover, earlier models of security are system based rather than tenant based. Additional, various commercial cloud platforms have limited security solutions [17].

The proposed secure multi-tenant application model is an integrated security model that defines a flexible pluggable component which is tenant specific and that can be easily integrated to the SaaS application. It works effectively in enforcing security controls and monitoring SaaS application security. The pluggable component’s tenant security specifications can be easily interleaved with the application at runtime without any interference from the provider. Thereby the multitenant SaaS provider focuses more on the application functionality rather on application security.

The proposed SMA provides effective tenant isolation at the database level, application level, and during runtime. We evaluated our proposed method by Monte Carlo method using Arena [15] and our customized tool [12]. Our evaluation and discussions based on parameters in [6] shows the effectiveness of the proposed model in securing SaaS multitenant application.

∗Corresponding author

© Elsevier Publications 2013.
The paper is structured as follows: Motivation & problem definition is presented in Section 2; Section 3 discusses related work, background information on multitenant architecture is presented in Section 4. Our proposed secure multi-tenant application model is presented in Section 5, the evaluation and discussions on the proposed model are presented in Section 6 and finally, the conclusion and future work are in Section 7.

2. Motivation and Problem Definition

Data security is another important area of research in cloud computing [3]. In SaaS model, the tenants depend on the SaaS application providers for their application security. The provider has to ensure that no tenant can view each other’s data; it is often complex for the provider to ensure the presence of right security measures [9]. Moreover the provider might replicate tenant’s data to multiple locations for maintaining high availability, the security measures should not be compromised during the process.

Clients of cloud computing services currently have no means of verifying the confidentiality and integrity of their data and computation [5]. SaaS offers various features that can significantly reduce IT costs by placing data and computation onto cloud platform. However, many organizations are reluctant move to cloud, primarily due to the security concerns [5]. The dynamic nature of multi-tenant application model further intensifies the application complexity and makes security more challenging [14].

Auditability can be generally achieved using remote verification methods. Remote verification requires a Trusted Platform Module (TPM). However, in a cloud where resources are virtualized, and virtual machines (VMs) dynamically migrate from one location to another, the above verification is not sufficient [5]. Enterprises are still not confident with the SaaS model due to lack of visibility about the means their data is stored and secured. Moreover, there is a huge anxiety over the lack of control and knowledge about tenant’s data in SaaS [8].

3. Related Work

Current multi-tenant cloud security solutions take an inert stand to permissions as systems are functioning around a simple concept relating to the access rights of the user [18], for example the user is granted entry or denied entry based on a key. All subsequent actions are carried out based on the outcome of the initial attempt to access. Not much research has been done in the area of security of multi-tenant SaaS applications [16]. Different commercial cloud platforms have limited security solutions [17].

An approach that transforms existing web applications into multi-tenant SaaS applications is proposed in [19,20]. It focused on isolation problem by investigating application to identify the potential isolation points that should be handled by the application developers. A framework that supports a set of common services that provide security isolation, performance isolation is presented in [21]. The security isolation pattern considers the case of different security requirements while still using a predefined, built-in, security controls. Moreover security policies depend on...
the tenant’s administration personnel configuration that is done manually which maps users and their roles in the application.

A model-driven platform that creates SaaS applications as a set of services is presented in [22]. The approach focuses on enabling cloud customers to create their system instances and specify their security requirements. However, tenants’ instances need to be deployed on separate VMs. Moreover, it lacks facility to update or reconfigure the defined security. A hierarchical based access control model is presented in [23]. The model increases the number of levels to the access control policy hierarchy; it includes new roles such as service providers’ administrators and tenants’ administrators. Service provider administrators delegate the authorization to the tenant’s administrators to grant access rights to their predefined resources.

Secure load distribution SaaS architecture is presented in [13]. The architecture is based on a set of services that provide routing, logging, and security. Their proposed security service delivers predefined authentication and authorization mechanisms. However, no isolation is provided between the authentication and authorization data of different tenants. Our proposed method considers tenant specific custom security implementations which are integrated with the SaaS application provider’s security model thereby providing an integrated pluggable security model as shown in figure 2.

4. Multitenant Architecture

Multi-tenancy is an application architecture where multiple clients or organizations (tenants) are served by a single application as illustrated in figure 1. A Multi-Tenant Application (MTA) is highly scalable and shared among multiple tenants hence it reduces cost, reduces software development time and brings down maintenance cost [7]. All tenants accessing the multi-tenant application run on the same source code, however certain tenant specific requirements are generated at run-time. Having a shared code benefits all tenants as updates can be applied centrally, moreover this enhances productivity and online collaboration.

To accomplish large scale scalability, a multi-tenant application is designed to be stateless. Tenant-specific data are kept either at the tenant side or at the database. Moreover, unlike traditionally systems, tenant-specific implementations of application logic are created during run-time for each tenant. The MTA queries the database for necessary tenant-specific business logic and creates a customized implementation at run-time. Multitenant application is multifarious in nature and accomplishes individual tenant’s requirements [10]. Tenants are offered with a well customized application that makes them experience as if the application is exclusively designed for them [7].

5. Proposed Secure Multi-tenant Application (SMA) Model

5.1 Assumptions

- One single multitenant application is used simultaneously by multiple tenants.
- Each tenant has ‘n’ number of users attached to them.
- The multitenant application consists of components that are modules of the application which is designed based on the business processes.
- Tenant’s users by default access the core application components, all add-on components including the security component are accessed based on tenant specific needs.
5.2 Secure multi-tenant application (SMA)

The proposed Secure Multi-tenant Application (SMA) model as shown in figure 4 consists of a Tenant Security Specification (TSS) which consists of custom security specification, securely stored either at the tenant side or the provider side; it is used to generate the pluggable secure component that secures the application. As shown in figure 3 the framework consists of a tenant pluggable security component that is integrated with the application of the tenant at run-time.

The Tenant pluggable components consists of the following core entities

- Digital Signature with IdM for Authentication and Authorization. Relevant tenant information is securely replicated to the SaaS providers. Authorization using XACML (eXtensible Access Control Markup Language) delegation method is used to provide decentralized administration of access policy.
- The custom Tenant Security Specification (TSS) is configured by the tenants with their location MAC address or using directory lookup LDAP.
- Network security is provided with the pluggable custom security options configured by the tenant’s predefined Application Programming Interface (APIs) signatures using Secure Socket Layer (SSL) and the Transport Layer Security (TLS) with an encrypted connection between a customer’s data and application.
- Data Security is implemented for all tenants’ data by encryption using the key of tenant and the provider; hence it is neither accessible nor tampered by unauthorized person.
• Data integrity is provided by data segregation and confidentiality by allowing fine-grained authorization controlled access to application and data.

• Accountability is effectively implemented by comprehensive logging. All actions on tenant specific data and application component access are monitored, correlated and alerts are sent to tenants and the provider.

The pluggable custom TSS is configured by the tenant with their location Media Access control (MAC) address or by using directory lookup Lightweight Directory Access Protocol (LDAP). This along with digital signatures, secure network access, data encryption ensures enforcing security at runtime based on the tenants configuration, moreover the proposed framework monitors the potential security breaches on the tenant data, manages data isolation between tenants services. These integrated measures ensure elevated application security by enforcing varied sets of custom security measures in application.

6. Evaluation and Discussions

6.1 Simulation setup

The application security risk analysis is done by Monte Carlo method [4] using automation software Arena [15] and our customized tool [12]. We simulate risk factors with inputs generated according to probability distribution model; this allows us to make optimal choices against composite cyber-attack models that otherwise cannot be easily identified. A risk level is defined as the probability of some defencelessness or weakness within a specific timeframe. Undesirable risks that occur either at the tenant or SaaS application provider side impact the system and reduces the availability, integrity, confidentiality of the MTA. Each risk is quantitatively measured using a scale that ranges between 1 and 6.

6.1.1 Parameters for evaluation

We use multi-variable defensive criterion \(X_i\) to evaluate the proposed model. 

\[X_1\text{-Unsafe Identify Management}, \ X_2\text{-Absence of Tenant application monitoring} \]

\[X_3\text{-Absence of finer Granularity of application and data access}, \ X_4\text{-Absence of Tenant data isolation}, \ X_5\text{-Absence of Secure Network access}, \ X_6\text{-Insecure Backups storage}, \ X_7\text{-Absence of High availability}, \ X_8\text{-Absence of Data recovery mechanism}\]

6.1.2 Risk classification

We consider 2 scenarios based on [11], scenario-I uses 3 categories of risks labeled a) Low b) Medium, and c) High; and in scenario-II we use of 6 categories of risks labeled as a) Not possible, b) Most unlikely c) Unlikely d) Likely e) Most Likely and f) Surely; for each of the variable, the defensive criterion \(X_i\) is expressed by probability distributions.
6.1.3 Discussions

The proposed SMA provides security isolation among tenants at various levels during runtime and for data offline. Our assessment of the proposed system in terms of probability of adversary risk factor shows that, the security risk associated in SaaS application is based upon multidimensional aspects. It can be expressed numerically as the product of the probability of occurrence and expected risk associated with one or many undesirable event that is defined in defensive criterion. Our evaluation is based on parameters in [6].

Our evaluation outcome in Fig.5 shows that the security risks of General Model (GM) and SMA using the 6, 3 risk factors respectively, the risk factor fluctuates in GM when compared to SMA. Moreover, the average risk factor of SMA is less than GM. This shows the effectiveness of the proposed solution in securing SaaS multitenant applications. Application security risks can be reduced by incorporating the pluggable SMA based security framework in SaaS application. Implementation the proposed SaaS security model increases the SaaS providers and tenants confidence and enables its speedy and wider adoption.

7. Conclusion and Future Work

Today, SaaS tenants demand transparency from the provider on how their data is secured at the datacenter. The challenge is to have a customizable tenant based approach that securely provide access to the tenant’s data and monitors any illegal access of both application and tenant data. We proposed an integrated Secure Multi-tenant Application (SMA) model that is custom defined and pluggable to multi-tenant application. Our evaluation based on the multi-variable defensive criterion and our discussions show that the proposed SMA framework is effective in reducing the security risk associated with multi-tenant application and alleviate the security concerns of the tenants.

Our future work is to integrate the security framework with the overall architecture of multi-tenant SaaS application.

References


