International Journal of Web Technology
Volume: 05 Issue: 01, June 2016, Pages: 17-21
ISSN: 2278-2389

A survey on Insecure API Challenges and Resolution Mechanisms on Cloud Computing

P.Sujatha[1],M.J.Balachandran[2]
[1]Head and Associate Professor,Department of Computer Science,Vels University,Chennai, India
[2]Research Scholar, School of Computing Sciences,Vels UniversityChennai, India
Email:suja.research@gmail.com,mjbalachandran@yahoo.co.in

Abstract - Nowadays, Cloud computing provides extended architecture which can be accessed through internet and also it reduces the setup of computing infrastructure for the IT based solutions and services. Cloud computing is a sharing of configurable resources like Application-as a service, Platform as a services and infrastructure as a services. Different kind of cloud services on the Internet exposed by application programming interfaces. As the APIs are accessible across the Internet, malicious attackers can compromise the confidentiality and integrity. In this paper, one of the top security threats of cloud computing shared by Cloud Security Alliance (CSA) has been analyzed.

Keywords: Application programming interface, insecure API, Activity role based Access & Cloud computing Challenges.

I. INTRODUCTION

Cloud computing supports the large storage capabilities and computational power. Recent past, tier3 and tier2 companies are moving to Cloud due to various reasons like minimal hardware, maintenance cost, payment based on usage, accessible demand based security controls, faster deployment, scalability, flexibility, independent location and the sophisticated automated process i.e., customer doesn't require software up-gradation. In this paper, one of the security issue is discussed and proposed a mitigation techniques. The interface is Insecure Application Programming Interfaces (APIs). Insecure Application Programming Interfaces grants access to third-party by exposing the applications to Application Programming Interfaces (APIs), such units may be required to relinquish their access credentials to third parties in order to enable their usage.

A risk which relates to illegal authentication will lead to security issues of the application. Iintrusion risks are higher in cloud environment which blocks visibility to network-based systems especially accessing through the channel of insecure APIs. Several root cause of the most application security issues lead to bad application development practices. Such kind of cloud environments also use applications developed for in-house deployments therefore it causes more bugs. Hence, the impact posed by insecure APIs cannot be underestimated.This paper detailed the ways the threats can be launched in cloud, the modernized ways to exploit these threats and their impacts on cloud. The threats are analyzed in detail and presented the resolutions of security for the prevention of these threats from literature. The purpose is to propose a proper access control model for enterprise environment through the integration of role based and activity based access control.

II. LITERATURE SURVEY

Emmanuel Adinkrah, J.B Hayfron-Acquah & Joseph Kobina Panford [1], two of the security issues are discussed and respective mitigation steps are given. They are Insecure Application Programming Interfaces (APIs) and Data Loss/Leakage. The purpose of this paper is to find out the benefits and drawbacks in regarding data security and data availability in a cloud based ERP; the concluding factors in terms of security of data and data availability, while adopting Cloud Computing, organizations should keep in mind the effective and efficient use of their information system. Muhammad Kazim, Shao Ying Zhu, University of Derby, UK, [2] elaborates the survey of top security issues related to cloud computing is analyzed. For each of these security threats author describes, i) how it used to exploit cloud components and its effect on cloud entities such as providers and consumers, and ii) the solutions of security that must be consider to prevent these threats. The focus is on most top threats on cloud computing which is considered relevant by most consumers and businesses.

Vineet Kumar Singh[3] briefed some efficient solutions, which is helpful in real time environment. He conducted an analytical performance analysis between different cryptographic algorithms done on the basis of Total Execution Time of the algorithms and their respective Speed up ratios on different Input Size. Comparative study analysis of different cryptographic algorithms revealed many facts about cryptographic algorithms execution on cloud network. Surbhi Aggarwal [4] analyzed and juxtapose the two hash algorithms, MD5 and SHA, using various key features and performance metrics. Their features have also been highlighted in order to provide the researchers a better comparative picture so that this can be reached to the final position, which algorithm has superseded the other. This paper proposed that the SHA algorithms should be given atmost importance in comparison to MD5 as SHA algorithms' performance is supercedes other cryptographic functions of hash algorithm Shelveen Pandey, Mohammed Farik [5], Author has discussed some top threats and the security techniques that can be adopted as countermeasures. Even though the counter measures are mentioned, author has not given a proven solution for security but to minimize threats to some extent in the future.

III. REASONS FOR API THREATS

In this section major security mistakes and threats with respect to API is discussed and its respective reasons are analyzed. These are:

International Journal of Web Technology
Volume: 05 Issue: 01, June 2016, Pages: 17-21
ISSN: 2278-2389

A. Insecure interfaces and APIs, malicious insiders, technology vulnerabilities and mis-handling of cloud computing services.
B. Analysis of API security mistakes - and how to avoid them

A. Environment specific Threats

Providers of Cloud service are majorly responsible in controlling the cloud environment. Based on some of the survey reports, it shows that fifty percent of the cloud consumers consider the issues from cloud service provider which causes major threat in cloud computing, some of the other threats are specific to cloud computing such as providing insecure interfaces and APIs to users, misuse of cloud services, malicious cloud users, threats of shared technology and insufficient due diligence by companies before moving to cloud. Insecure Interfaces and APIs: Application Programming Interface (API) is a set of standard protocols which defines the communication medium between software applications and internet. Cloud APIs are used at all the infrastructure, platform and software service levels for an effective communication with other services. IaaS-API used to access and manage infrastructure resources like virtual machine and network, PaaS-API gives access to the cloud services such as storage and SaaS-API connect infrastructure with software applications. Cloud services security depends on the APIs security. Weak set of APIs and interfaces cause many security issues in the cloud. In general, service providers provide services to third party through their API. Third party having access to security keys and sensitive information in the cloud is due to weak APIs. The encrypted customer data in the cloud can be read with the security keys which results in loss of data integrity, availability and confidentiality. Access control principles and authentication are violated via insecure APIs.

B. Analysis of API security mistakes -- and how to avoid them

At the time of development of application, coders to keep the API security in mind, they are failing to write secured code resulting in putting both the application and the underlying data at risk. Poorly written code becomes dangerous when an API is involved.During the building of an application API code should be manually checked (before the release) by the project leaders or security expert to test whether it could be vulnerable by an attacker. Documentation is also vital. Properly documented code allows reviewers to see exactly what the APIs to be or not to be performed, and also the developers incorporating the APIs into an application to understand how to implement them correctly.During documentation, coders should specify how to call the API, what kind of format, what data will be returned, and specific error messages can be expected. Internally maintained records also note what information will be logged to capture who, what and when resources have been accessed for audit purposes, who can access the API etc., For access related, machine IDs must ensure authentication checks as appropriate. Every API call must be checked to ensure that the user or device has the right permissions to modify, view or delete the requested data. Post authentication of a user, many coders ignore secondary access control checks.

Fuzzing and black-box testing are critical to watch how APIs handles unexpected inputs and requests. Validation of data routines to prevent standard injection flaws and cross-site request forgery attacks are also important, as calls to APIs are coming from unreliable sources. In addition to this, testing plays an important role and it must be carried out from all types of key endpoints, not just from a Web browser. When there is an access communication via a non-browser application, many APIs fail to implement SSL such as a mobile applications, hence atmost importance to be given to check that sensitive data to be encrypted when not required in plaintext. More focus to be on Penetration tests and vulnerability assessments because APIs are the entry points of the application.Primarily, in many in house organizations and third party organizations reuse the existing source code found on the web, especially when searching for an example of how to call a particular API. General way of introducing API-related vulnerabilities are due to improper checking, validating, copying and pasting of source code.

IV. PROTECTION TECHNIQUES AND ITS FLAWS

In the below section the security methods denoted to avoid the exploitation of threats mentioned in the above section III are discussed. Here, various security techniques are detailed to secure cloud from threats.

Protection from Insecure Interfaces and APIs:

While the process of protecting the cloud from insecure API threats, it is important to note that the designers should design these APIs by below mentioned the principles of trusted computing. Providers of cloud should ensure that all the implemented APIs in cloud are securely designed, and also make sure on deployment for possible flaws. To secure data and services from insecure interfaces and APIs, proven authentication mechanisms and access controls should be implemented. Before development, application developers can get the help of the standards and guidelines provided by Open Web Application Security Project (OWASP) and develop secure applications that can help in avoiding such application issues.

Vendor to adhere standards such as SAS-70 (Statement on Auditing standards) for information security and protection of data. Also, users to analyze the interfaces and APIs of cloud provider before moving the data to cloud. There are multiple access control methods to access a resource in computing systems.

A. Access Control Matrix (ACM)

ACM provides the rights of every subject to each single object in a table. The primary aim is to offer a access to authorized users (Subject) can access information resources (Object). Subjects are labels for processes, while objects are generally labels for files being accessed. When an organization has a multiple or huge number of users, it is difficult to store and managing their related information. Hence, currently there are performance and scalability issues arising in ACM.

B. Access Control Lists (ACL)

Each object has all its permissions list and also the object has all the subjects that have access rights. Here, object who has the access can be easily found out, but access rights of a user is very difficult to find. Access control list can be implemented in latest operating systems and grants or denies access for the

International Journal of Web Technology
Volume: 05 Issue: 01, June 2016, Pages: 17-21
ISSN: 2278-2389

mentioned user or user groups. ACL lacks flexibility when the organization is large, where the policy of access control changes constantly.

C. Discretionary Access Control (DAC)
DAC is purely based on ownership and it doesn't provide a high degree of security. It is basically depend on the discretion of an object's owner who is authorized for the information control on resource access. Assurance cannot be guaranteed on the security when an owner is given a read access for a file to another user, the user can still copy the contents of the file. The access rights are decided by the owner and do not follow the organization's security standards.

D. Role-Based Access Control (RBAC)
In this access control mechanism, roles are assigned to users and access rights or permissions. The basic components of mechanism is user, role, session and permission. Hierarchy of role allows supervisor / authorizer roles to inherit roles from junior roles. It is a passive access control model, it lacks in capturing dynamic responsibilities of users to support workflows, which requires dynamic activation of access rights for certain activities. The challenge of RBAC is the contention between strong security and easier administration. For stronger security, it is better for each role to be more granular, thus having multiple roles per user

E. Mandatory Access Control (MAC)
A central authority is a decider or a decision maker on who has access to what information and this Mandatory Access Control is widely accepted or used in military security. The owner doesn't have permission to change their respective access rights. A labeling mechanism is accepted to find the MAC policy. For instance when an user is labeled with 'secret' code is not allowed to read a file with the 'top secret' code label. For task execution, security labeling in MAC is not flexible and is not convenient.

F. Activity-Based Authorization Control (ABAC)
Permissions are activated or deactivated based on the current task or process state. The Protection step is an authorization step and with each authorization step the usage count is incremented by one. When the multiple usage or the number of count based on the usages exceeds or reaches the limit, the respective permissions are deactivated. It is an active access control model based on tasks. However, there is no separation between tasks and roles.

G. Activity role based access control model
In Activity-Role-Based Access Control (ARBAC), users mapped with permissions through roles and tasks. It is an active access control model and delegation of activities or tasks when one user is not available is to be introduced. This model has constraints on application for enterprise environment, it is necessary to investigate a proper model for enterprise environment.

V. SUGGESTED MECHANISM

When we want an API is secured, the general trend is to have different authentication techniques like passwords etc to validate the identity of a user and to ensure that the authorized user access the cloud services.In cloud computing, the provisioning of any of their services, orchestration, monitoring and their management are all done using the cloud API. Hence, just validating the fact that the consumer indeed belongs to the right organization or tenant for which this particular service is intended is not sufficient. When an intruder, malicious insider or an employee of an organization, tries to misuse its rights to modify the configuration setting, parameter or others features through the cloud API, the proposed mechanism using access control policies along with authentication mechanism to achieve a secure cloud API. Until now, Role based access control mechanism is familiar and widely used because its users-roles permissions mapped based on the business processes and their respective assignment of work. However, few modifications to be incorporated especially in cloud computing. It fails to capture the dynamic changes in the user responsibilities because it is a passive model. Hence, based on the survey it is proposed to include Activity-Role Based Access Control mechanism in which consumers have relationships with permissions through roles and their respective activities.

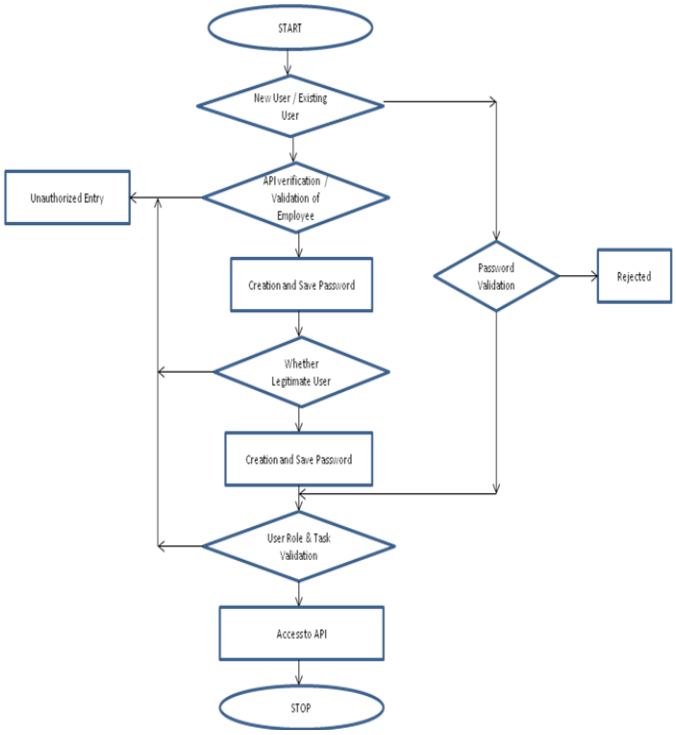
It is based on the analysis referred in the below given references. This gives and ensure the nature of access control mechanism dynamic in nature. When a user completes his or her activity, respective access rights are revoked and will not be entertained to access, until their access been granted permission again. Below process steps to be followed when a consumer requests to access the cloud resources via its API:
a) When a new user registers at the API by filling various fields like name, organization, designation, business need and tenant name else go to step (d).
b) First, the API verifies whether the right tenant name is entered and also validates the employee name against matching it with the employee id which is pre stored in one of the servers at the cloud service provider end. If the validation is success and output is true go to next step, else consumer's request is rejected, alert notification will be sent to the tenant about an unauthorized attempt.
c) User is suggested to create and save a password for subsequent validation of his / her identity in the near future. Go to step (e).
d) If it is an existing user, Password entered by the user will be validated by the API after computing its hash and matching it with the corresponding value precomputed and saved in a separate table. If authentication is true, go to step (e), else request is rejected, notification alert will be sent to the tenant about an unauthorized attempt
e) API will permit a request if a legitimate user is accessing. API validates the user's role once the user puts his or her request and maps it with the activity he or she has been granted permission to work further.
f) If mapping is succees between the role and the respective activity, the user is approved and allowed by the API to perform the activity, else notification alert will be sent to the tenant about an unauthorized entry giving information about the malicious insider / employee.

i. Performance Analysis
This section demonstrates the model performance for ARBAC. There are two aspects in the study analysis: on one hand, comparative analysis the initialization time of traditional

International Journal of Web Technology
Volume: 05 Issue: 01, June 2016, Pages: 17-21
ISSN: 2278-2389

RBAC model and ARBAC model. Specifically, to compare the time overhead of establishing roles table respectively with RBAC access control policy and ARBAC access control policy, and to test the change of the running time for two models with increasing number of roles. On a flip side, analyzed running efficiency of decryption algorithm in RBAC and ARBAC access control systems. In the analysis, cipher text-policy attribute-based encryption algorithm is used to execute the assignments of permissions, and to check the efficiency of decryption.

ii. Process Flow Algorithm



iii. Comparative study of RBAC and ARBAC

S.No	ARBAC	RBAC
1	Dynamic role allocation	Static roles allocation
2	Performance advantage will be more obvious with the increasing of role numbers	Less performance when increase in number of roles
3	Role classification will largely reduce the number of role rules	No or limited role classification
4	Less system cost	More system cost
5	Efficiency of system initialization is significantly high	Efficiency of system initialization is low
6	For Decryption algorithm – It has better time performance	Time performance for decryption algorithm is limited or slow.

VI. CONCLUSION

In this paper, Although widely used access mechanisms are discussed, multiple combinations, many extensions and different mechanisms are possible. Trade-offs and limitations are involved with all mechanisms and access control designs, so it is the user's responsibility to determine the best-fit access control mechanisms that work for their business functions and needs. Approach mechanisms are currently being reviewed in many IT industry. When compared with the RBAC model, Activity role based access control model has better performance in supporting fine-grained authorization, simplifying policy management, having scalability, supporting expansion of roles dynamically and, has higher operational efficiency. I have analyzed the relationship of roles and attributes, establish a mapping between roles and static attributes and dynamic attributes, formally define the multi-authorized relationship of users, roles, attributes and permissions, and finally suggesting the access control algorithm of ARBAC model. But I still feel some bottlenecks of a wider range of large-scale access control systems. There is no full proof mechanism for security even though the countermeasures are in place. The mentioned solutions based on the study can be implemented to minimize threats to some extent in the future.This paper is specially intended to analyze the best mechanism to overcome the current challenges in API, for this research I thank my guide Ms.Sujatha for her review and her motivational background helped to narrow down the analysis.

REFERENCES

[1] Emmanuel Adinkrah J.B Hayfron-Acquah & Joseph Kobina
[2] Panford - Mitigation Techniques to Security Flaws in Cloud
[3] Computing, "International Journal of Emerging Technology and Advanced Engineering" Volume 5, Issue 6", June 2015)
[4] Muhammad Kazim, Shao Ying Zhu, University of Derby, UK,. "A survey on top security threats in cloud computing", International International Journal of Advanced Computer Science and Applications, Vol. 6, No. 3, 2015
[5] Vineet Kumar Singh, "Analyzing cryptographic algorithms for secure cloud network", "International Journal of advanced studies in Computer Science and Engineering", Volume 3 , Issue 6, 2014
[6] Surbhi Aggarwal, "A review of Comparative Study of MD5 and SHA Security Algorithm", "International Journal of Computer Applications (0975 – 8887)", Volume 104 – No.14, October 2014
[7] Shelveen Pandey, Mohammed Farik, "Cloud Computing
[8] Security: Latest Issues & Countermeasures", "International Journal of Scientific & Technology Research", Volume 4, Issue 11,November 2015, ISSN 2277-8616
[9] Prof.Rajesh Kumar Kashyap, "Security challenges and issues in
Cloud Computing – The way ahead" International Journal of Innovative Research in Advanced Engineering (IJIRAE) ", Issue 9, Volume 2
[10] Shraddha More & Rajesh Bansode, "Implementation of AES with Time Complexity Measurement for Various Input"

International Journal of Web Technology
Volume: 05 Issue: 01, June 2016, Pages: 17-21
ISSN: 2278-2389

[11] "Global Journal of Computer Science and Technology: ENetwork, Web & Security", Volume 15 Issue 4 Version 1.0
Year 2015.
[12] Noura Aleisa, "A Comparison of the 3DES and AES Encryption Standards", "International Journal of Security and Its Applications", Vol.9, No.7 (2015) pp.241-246
[13] Krunal Suthar, Parmalik Kumar, Hitesh Gupta & Hiren Patel,"Analytical Comparison of Symmetric Encryption and Encoding Techniques for Cloud Environment", "International Journal of Computer Applications (0975 – 8887)", Volume 60– No.19, December 2012
[14] Dr. Prerna Mahajan α & Abhishek Sachdeva, "A Study ofEncryption Algorithms AES, DES and RSA for Security", "Global Journal of Computer Science and Technology Network, Web & Security", Volume 13 Issue 15 Version 1.0 Year 2013
[15] Kumar Gunjan #1, R. K. Tiwari *2, G. Sahoo "TowardsSecuring APIs in Cloud Computing" International Journal of Computer Engineering & Applications, Vol. II, Issue II