Two Level Trusted Approach for Ensuring Cloud Security

Shanty S R, Aby Abahai T

Abstract— Security is an important aspect in cloud computing and has numerous issues related with it. Inorder to adopt cloud computing by the enterprises and individuals, several issues have to be resolved. Security in cloud is an important issue that needs special attention and an important component of cloud security is trust management. The user should not face problems such as data theft or data loss. The cloud service provider and the user have to make sure that the cloud is safe from all external threats and attacks. This paper focuses on the development of a secure cloud environment to determine the trust of users. Only trusted users are allowed to access the cloud resources. The method proposed in this paper provides security from Denial of Service, Cross-site Scripting and SQL injection attacks. The uniqueness of the proposed system lies in the fact that it provides security at the service provider level as well as the user level in cloud environment.

Index Terms— Attacks, Cloud Computing, Cloud Security, Cross-Site Scripting, Denial of Service, SQL Injection, Trust Management

1 Introduction

Cloud Computing provides us a means by which we can access the applications as services, over the internet. Business applications can be created, configured and customized online using cloud. The term Cloud may refer to Internet. Cloud can provide services over public networks or private networks.

Cloud computing is sharing of resources on a larger scale which is cost effective and location independent. Resources on the cloud can be used by the client and deployed by the vendor such as Microsoft, amazon, rackspace, ibm, salesforce. It shares many on-demand tools and software for various IT Industries. Cloud computing has many advantages. The most important benefit of using cloud is that there is no need for the customer to buy the resource from a third party provider, instead they can use the resource and pay only for the use as a service thus helping the customer to save time and money.

For providing resources in the cloud, several cloud service providers are available. Their resources are hosted in the internet on virtual computers and make them available to multiple users. Multiple virtual computers can run on one physical machine and share the resources such as CPU, storage, memory etc. giving the feeling to the user that each user has his own dedicated hardware to work on. Thus virtualization enables the providers to sell the same hardware resources among multiple users. This sharing of the hardware resources by multiple users helps to reduce the cost of hardware for users while increasing profits of providers. In cloud computing selling or accessing hardware in the form of virtual computers is known as Infrastructure as a Service (IaaS). After procuring infrastructure from a service provider any operating system platform can be installed and run on it. Other types of services that are made available to the client through cloud computing are Software as a Service (SaaS) and Platform as a Service (PaaS).

To date from a small investor to a big IT company everyone is now relying on cloud. Cloud computing provides several benefits such as easy to use and maintenance, reductions in the overhead for storing and servicing the data, low power consumption for operation etc. In spite of these benefits cloud suffers from different security threats and risks to protect its resources from hackers and unauthorized users. These security threats and attacks are the biggest concern towards the improvement of a more secure cloud infrastructure. Existing methodologies are not enough to use for protecting cloud resources as they have no use with respect to the ever evolving security threats as well as to avoid data losses in the cloud environment. The data in cloud is not stored as it is, but rather this data is accessed by large number of times and gets changed in the form of insertion, updation or deletion.

The primary focus of this paper is to introduce a two level trusted security framework for securing cloud resources. Access to the cloud resources is provided only if they can be trusted. The remaining sections are organized as follows: Section 2 provides a state of the art security review in cloud computing environment. Section 3 will present the proposed security model in details. Section 4 presents the experiments and the results. Finally, section 5 concludes the paper.

2 State Of The Art Review

Providing data security is considered as the biggest factor in cloud computing [1]. Security challenges in a cloud computing environment may be classified as: Protection of data which is transferred towards the user, protection of data sent towards the service provider and protection of data in storage server or Cloud Data Center (CDC). Most of the security models so far are based on traditional cryptographic approaches. Some referred to dynamic security measures for a cloud environment [3] while other domain based applications discussed the growing security concerns of cloud infrastructure. In [2] domain trust concept is used to develop a secure cloud infrastructure. In [4], Jin-Song Xu et al, separates format and content from documents, before storing and handling of data in the remote cloud data center to protect cloud resources from unauthor-

ized users or hackers. An optimized authorization method (using encryption functions) is used for accessing database for trusted CSUs.

In Software as a Service, SQL injection attacks are one of the most important factors to be considered. In SQL injection attack [5], an attacker introduces malicious codes that the application passes to the back end database. By performing this, the attacker may be able to get the sensitive information in the databases. This is an important concern for the cloud storage from non-trusted CSUs. If an attacker injects malicious scripts into the Web contents then it is known as Cross-Site scripting attack. If any user clicks on them, the sensitive information will automatically redirects to the attacker machine. In cloud computing this may happen if any attacker affects the cloud service interface because cloud application requires Web interfaces for their services. In Denial of Service Attacks (DoS) a huge number of requests are sent by the attackers, thus the actual service become unavailable to the trusted users.

3 Proposed Cloud Security Model

Two-tier cloud security architecture has been proposed in this paper for providing security to cloud resources. The first level is the Broker Domain and the other one is Cloud Service Provider Domain. The system architecture is shown in Fig 1. The Broker Domain is denoted by Level 1 and Cloud Service Provider Domain is denoted by Level 2.

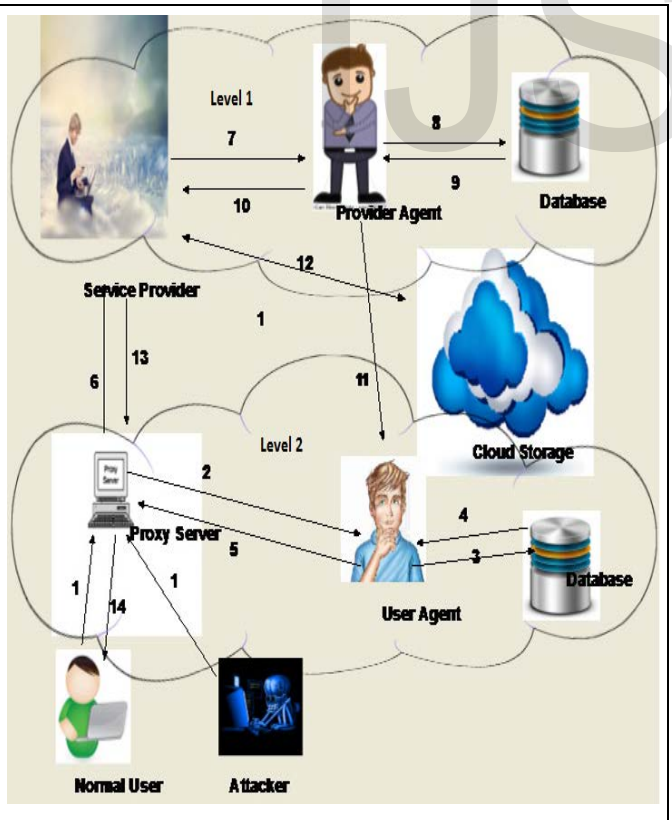

Fig.1. Proposed cloud security framework.

This two level approach for communication authenticates the trusted cloud users for accessing information or resource from cloud. Proposed technique ensures the trustworthiness of the system by calculating trust value for each service requested by the user and the domain from where the request is coming. In this model the user access resources in the cloud by requesting information from the Cloud Service Provider.

As the first step, when any user who wants to access resource from the cloud sends the request. It has to pass the correct authentication data through a proxy server situated in its domain (Broker domain). The proxy server is used as a communication channel between the two domains. For example if two users, User A (normal user) and User B (Attacker) denotes two specific group of users who requires access to their account details from a bank hosted in cloud. Their requests are passed to the provider via the proxy server. When the request is passed through the proxy server it reaches to the user agent situated at broker domain. The trust value of the service requesting user is checked by this user agent with its local database present in the same domain. If the calculated trust value for this request is greater than the previous value, then only the request is forwarded to the cloud service provider situated in the Cloud Service Provider Domain. When this request reaches the cloud service provider domain, it's immediately passed to the provider agent to check the trust value of the domain from where this request arrived. The provider agent then calculates the trust value for this domain and then sends the requested information back to the provider, only if the calculated trust value is greater than the previous value. Then the provider will allow access to the requested information to that user through the proxy server. The agent in Broker Domain and the Provider Domain will update the trust value after performing the task. If the calculated trust value is less than that of the trust value in the database, the agent in the provider will inform this to the agent situated at broker domain for taking necessary actions. The user agent will in turn decrease the trust value for this particular user. When the trust value reaches the threshold value the user will be added to the blocked list and will not be able to access the information further. The main advantage of this approach is that, domain remains unaffected when a said non-trusted user does malicious activities in the system. The trust value of the domain will decrease accordingly with the malicious activities and updating policies. The agents in both levels have their own databases, for maintaining user activities information and for storing the updated trust value. The flow of information as denoted in Fig 1 is discussed below to understand the process of the proposed model.

Step 1: User provides the request to the proxy server for obtaining information

Step 2: When the information reaches the proxy server it forwards the request to the user agent in the broker domain.

Step 3: The user agent checks whether the request is malicious or not and calculate the trust value and checks with the database.

Step 4: If the value is greater than the previous value for this domain the request will be forwarded to the agent.

Step 5: The agent will forward the request to the proxy server.

Step 6: The proxy server will forward the request to the cloud service provider

Step 7: The provider will pass the request to the provider agent.

Step 8: The provider agent checks whether the request is malicious or not and calculate the trust value and checks with the database.

Step 9: If the trust value is greater than the previous value for this domain the request can be granted.

Step 10: Provider agent forwards the request to the provider.

Step 11: If the trust value is less than the previous value the provider agent will sent the new value to the user agent in the broker domain.

Step 13: The provider gets the required information from the cloud storage.

Step 14: The provider sends data to the proxy server.

Step 15: Proxy server sends data to the user.

3.1 Architecture and Design

The architecture of the proposed trust based model is given in Fig 2. As mentioned earlier the model consists of two levels. The first level consists of the proxy server and the agent. The second level consists of a service provider, agent and the cloud storage which contains the resource.

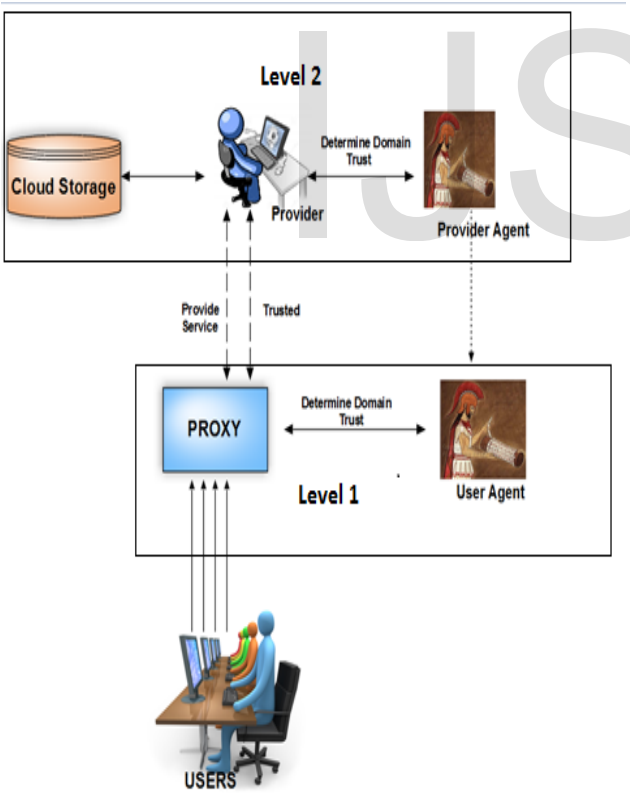

Fig. 2 Proposed two level cloud security architecture.

The components in the two levels are discussed below

User: Request resources from cloud storage.

Service Provider: Provides access to the requested resource to user if it is a trusted one.

Proxy Server: Used for monitoring request of the user.

User Agent: Determine whether the request is malicious or not and calculate the trust value in level 1.

Provider Agent: Determine whether the request is malicious or not and calculate the trust value in level 2.

3.2 Algorithm for Requesting Information

A cloud service user α sends request δ to cloud service provider S for information ε. The request δ is passed as input to the algorithm.

Input: Service Request (δ)

Output: Forward request to provider S or reject request

Begin Algorithm

1. Proxy server extract the url of the information received from the user α
2. Proxy Server sends the extracted data to the user agent
3. User agent calculates the trust value V by checking whether it contains Sql injection, Cross-Site scripting or denial of service.
4. If V > value in database
5. The request δ for information ε is passed to the provider S;
6. Else
7. The request δ is dropped;
8. Update trust value
9. If V< threshold
10. Block the user

End.

3.2 Algorithm for Delivering Information

The second algorithm is for delivering information. This algorithm is called when the request δ reaches the provider S.

Input: Service Request (δ)

Output: Permission for access to the information ε or reject request δ

Begin Algorithm

1. Provider S sends the request to the provider agent
2. The Provider agent checks the request and calculates the trust value
3. If V> value in database
4. Send to ε user α via Proxy Server;
5. Else
6. Provider agent sends the trust value to user agent
7. Drop the request δ;
8. Update the trust value
9. If V< threshold
10. Block the user;
11. User agent updates the trust value

End.

4 EXPERIMENTS AND RESULTS

Experiments are conducted by performing normal activities and malicious activities. Malicious activities include performing Sql-injection, Cross-site scripting and Denial of service attacks. The two activities are distinguished by checking the url. If the url is free of any of the attacks the requested re-

source will be granted. If the url contains any kind of vulnerable strings then it will be detected and the request will not be granted. The trust value will also be updated to block the user if it falls beyond the threshold. The trust value is calculated using the equation,

$$v = (1 - n/t)^m \quad (1)$$

where m is the level, n represents the number of negative operations performed by the user, t is the total number of operations performed.

4.1 Detection of Sql Injection Attack

Fig 3 represents performing an Sql injection. If instead of the username or password if we give the input as ' or 1 ='1 the condition will always be true and we will be able to access the resource. We will be logged as the user who had registered first in the database. This attack can be detected by using the regular expression (.*)'\\sor\\s\\d*\\s='\\d*(.*). It is also able to log on as a particular user if we know his user name. In the username field if we provide the username and put a space and --, the rest will be treated as comment and password will not be checked. Inorder to overcome the validation in the form field we can submit any password. As the password is treated as a comment it will not be checked. Similarly we can perform different types of attacks and will be able to update the passwords of users or drop the tables that the application uses. By performing the union attack we will be able to get the table name and column names of the database used in the application. Different regular expressions are given to detect each of the possible sql injection attacks.

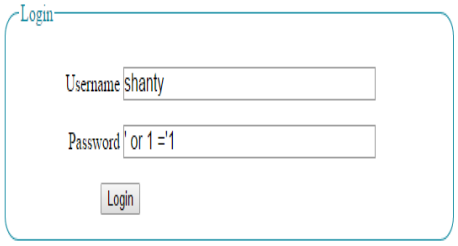

Fig.3. Performing sql injection attack

4.2 Detection of Cross-Site Scripting (XSS) Attack

XSS can allow an attacker to add their own malicious script code onto the HTML pages displayed to the users. Fig 4 represents cross-site scripting attack. The attack shown in Fig 4 will make the users view video in youtube. Once executed by the user's browser, script inserted by XSS could then perform actions such as stealing private data, completely changing the behavior or appearance of the website, or performing actions on behalf of the user. It can also take the users to some other site which may steal the credentials of the normal users. Many XSS attacks were performed. The XSS attack is detected by checking the presence of script, <,> etc in the data submitted by the user.

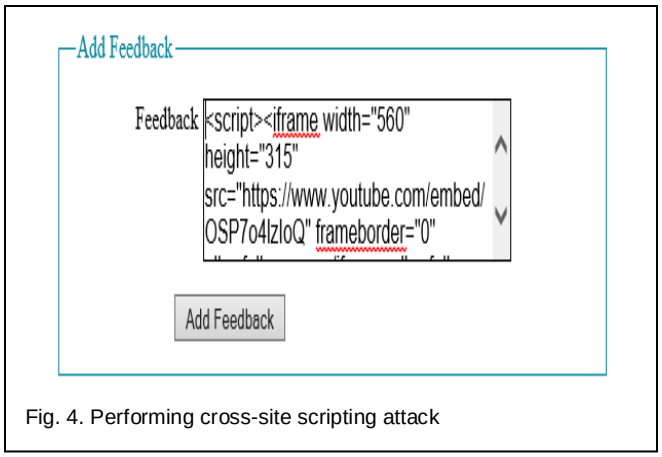

Fig. 4. Performing cross-site scripting attack

4.3 Detection of Denial of Service Attack

Denial of service attacks are performed by flooding the server with large number of request so that the server will be unable to process the requests of legitimate users. These attacks are detected by checking whether the number of request for a particular resource from a particular ip address is exceeding the request rate. Request rate is the number of request that the server can process in one minute. After each attempt the trust value of the user performing DoS attack will be decreased. When the value reaches the threshold the user will be blocked. So the effect of DoS will not affect normal users. So atleast assigning them a particular request limit for a particular time interval may increase the possibility of protecting the server from denial of service attack.

5 Conclusion

The proposed approach tries to protect the cloud resources from Sql injection, Cross-Site Scripting and Denial of service attacks. User's request is monitored by the proxy server which forwards the request to the service provider only if the request is a trusted one. Otherwise the request will be rejected. The service provider provides access to the request sent via the proxy server if it can be trusted. The proposed approach maintains reputation of the two domains. Future work can include detection of more types of attacks such as DNS attacks, Distributed denial of service attacks etc.

References

[1] Cyril Onwubiko, "Security Issues to Cloud Computing", in Cloud Computing: Principles, Systems and Applications, Computer Communications and Networks, N. Antonopoulos and L. Gillam (Eds.), Springer-Verlag London Limited 2010, DOI 10.1007/978-1-84996-241-4_16, pp. 271-288, 2010.

[2] Wenjuan Li and Lingdi Ping, "Trust Model to Enhance Security and Interoperability of Cloud Environment", *Proc. of CloudCom 2009*, Springer-Verlag Berlin Heidelberg 2009, LNCS 5931, pp. 69–79, 2009.

[3] Jeff Sedayao, Steven Su, Xiaohao Ma, Minghao Jiang, and Kai Miao, "A Simple Technique for Securing Data at Rest Stored in a Computing Cloud", *Proc. Of CloudCom 2009*, Springer-Verlag Berlin Heidelberg 2009, LNCS 5931, pp. 553–558, 2009.

[4] Jin-Song Xu, Ru-Cheng Huang, Wan-Ming Huang, and Geng Yang, "Secure Document Service for Cloud Computing", *Proc. of CloudCom 2009*, Springer-Verlag Berlin Heidelberg 2009, LNCS 5931, pp. 541–546, 2009.

[5] Marios D. Dikaiakos, Dimitrios Katsaros, Pankaj Mehra, George Pallis and Athena Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research" in IEEE Internet Computing, IEEE Computer Society, pp. 10-13, September/October 2009, Vol. 13 No. 5, DOI: doi:10.1109/MIC.2009.103

[6] Char Sample, Senior Scientist, BBN Technologies, Diana Kelley, Partner, Security Curve, "Cloud computing security: Routing and DNS security threats," http://searchsecurity.techtarget.com/tip/0, 289483,sid14_gci1359155_mem1,00.html

[7] Siani Pearson, Yun Shen and Miranda Mowbray, "A Privacy Manager for Cloud Computing", in CloudCom 2009, LNCS 5931, M.G. Jaatun, G. Zhao, and C. Rong (Eds.), Springer-Verlag Berlin Heidelberg 2009, pp. 90–106, 2009.

[8] Florina Almen´arez, Andr´es Mar´ın, Celeste Campo and Carlos Garc´ıa R., "PTM: A Pervasive Trust Management Model for Dynamic Open Environments", Proceedings of First Workshop on Pervasive Security, Privacy and Trust PSPT'04, Boston, MA, USA, 2004.