International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August, 2019

Authentication and Authorization Mechanism for Cloud Security

J. Vijaya Chandra, Narasimham Challa, Sai Kiran Pasupuletti

Abstract: ***Cloud Computing is a most widespread and popular form of computing, promising high reliability for customers and providers both at the same point of time for many fields, where cloud storage security is based on Authentication and Authorization in cloud computing. Data is uploaded into a cloud and stored in a datacenter, Authentication and authorization are the major concerns to access the data stored in cloud by users from the data center. Security is a major issue; these are mainly deal with identity and access management, prevention of data loss and malware attack control management. In this paper we majorly concentrated on Authentication and Authorization to cloud access, we focused on identity management mechanism as cloud security solution that provides directory services for application access management. We discussed the protocols that support authorization and allows the communication across applications with the help of tokens instead of credentials. We even concentrated on the different mechanisms which plays a major role in designing a secured cloud computing architecture from malicious intrusions and attacks, it is a step to verify the presence and functioning of the cloud customers and cloud providers through security mechanisms to protect from different risks, threats and attacks. In this paper we discussed different security Algorithms and Authentication architecture along with the proposed algorithm, where analysis is done along with the computational evaluation with output.***

Index Terms: ***Authorization, Authentication, Auditing, Confidentiality, Integrity, Accounting, Cloud Security.***

I. INTRODUCTION

Authentication is a primary security service; the most commonly used procedure is verification of username and password; it is the processes of verifying who you are? and Authorization is the processes of giving boundaries that is how much you can be accessed. Auditing consists of examination based on previous history to determine whether security violations took place. Audit data is recorded in audit log files, continuous monitoring system and auditing procedures should be followed by an intelligence system for securing the cloud. A secure method for the initial distribution of passwords is for the user should be authenticating by the cloud administrator. Several Important concepts that are used in Cloud Security and storage maintenance are Identification, Confidentiality, Authorization, Accounting, Authentication, Auditing and Privacy. Once user identification and Authentications are established, then authorizations levels will be determined. Cloud computing environment is reliable, based on authentication, authorization, accountability and auditing where accountability is the major character which can check performance, actions and behavior of a system, where audit trail or logs supports accountability. By Audit we mean the review of data for its integrity, it is no doubt that the procedure is used to check all the functions, namely; standard, methods or practices are being followed by the organization or not. The cloud security majorly focuses on CIA triad and AAA model which is the process of affirming the correctness of information stored in the cloud [1]. An open protocol for authorization OAuth is used for conventional web environment that enables the client applications on HTTP Services, which allows communication across applications with the help of tokens instead of credentials. It permits sharing of resources from one site to another site without using their identifications, authorizations and authenticated credentials. It practically bridges the authentication between the service providers and end-user with higher entropy towards the security [2]. It is a secured protocol that relies on secure sockets layer that ensures data between web server and end-user, and for the valid clients after verifying the clients authorization credentials the client access with the authorization server and requests to access token and then uses tokens, which is used to ensure cryptography protocols to provide stronger authentication and data security, it allows limited access based on time, when authentication token expires, its logout the session [3]. Open ID connect provides an identity layer, that will be top of the authorization server layer stack, it verifies the client's identity, it transports data about the user as token which contains basic profile and authenticates the identity of the user. OpenID connect is suggested if a web-based application is hosted on a server and accessed via a web browser. based on the clients credentials, the authentication server sends the access token, where authorization code will be activated which allows accessing the authorization request and grant access to the client application to fetch the owner resources,

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
J.Vijaya Chandra*, Research Scholar, Dept of CSE, KLEF (Deemed to be University), Guntur, Andhrapradesh, India, vijayachandra.phd@gmail.com.
Dr. Narasimham Challa, Dean IQAC and Professor, Dept of CSE, Vignan's Institute of Technology and Science, Vishakapatnam, A.P., India. narasimham_c@yahoo.com
Dr.Sai Kiran Pasupuleti, Professor, Dept of CSE, KLEF(Deemed to be University), Guntur, Andhrapradesh, India.
psaikiran@kluniversity.in

if the token access request, which is issued by the authorization server is invalid or unauthorized, then the authorization server returns an error response. The refresh token can be used by the client to acquire a new access token [4].

II. RELATED WORK

A. Microsoft Azure Active Directory

Microsoft Azure Active Directory is a multi-tenant cloud based directory and identity management service, it is a flow of requests and responses for the authentication process, which is determined by the authentication protocols, OpenID connect is recommended from Microsoft, if web application is hosted on a server and accessed via a web browser, they connect standards make and extensive use of bearer token including JWT tokens. Microsoft Azure Storage provides scalable, durable and redundant storage. Azure AD provides identity protection and identity management services to provide the application access management and combines core directory services [5].

B. OAuth

OAuth is an Authorization Protocol rather than Authentication Protocol, It is open standard for access delegation, It provides to clients a secure delegated access to server, resources on behalf of a resource owner, OAuth is a framework that enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by arranging an agreement to approve interface in between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf [6]. The resource server is the server that hosts the protected resources, which can accept and respond to protected resource that requests using access tokens. The authentication server issues access tokens to the client after successfully authenticating the resource owner for obtaining authorization.

C. OpenID Connect

It extends the OAuth Authorization Protocol and can be used as Authentication Protocol, which is used to sign in securely the users web application. OpenID Connect introduces the concept of an ID token, which is a security token that allows to verify and identify the user. It is used to acquire access tokens for security, the different tokens generally used are access tokens, Bearer tokens and Refresh Tokens. The Bearer token is a lightweight security token that grants the “bearer” access to a protected resource. Bearer tokens doesn’t have any built-in mechanism for preventing unauthorized parties from using them and hence they must be transported in a secure channel such as transport layer security. A refresh token signifies a continuing authorization of a certain client to access resources on behalf of a resource owner. Such tokens are switched between the client and authorization server. Clients use this kind of token to obtain ("refresh") new access tokens used for resource server invocations [7].

D. Redirect URI

A redirect URI helps to detect malicious clients and prevents risks, threats, vulnerabilities and phishing attacks from clients attempting to trick the user into that trusting the phisher as the client. The value of the actual redirect URI used in the authorization request must be presented and is verified when an authorization "code" is exchanged for tokens. It is included in the authorization request; the authorization server must compare and verify the value received [8].

III. SECURITY CONSIDERATIONS

Security Considerations are managed based on deployment, associated to different characteristics, formats, procedures and mechanisms such as Token scope values, Token formats, Resource URLs, communication between authorization server and resource server. The authentication server consists of accessible data elements such as client ids and encrypted code, usernames and passwords, client specific refresh tokens and access tokens, HTTPS Certificate/Key and Redirect URI. The Resource server consists of user data, HTTPS Certificate/Key and access tokens [9].

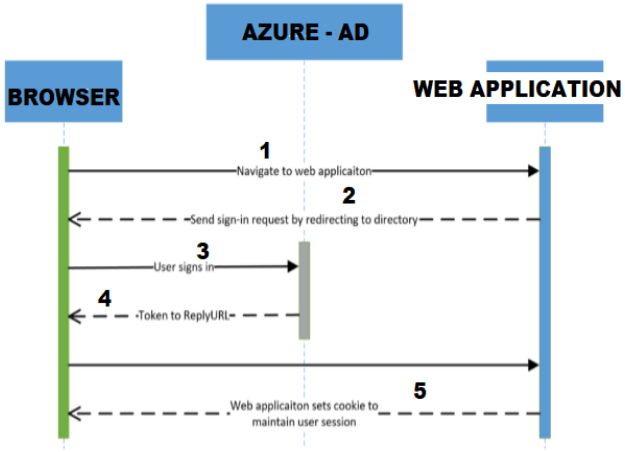

Figure 1: The Basic Authentication in AzureAD

The Basic Scenario of Authentication in Azure AD consists of the following steps as shown in the above sequence diagram.

1. Navigate to Web Application
2. Send sign-in request by redirecting to directory
3. User Signs in
4. Token to ReplyURL
5. Web application sets cookie to maintain user session

A user visits the application and needs to sign in, they are redirected to Azure AD authentication end point, that is User sign-in the page, If authentication is successful, Azure AD creates token and returns to the application Reply URL that was configured in Azure Portal then the application validates token by using public signing key and issuer information. Finally Azure AD starts a new session where in user access the application until it expires.

The Steps involved in the Authentication flow using OpenID Connect are the Web application that needs to authenticate user, it must redirect user to or authorize endpoint then they authorize returns token, it redirects to the Reply URL configured in Azure AD and then finally Web server validates token, start user session and set session cookie.

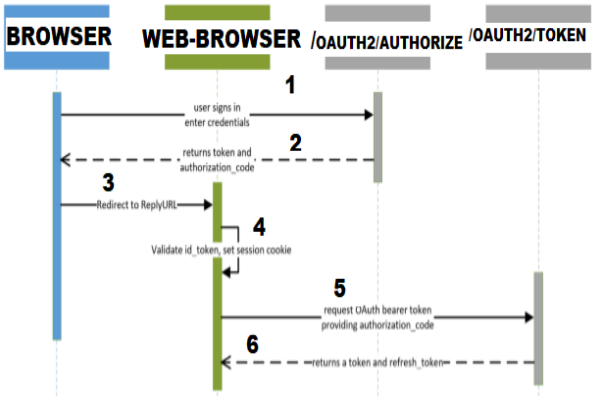

Figure 2: Authentication flow using OpenID Connect

The Authentication flow using OpenID connect has the following steps involved as shown in the above sequence diagram.

1. user signs in by entering credentials
2. return token and authorization-code
3. Redirect to ReplyURL
4. Validate id_token, set session cookie
5. request OAuth bearer token providing authorization _code
6. returns a token and refresh token

IV. THREATS – ATTACKS AND SECURITY MEASURES

The attackers could try to get access to the data of a client in order to obtain clients secrets, accesses the authorization codes. The attacker even tries to access the user login and tries to get the confidential data of the cloud user. Eavesdropping or Leaking of codes where data at transit or at rest is another major problem in the cloud computing.

S. No	Risks and Threats
1	End-User Credentials Phished Using Compromised or Embedded Browser
2	Password Phishing by Counterfeit Authorization Server
3	Eavesdropping / Obtaining Access Tokens from Authorization Server Database
4	Obtaining Client Credentials during Transmission
5	Malicious Client Obtains Existing Authorization by Fraud
6	Obtaining Users Authorized Password Credentials from server

Table 1: Different Risks and Threats on Cloud

In cloud data security is mandatory, authentication is mainly required for secured data sharing, the security considerations are communication between the authorization server and resource server, token formats, the mechanism used by authorization servers to authenticate the user. A redirect URI prevents phishing attacks and detects malicious clients, where authorization code is revealed through redirectors and counterfeit web application clients, authorization server differentiates the public and private clients using the implicit grant type based on registered credentials and authorization request [10].

The Eavesdropping or Leaking Authorization "codes" by an attacker is done between the authorization server and client. As the authorization codes are passed via web browser, the usage of untrusted websites, the different ways the attacker try to get codes are Referrer headers, Request logs, open redirectors and Browsers history. The Referrer header contains the address of the previous webpage and link of the currently requested page, it allows servers to identify from where people are visiting them, it is used for data analytics, logging or optimized caching. Request logs are the log files which maintains automatically by server, which consists of list of activities it performed, these commonly includes the query parameters on requests. Web sites sometime send users to another destination via a redirector, An open Redirection redirects the user to vulnerable website, and directs the user to malicious web-page, to execute further attacks, the open redirectors pose a particular risk to web-based delegation protocols because the redirector can leak verification codes to untrusted destination sites. Phishing is a procedure of fraud where malicious client uses different mechanisms to masquerades as a reputable entity and distributes malicious links or attachments to extract the login credentials from servers. Generally, link manipulation is done where the hidden links will be activated on mouse clicks, so to prevent such phishing messages from reaching end users the layered security controls are used such as anti-virus software, firewalls, Intrusion Detection and Prevention systems, Gateway email filters, web security gateways [11].

The Authentication and Authorization protocols are used for the defense system to protect cloud from different Attacks, Strong Authentication is provided based on the security key authentications, the different threat models or attacks on the cloud environment to be defined by above protocols are

S. No	Attack Model
1	Insider Attacks
2	SQL Injection
3	Cross-Site Scripting
4	Session Hijacking
5	Cookie Tampering
6	Takeover of Remote Web Server
7	DDoS Attack
8	Advanced Persistent Attack

Table 2: Different Attacks on Cloud

An Insider is who has privileges to access cloud database, uses credentials for data theft or data destruction in cloud. The firewalls can be deployed to stop outsiders, but firewall cannot restrict insider attacks. Based on the levels of privileges to access cloud database, the insiders are broadly classified as pure insiders, insider associate, insider affiliate and outside affiliate.

A Secure Protocols based on Authentication and Authorization protocols are used for cloud data security [12]. SQL Injection attacker uses different commands to break the verification security credentials and log into the cloud environment, and even links to the SQL Databases, Authentication is the major security credential provided to the cloud and maintained as per the Global Standards, using Azure Active Directory, It protects the credentials of the employees in a company, once the authentication is successful, the employees can access to the specific company applications [13].

Cross-Site Scripting models are classified in to three, they are Persistent or Stored Cross-Site Scripting, Non-Persistent or Reflected Cross-Site Scripting and finally DOM based Script. The Persistent XSS is stored malicious cross scripting, which is embedded by the different websites, where attacker embedded different malicious scripts and stored in the server. In reflected XSS the attacker builds malicious URL and sends the link via Email. Document Object Model is used to attack on web applications [14]. Session Hijacking takes place to get an unauthorized access to the data and services in the cloud, it is a Security Attack using TCP Session, using IP spoofing on the protected network. Cookies are used when user logs in to the website or cloud environment, if the user credentials are accurate then user is authenticated to use the cloud, Cookies manipulation on browser side and is command extensive, allows you to list, delete, edit and add cookies arbitrarily. Distributed Denial of Service Attack sends floods of messages which makes the targeted server to slow down or shut down. Advanced Persistent Threats are Targeted Attacks are sophisticated and stealthy continuous process and uses zero-day application and back door exploits. A sophisticated defense system must be designed to over-come threats and attacks and implementation of a secured system to protect confidential data for cloud security [15].

V. EXPERIMENTAL ANALYSIS

A. Architectural Design of Azure AD Authentication

Azure AD Authentication is a cloud security-based service for identity management used by the cloud service provider. It's a library from .NET which helps developers to authenticate users and have out of box features like async support, token cache & refresh tokens. Much of the implementation is abstracted and the developer can focus on logic and securing resources while maintaining high security. Authentication is classified into two categories the first one is Azure AD Authentication where Authentication will happen against Global (GTS) maintained Azure Active Directory for all employees of a company. The Second one is Application Authentication, where once the authentication is successful, not all the employees of the company will have the authentication, to specific company application. Though the company user got authenticated against Global AAD but there is a need to authenticate whether user have access to specific application or not and this can be achieved at application level, though the company user got authenticated against Global AAD but there is need to authenticate whether user have access to specific application or not and this can be achieved in application level authentication. To ensure that the token size does not exceeds HTTP header size limits, Azure AD limits the number of object-Ids that it includes in the groups claim. If a user is member of more groups than the average limit such as 150 for SAML tokens, 200 for JWT tokens, then Azure AD does not emit the groups claim in the token. Instead, it includes an average claim in the token that indicates to the application to query the Graph API to retrieve the user's group membership [16].

B. Azure Active Domain Services and Azure AD for Authentication and Authorization

Users and Groups are created in on-premises Active Directory, where it will get syncronized with Global Azure Active Directory with the help of Azure Active Directory Connect. Group Membership Management will be in contact with the Active Directory on-Premises to get access where Azure AD Connect to users and group syncronization and tokens contains group claims which use graph API once exceeds limit. The major responsibility in this process is done by Azure AD Group Access Management [17].

The Identity management for Cloud Security using Azure AD Authentication is categorized into three identity models they are cloud identity, synchronized identity, federated identity and finally Hybrid identity. The cloud identity provides the cloud-based backend security service for authentication such as Azure AD for identity and authentication service. These are used for authentication, licensing and authorization. In the Hybrid Identity the user accounts are stored both on premises and on cloud where it uses the Active Directory Domain Services (AD DS) and Azure AD, where AD DS is the original authenticated source for user accounts, where azure AD is the synchronized set for user accounts. essentially a copy of accounts in AD DS is preserved for counter verification. AD DS provides access to the cloud-based software's, where Azure AD connect runs on premises server and synchronizes the account from ADDS to Azure AD, those user account will only flow between both, where the authentication is of two types basically managed authentication and federated authentication, with in the management authentication there are two different methods of authentication they are Password hash synchronization (PHS), where AD DS synchronizes with Azure AD and collects the Hashed Passwords and the second type of authentication is Pass-Through Authentication (PTA) where it synchronizes the accounts without hash passwords for PTA, federated authentication do not use hash passwords, rather Azure AD redirects you to Federated Authentication Infrastructure to perform authentication [18].

C. Steps Involved to Implement the Approach

The different steps involved in this Approach to secure the Cloud are

1. On-premise Active Directory
2. Create web app in Azure Service Plan
3. Register web app with Azure Active Directory
4. Implement code for Azure Authentication

5. Implement code for Azure AD Graph API
6. Implement code for Application Authentication
7. Implement code for Authorization

AD DS is used to authenticate identities associated with users within the security boundary. Directory and identity services are typically hosted on-premises, implementing directory and identity services in Azure. On-Premise Network consists of Domain Controller and Azure Active Directory Connect Synchronization, where Group Membership Management is done, On-Premises Client and Request from External user are the other objects that connected to the Azure Active Directory Tenant. Through the Application Subnet they are connected to the Virtual Network which consists of Management Subnet and three tier architecture of Web Tier, Business Tier and Data Tier. The major tasks at this stage are creating users, creating groups, assign user to the groups, synchronization of on-premise Active Directory with Azure AD through Azure AD Connect and finally provides a Group object ID to the Development Team.

Creating Web App in Azure Service Plan is possible with three different languages such as Microsoft.Net framework, Java based PowerShell framework and finally by deployment using an Express Node.js web application and deploy to Azure using the command line [19].

Registering the Web App with Azure AD is another step where we concentrate on Application and Reply URL and to register the web app with Azure AD and can share the following client details such as ClientID, ClientSecretID, Tenant, TenantID, AadInstance, PostLogoutRedirectUri and GraphResourceId.

To Implement Azure Authentication Packages used are

Microsoft.AspNetCore.Authentication
Microsoft.AspNetCore.Authentication.Cookies
Microsoft.AspNetCore.Authentication.OpenIdConnect

Finally, snippets for Code implementation discussed in next section.

VI. CODE IMPLEMENTATION

Implementation of Azure Authentication where we must consider ASP.NET Core – Azure AD – Web App – OpenId Connect and method configure services are handled. The Start.cs program or clause is used, as the name indicates that the Application calls of ASP.Net Code for configuring the methods and services, As the name suggests it is implemented first, when the application starts, The `UseStartup<T>()` method is used first at the configurating the host first in the main() method of the program class, where we create a host for the web application. first, The time span can be seconds, minutes, hours or days that can be used to manage the user logged session by using configure services and when you use the configure that is `app.useAuthentication()` and as options where the timespan is given by the code

```
options.idletimeout=Timespan.fromhours(X);
Services.configureApplicationCookie
(options=>
{options.expireTimespan=Timespan.fromhours(x);
```

The next step here is Exception Handling, which is the most important feature of the Azure AD Authentication, by default ASP.NET code returns a simple status code for any exception that occurs in an application and configure method throws an error, the exceptions should be handled and user friendly messages should be displayed [20].

The Controller Class is used to authorize the attributes in order to restrict controller accessibility, where as controller level enforce the user requirements and action level enforce the admins requirement. Controller class contains action methods which comes under the categories of public methods, controller and its action methods handles incoming browser requests, retrieves necessary model data and returns appropriate responses.

Azure AD Authentication Library relies on its token cache for efficient token management, Access tokens enable clients to securely call applications that are protected by Azure. OAuth enables a secured authorize access to web application with Azure Active Directory Tenant, it even performs the authentication and authorization mechanisms for security in web apps. OpenID connect is a layer built on top of the OAuth protocol, it is used to obtain and use access tokens to access protected resources, it implements authentication as an extension to the OAuth authorization process. It provides information about the user in the form of id token and verifies the credentials based on existing data [21].

```
public class DistributedTokenCache: TokenCache
  {
private static readonly object Filelock = new object ();

    public DistributedTokenCache(string userId,
    ISession session)
     {
  }
  }
```

Azure AD Authentication Library manages the cache structure as a private implementation detail with the help of TokenCache class which stores a directory of tokens, indexed by issuer, resource, ClientID and user. Tokens are sensitive data that grant access to the user resources, where custom cache class is derived from the TokenCache.

Implement code for Azure AD Graph API is the next task where the Nuget Packages are used

```
Microsoft.Azure.ActiveDirectory.GraphClient
Microsoft.IdentiyModel.Clients.ActiveDirectory
```

To fetch an instance of Active Directory Client for communication Azure AD is used to acquire tokens for application with the help of Graph API, the next step is to implement code for the custom application access and for the comparison of user groups and application groups and determines whether user have access to the application or not. An Application uses Azure AD Graph API to perform create, read, update and delete operations on directory data and objects. It is used to perform major operations such as to create a new user in directory or to disable the user or even to delete account or user entirely. It is used to get detailed properties such as their groups, it can update the user's properties [22].

Implementation code for Application Authentication is used for comparison of user groups and application groups and determines whether the user have access to the application. The steps involved in accessing application is as follows

1. Application Setting
2. ADGroupRetrieval
3. ADGroupConfigFactory

4. ADGroupConfig
5. PharmaBrossardAuthenticationAttribute

Implement code for custom authorization according to the requirement, there are different filters used for execution, such as Authorization filters, Action filters, result filters and exception filters. A user request is routed to the appropriate controller and action method, based on the logic these methods are executed and passes through the filters. Authorization stands for verifying the permissions, it is a process of defining and allotting specific roles to specific users [23].

VII. RESULT ANALYSIS

The traditional model is designed for cloud security using Microsoft Azure Active Directory, OAuth, OpenID connect, Redirect URI. We concentrated on cloud security and build a secured model based on the parameters of Confidentiality, Authorization, Accounting, Authentication, Auditing and Privacy which can defend against different risks, threats and attacks. We implemented security consideration to protect from risks, threats and attacks. The implementation is given along with the steps involved in securing the cloud. Administrators should apply the appropriate updates and should allow only trusted users to have network access, security tools can be identified as common security misconfigurations and missing security updates on system endpoints. administrators should able to monitor the cloud security by means of different system tools. Administrator's should follow the standardized security directions and guidelines for security monitoring and enforcement which improve the efficiency of cloud security levels, for extra security on-premises domain controllers are used. To improve the security levels Azure Active Directory can organize the pass-through authentication which permits cloud users to sign-in to together on premises and Artificial Intelligence cloud-based applications by using authentication. Pass through authentication users signs in through password hash synchronization by validating and verifying their passwords straight in contradiction of on-premise active directory.

VIII. CONCLUSION

The proposed solution enables applications, running on secured cloud environments to use authentication and authorization which brings:

– Indication and verification that the user is authenticated

– Indication and verification that the authorization to the user groups according to the requirements.

REFERENCES

1. M. Yingkai and C. Jia, "A Kind of Identity Authentication under Cloud Computing Environment", *2014 7th International Conference on Intelligent Computation Technology and Automation*, Changsha, 2014, pp. 12-15.
2. Yang Jingbo and Shen Pingping, "A secure strong password authentication protocol," *2010 2nd International Conference on Software Technology and Engineering*, San Juan, PR, 2010, pp. V2-355-V2-357.
3. K. Liu and K. Xu, "OAuth Based Authentication and Authorization in Open Telco API", *2012 International Conference on Computer Science and Electronics Engineering*, Hangzhou, 2012, pp. 176-179.
4. R. Weingärtner and C. M. Westphall, "A Design Towards Personally Identifiable Information Control and Awareness in OpenID Connect Identity Providers," *2017 IEEE International Conference on Computer and Information Technology (CIT)*, Helsinki, 2017, pp. 37-46.
5. A. Binduf, H. O. Alamoudi, H. Balahmar, S. Alshamrani, H. Al-Omar and N. Nagy, "Active Directory and Related Aspects of Security," *2018 21st Saudi Computer Society National Computer Conference (NCC)*, Riyadh, 2018, pp. 4474-4479.
6. M. Darwish and A. Ouda, "Evaluation of an OAuth 2.0 protocol implementation for web server applications", *2015 International Conference and Workshop on Computing and Communication (IEMCON)*, Vancouver, BC, 2015, pp. 1-4.
7. G. C. Batista, C. C. Miers, G. P. Koslovski, M. A. Pillon, N. M. Gonzalez and M. A. Simplicio, "Using Externals IdPs on OpenStack: A Security Analysis of OpenID Connect, Facebook Connect, and OpenStack Authentication," *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, Krakow, 2018, pp. 920-927.
8. Dandan Zhang, Hongqi Zhang and Xuehui Du, "An authorization model for Multi-classification Interconnected System," *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, Taiyuan, 2010, pp. V11-685-V11-688.
9. F. Ghaffari, H. Gharaee and M. R. Forouzandehdoust, "Security considerations and requirements for Cloud computing," *2016 8th International Symposium on Telecommunications (IST)*, Tehran, 2016, pp. 105-110.
10. D. R. Bharadwaj, A. Bhattacharya and M. Chakkaravarthy, "Cloud Threat Defense – A Threat Protection and Security Compliance Solution," *2018 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, Bangalore, India, 2018, pp. 95-99.
11. J. V. Chandra, N. Challa and S. K. Pasupuleti, "A practical approach to E-mail spam filters to protect data from advanced persistent threat," *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, Nagercoil, 2016, pp. 1-5.
12. J. V. Chandra, N. Challa and S. K. Pasupuleti, "Advanced Persistent Threat defense system using self-destructive mechanism for Cloud Security," *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, Coimbatore, 2016, pp. 7-11.
13. N. Singh, M. Dayal, R. S. Raw and S. Kumar, "SQL injection: Types, methodology, attack queries and prevention", *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, 2016, pp. 2872-2876.
14. M. Dayal Ambedkar, N. S. Ambedkar and R. S. Raw, "A comprehensive inspection of cross site scripting attack", *2016 International Conference on Computing, Communication and Automation (ICCCA)*, Noida, 2016.
15. P. P. Nikam and R. S. Suryawanshi, "Developing and deploying applications for highly available storage of cloud service through secured channels", *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, Pune, 2016, pp. 93-98.
16. A. Binduf, H. O. Alamoudi, H. Balahmar, S. Alshamrani, H. Al-Omar and N. Nagy, "Active Directory and Related Aspects of Security," *2018 21st Saudi Computer Society National Computer Conference (NCC)*, Riyadh, 2018, pp. 4474-4479.
17. B. Mucahit, "Cloud computing and management processes", *2015 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Bucharest, 2015, pp. S-1-S-4.
18. L. Dostalek and J. Safarik, "Strong password authentication with AKA authentication mechanism," *2017 International Conference on Applied Electronics (AE)*, Pilsen, 2017, pp. 1-6.
19. D. Agarwal and S. K. Prasad, "AzureBench: Benchmarking the Storage Services of the Azure Cloud Platform," *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, Shanghai, 2012, pp. 1048-1057.
20. B. Gao, S. Wang, L. Kang, X. Shu and X. Yang, "Diagnosis and Handling of Exception in Cloud Manufacturing," *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*, Chongqing, 2018, pp. 866-870.

21. S. Chung, J. H. Kim and Y. Kim, "Pragmatic approach using OAuth mechanism for IoT device authorization in cloud," *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Greater Noida (UP), India, 2018, pp. 1-4.
22. M. Grothe, C. Mainka, P. Rösler, J. Jupke, J. Kaiser and J. Schwenk, "Your Cloud in My Company: Modern Rights Management Services Revisited," *2016 11th International Conference on Availability, Reliability and Security (ARES)*, Salzburg, 2016, pp. 217-222.
23. S. Deepika and P. Pandiaraja, "Ensuring CIA triad for user data using collaborative filtering mechanism," *2013 International Conference on Information Communication and Embedded Systems (ICICES)*, Chennai, 2013, pp. 925-928.

AUTHORS PROFILE

J. Vijaya Chandra is a Research Scholar at KLEF (Deemed to be University); Koneru Lakshmaiah Education Foundation. Research areas are Cloud Security, Network Security, Intelligence Security and Data Security. Published 10 Research Papers for International Journals. He is Oracle Certified Associate and Member of IEEE and ACM.

Dr. Narasimham Challa, Professor of Computer Science Engineering & Dean IQAC and Former Principal of Vignan's Institute of Information Technology have been working in the field of Teaching and Research for the last 24 years. The Author received Ph D in Computer Science in the year 2009. Guided three Ph D scholars and guiding four research scholars. Published 79 research articles in various National and International journals. At present, registered for Post-Doctoral work leading to D. Sc with UoS Panama and doing work in the area of verifiable encryption, cryptography and security.

Dr. Sai Kiran Pasupuleti, Ph.D., is Professor, Dept. of Computer Science and Engineering, KLEF (Deemed to be Univesity), Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur District, A.P., INDIA. He is having rich teaching and Research Experience. His research areas are Mobile Computing, Cloud Computing and Computer Networks. He published about 30 Research Papers in International Journals.