API Vulnerabilities In Cloud Computing Platform: Attack And Detection

Muhammad Azizi Mohd Ariffin, Mohd Faisal Ibrahim, Zolidah Kasiran

Fakulti Sains Komputer dan Matematik, Universiti Teknologi MARA, Shah Alam, Malaysia

ABSTRACT

Nowadays most of cloud management software such as OpenStack and CloudStack provide an API to facilitate the communication and exchange of data between users, application, cloud components and infrastructure. Due to complexity of cloud management software implementation, the provided API has vulnerabilities which can be exploited by malicious party. Once exploited, it can cause security issue and disrupt the availability of services running on the cloud infrastructure. Hence; it is importance to address cloud API security by identifying potential threats, demonstrating how such threats could be exploited and how to detect such threat. This paper presents the topic of API Vulnerabilities in Cloud Computing Platform: Attack and Detection. We will discuss the vulnerabilities of the API in cloud management software. Based on these vulnerabilities, this paper will demonstrate how eavesdropping on cloud API authentication services and API exhaustion attack can be initiated. To address the threat due to the vulnerabilities of the API, we need to detect attack which exploits the vulnerabilities. Thus this paper also proposes and demonstrates methods to detect such attack effectively. Method to detect ongoing API exhaustion attack will be based on AD3 algorithm. From the experiments result, it shows that attacks on the cloud platform AP can be detected effectively.

Keywords: *Cloud Computing Security, Cloud API Vulnerabilities, API Exhaustion Attack.*

I. INTRODUCTION

The evolution of cloud computing technology has introduced several advantages such as elasticity, scalability and more efficient use of computing resources. This caused it to be commonly used by education sectors, government, private enterprise and IT industry to replace their legacy-computing platform in their organization[1] [2]. However, the popularity of the platform has attract hackers to exploit it weaknesses and made it vulnerable to various security threats [2]. In private deployment, cloud-computing platform consists of hardware and software components pooling and sharing its resources together, usually all of these cloud components communicate with each other via common interface called as Application programming interface (API). This API can be overwhelmed with large amount of request intentionally by malicious party until it exhausted all of its computing resources and thus causing denial of service. When the API is overwhelm, the cloud components unable to communicate with each other and the cloud platform itself unable to perform its normal operation. This will cause outages to the platform and will affect the availability of applications running or hosted on the cloud platform.

Several studies have been conducted to identify the security threats to the cloud computing platform [4] [5] [6] [3]. But those studies did not focus on cloud API exhaustion attack and provide a method to detect and mitigate such attack. Most of the studies, discuss the threat of data breach, weak authentication and virtualization vulnerabilities on the cloud. Work in [5] mentioned the insecurity of the cloud API that may lead to security issue but did not discuss further on how the cloud API could be exploited further to cause denial of service attack to the cloud platform components. The impact to service availability of cloud platform has huge repercussion, for example recent outage on Microsoft Azure platform, causing thousands of users unable to access the Office 365 application and run their business normally [7]. Therefore, there is a need to address the threat of API exhaustion attack to the cloud-computing platform. Moreover, when storing data in networked environment such as in Cloud, it was crucial to address the vulnerabilities so that it able to pass the threat assessment such as MyRAM and HiLRA [27].

This paper discusses the vulnerabilities of the API in cloud management software. Based on these vulnerabilities, this paper will demonstrate how eavesdropping on cloud API authentication services and API exhaustion attack can be initiated. To address the threat due to the vulnerabilities of the API, we need to detect on-going attack which exploits the vulnerabilities. This paper will also demonstrate how such attack can be detected effectively using AD3 algorithm. OpenStack software platform will be used to construct the cloud testbed which will be used for the demonstration of attacks. This will provide more accurate representation of real world cloud deployment

as OpenStack is commonly deployed as cloud platform in production environment [8].

II. API VULNERABILITIES IN CLOUD COMPUTING PLATFORM

A. Cloud Management Software

Building and operating own private cloud platform is a complex and challenging job. Hence, cloud management software was regularly utilized by organization for simplifying the task of managing their cloud. The definition of cloud computing management software is "Software and technologies utilized by public or enterprise organization to build and operate in-house or on-premise cloud platform and infrastructure, the tool in cloud management guaranteed resources of cloud computing are utilized or distributed effectively and able to appropriately interacting external services and users via common interface" [9]. The cloud management software has two deployment categories; it can be deployed using open source software such as OpenStack, Eucalyptus and CloudStack or using closed-source solutions such as Citrix XenServer and VMware vCenter

One of the core functions of the cloud computing software is to manage and orchestrate the operation of physical hardware which pools it resources for the cloud platform [10].

It guarantees the physical resources are assigned optimally to virtual machine and provides a common interface (e.g. API) which enables users, infrastructure and services to communicate. For example, in OpenStack it has dedicated software component or module which dedicated in managing the allocation of computing resources (e.g. CPU, Memory) to the virtual machine. Besides that, OpenStack also has others components which dedicate in providing other service such as Networking via its Neutron and block storage via Cinder. Usually all of these different cloud components communicate with each other via common interface known as API.

B. Type of Cloud API

Cloud APIs are the common software interface which enables the communication between the cloud infrastructure, applications and users. There are several types of API have been widely deployed in cloud platform such as SOAP and REST based API [24]. SOAP (Simple Object Access Protocol) is a protocol that define many things such as how data is transmitted and security mechanism which introduce more overhead. While REST stands for Representational State Transfer and it is simpler than SOAP. REST is a web services as it uses HTTP protocol and URIs. Table 1 shows the comparison of SOAP and REST API.

Table 1: SOAP and REST API Comparison

Comparison	SOAP	REST
Data Format	XML	JSON, Plain Text, HTML
Style	Protocol	Server-Client Architecture
Security	Support SSL, WS-Security	Support SSL, HTTPS
ACID Compliance	Yes	No

In cloud management software such as OpenStack, the API is based on REST and uses the same HTTP protocol as other web based system. Disruption on the API availability of OpenStack will cause disruption on the normal operation of the plat-form.

C. Type of API Attack

API plays a crucial role in cloud computing infrastructure communication as it allows different users and cloud components to interact and exchange data. Therefore, attacker could exploit the weaknesses of cloud management software such as Open-Stack [11] and its API implementation for malicious intent.

The first type of attack is API Authentication Services Attack. This type of attack can be initiated by exploiting the weaknesses of cloud API which provides authentication services in cloud infrastructure. In cloud management software such as OpenStack or CloudStack, API was provided for interacting with the authentication services. Communication between hosts and the authentication is sensitive as usually credentials data such as password and session token is been exchange during the session.

Majority of API in the cloud management software is based on REST or SOAP which is web standards [26]. Hence it is exposed to attacks which are web-based, such as eavesdropping, session hijacking, malicious code execution, XSS and denial of service attack [14] (see figure 1). One of crucial services in OpenStack is the API which handle authentication, which a module known as Keystone. The work of [15] has identified that the API of Keystone is also exposed to eavesdropping attacks, because during the procedure to authenticate users, the credentials data are communicated in plain-text. Besides, authentication mechanism based on token exchange in Keystone also has weaknesses. This is because, hackers will able to gain users privileges and access the services of other cloud components if they able to obtain the password contained in the authentication token.

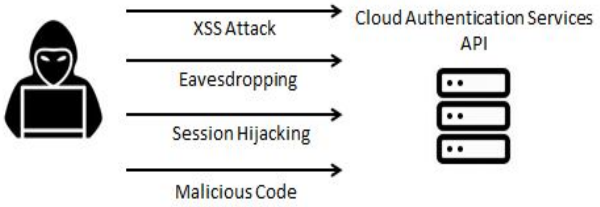

Figure 1. Multiple Type of Attack to the Authentication Service API

The second type of attack is API Exhaustion Attack. This is a type of DOS attack to the cloud API services. Denial of service (DOS) attack occurs when attacker disrupts the services by intentionally sending large amount of traffic with the purpose of overwhelming the system. This prevents the system from processing the request of legitimate users and thus denying them from using the service. In the context of cloud computing, the DOS attack can be targeted to the applications running on the cloud or it can target the cloud platform infrastructure [13] [12].

When the DOS attack is targeting the cloud platform API, it can cause API exhaustion attack. Majority of cloud management software offer web-based API protocol, for compatibility and simplicity. For instance, CloudStack and OpenStack APIs are built on REST, and during the communication session the data is formatted as JSON [16] [17]. The work of [15] have found that the OpenStack Keystone API which uses web based protocols to provide the identity and authentication service is vulnerable to information disclosure, DOS and replay attacks.

API exhaustion attack is when the attackers maliciously misuse the API of the cloud platform by sending a large quantity of malicious API requests to overwhelm the system. The cloud components will be unable to respond to legitimate API request from other components and users while it is overwhelm. This is because the web (HTTP) protocols utilized TCP as the transport protocol, thus when the server receives API requests using HTTP; it will allocate additional resources for the new TCP session. The physical hosts of the cloud management system components will eventually run out of resource if this continues for a long period. Hence, it unable to process the legitimate API request, which will lead to a DOS attack and disruption to its availability. The cloud management software are exposed to these type of attack since it uses web-based technology in API services and this issue has been highlighted by many cloud admin on portal to track bugs and vulnerabilities database [18] [19]. This paper highlights the API exhaustion because the communication using the common TCP/IP stack has become the source of weaknesses of a cloud management system. Figure 2 shows API exhaustion attack overview.

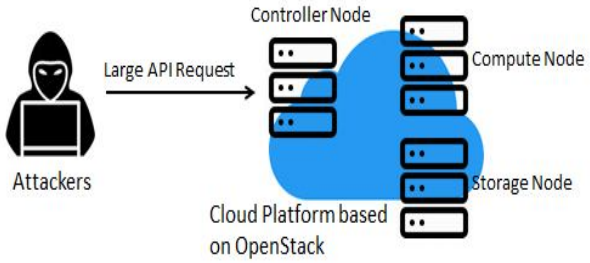

Figure 2. API Exhaustion Attack

III. ATTACK DETECTION EXPERIMENT ON CLOUD API SERVICES

This paper, will conduct experiments to simulate authentication token eavesdropping and API exhaustion attack. During the experiment, we will apply a method to detect such attack. To simulate an attack, we setup a cloud Testbed based on OpenStack platform.

For the purpose of emulating a cloud provider setup, we configured a multiple node cloud cluster with two compute and one controller nodes. 3 Servers hosts running a Dell OptiPlex 990 (3.40 GHz Intel i7-2600), 1TB of Hard Disk, 16 GB memory, 2 Gigabit-Ethernet network interfaces were used to form a cluster. One of the Ethernet inter-faces is connected to the management network and the other is connected to the data network. The management network is primarily used for communication between cloud nodes (e.g. communication between controller and compute node, transferring data between VM during migration); it can also be used to access the cloud system panel by the cloud admin. Meanwhile, the data network enables VM hosted on the cloud infrastructure to interact between each other and enable the VM to access the Internet. The topology of cloud cluster testbed is shown in figure 3.

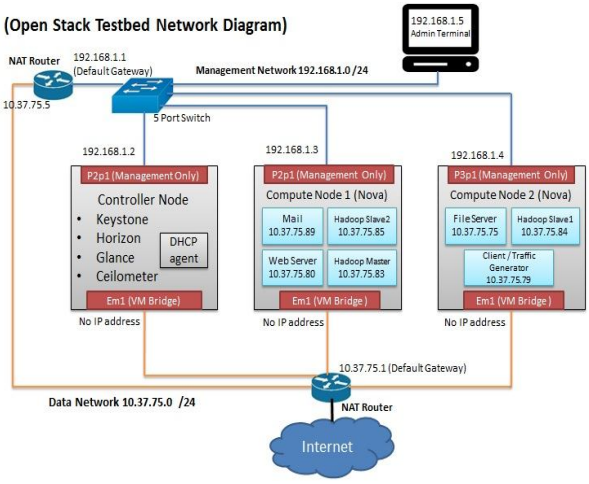

Figure 3. OpenStack Cloud Testbed

A. Authentication Token Eavesdropping Experiment

To investigate the security issues on the authentication services API, this paper recreates the issue on the testbed. First we simulate authentication request via OpenStack API and then we eavesdrop the session by capturing the traffic between client and OpenStack controller using Wireshark tool. From the packet dump, it was shown that the token packet for the API services was not encrypted. Figure 4 shows that we can read the API data in clear-text and as a consequence we able to capture the password of the cloud administrator as highlighted in the red box. It shows that we able to exposed the password by capturing the packet of the authentication API when the cloud admin request a token from the OpenStack Keystone.

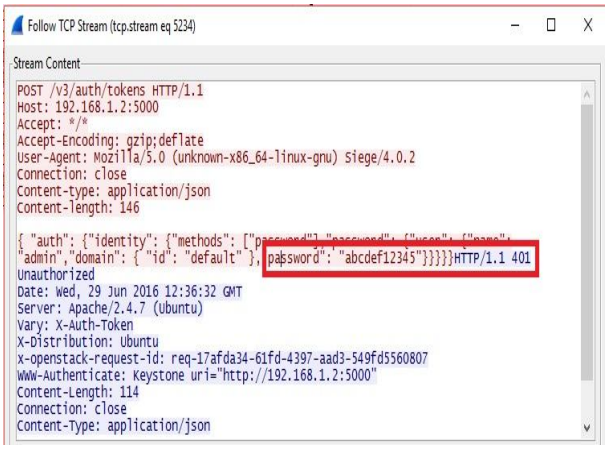

Figure 4. The API data exposed in plaintext during eavesdropping

B. API Exhaustion Experiment

To simulate API Exhaustion attack to the OpenStack API, this paper utilized open source software called Siege [20]. The software will simulate 15 hosts concurrently sending a huge numbers of requests to the API services using JSON format. Furthermore, the background traffic will be running while the experiment is running for 10 minutes. While the attack is running during the experiment, this paper will apply the AD3 algorithm in order to detect such attack. To ensure the detection based on proposed methodology able to distinguish between the anomaly produced by an attack and normal operation of API during experiment, we execute two normal VM operations. The First operation is restarting a VM and the second operation is deleting a VM. Both of the operations will invoke the OpenStack to utilize the keystone API to invoke the authorization and authentication process.

C. API Exhaustion Attack Detection Method

In order to effectively minimize the security risk in the cloud environment, we need to detect anomaly that possibly cause by the malicious activity [21]. Inspired by the work of [22] regarding a method applied to the cloud for malware detection system, this paper proposed an anomaly detection method based on AD3 algorithm [29] and non-parametric data density [23] which designed for detecting anomaly in the cloud platform. The anomaly detection methodology is shown in figure 5.

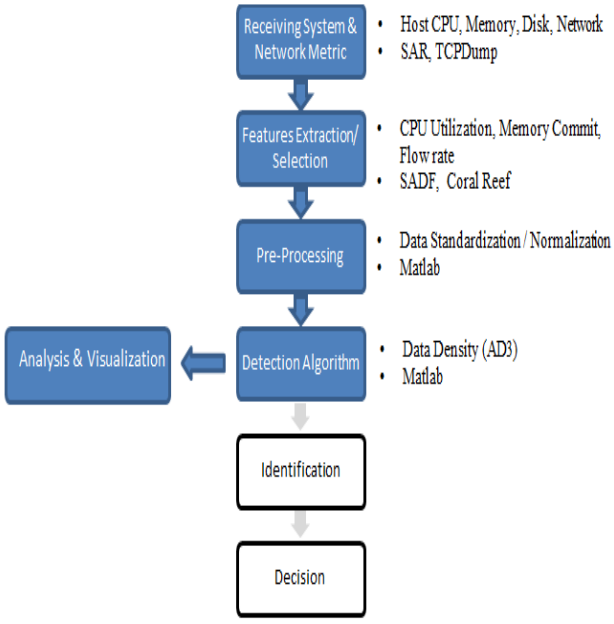

Figure 5. Anomaly Detection Methodology [25]

During experiment, we collected several parameters such as CPU or memory utilization from the testbed and then fed it into the anomaly detection algorithm. The algorithm will then extracts features from the parameters and perform a pre-processing on those features. The algorithm and pre-processing was performed using Matlab Software. Next, it will detect an outlier data in a form of graph, which is then, can be used for further analysis and visualization.

AD3 [29,30] is a machine learning algorithm for approximate maximum a posteriori (MAP). In this project, the anomaly parameters is extracted and then normalized to be become data density value [23]. From there, the density value is further processes using AD3 so multiplier value can be obtained. As a result we can differentiate the noise of normal VM operation and value (in the form of data density) of an anomaly cause by an attack.

D. API Exhaustion Attack Detection Result

We simulate the attack on the cloud APIs and collect the system parameters during the attack. The parameters are then fed into the AD3 algorithm with the purpose to detect and visualized the anomalies due to the attack. The parameters which are collected during experiment and then feed into the detection algorithm are system CPU utilization, memory utilization, byte count, flow rate, interface TX and RX rate and the packet count. The experimental result of the API exhaustion attack is shown in Figure 6.

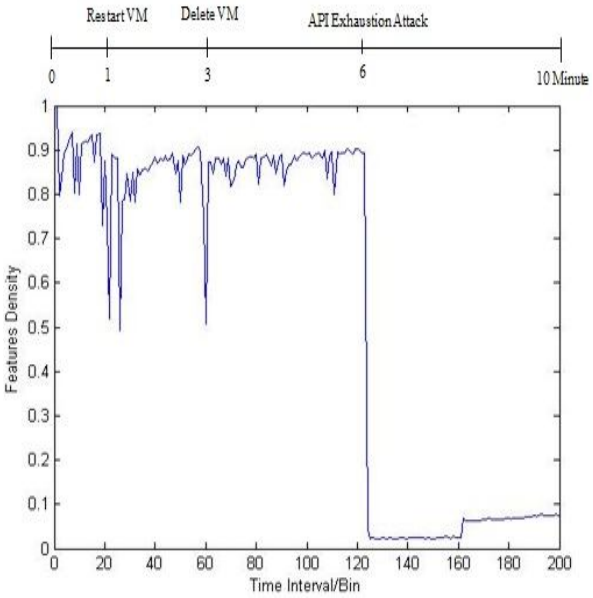

Figure 6. API Exhaustion attack detection result

In the simulated attack, huge quantities of API requests were directed to the Open-Stack authentication service API to overwhelm it. The X-axis signifies the time interval at which the metric data was gathered while the Y-axis show the calculated density values for the selected features. Figure 6 shows, the algorithm able to detect 4 anomalies as there is 4 distinct variation which shown as drops on the graph. At time interval 20, the first anomaly is detected which during the period the system initiate a VM's reboot operation. During first anomaly, the density value drops from 0.9 to 0.52 (0.38 differences). The graph line starts to recover (return upward) at time interval 21. Shortly after, at time interval 24 the second anomaly is detected, after the first anomaly. During the period of second anomaly the density value drops from 0.9 to 0.5 (0.4 differences), the graph shows recovery when time interval is at 25.

The system initiation of reboot operation of VMs triggers the first and also the second anomalies; the first anomaly happens when the VM is being shut down, while the second anomaly is detected when the VM is turned on back again, to complete the whole reboot procedure. When the time interval is at 60, the third anomaly is spotted which during this period the system is initiating the VM deletion process. The density value drops from 0.9 to 0.52 (0.38 differences) during the deletion process. The deletion process completed at time interval 64 and during this period the line on the graph has recovered. At time interval 120, the fourth anomaly is spotted which during the period simulated API exhaustion attack begins, the density value drops from 0.88 to 0.02 (0.86 differences) which shows an extreme drop on the graph. The API exhaustion attack ended at time interval 162 and the line on the graph only begins to recovers after that period. No other operation is initiated after the attack.

IV. DISCUSSION

During the experiment to simulate the eavesdropping of the authentication token, it shown that the communication session between the API requester and the cloud controller is not encrypted, hence we able to read the token packet in plaintext and obtain the cloud administrator password. In any secure system, password privacy is very crucial; this will cause security issue to the cloud platform as it could lead to data breach, privacy issues and session hijacking. Even if the web interface has mechanism to filter input to prevent SQL injection [28], it may not able to address security on cloud API due to eavesdropping if the packet was not encrypted.

Besides that, there is a clear difference between the anomaly caused by normal VM operation and an anomaly trigger during API exhaustion attack from the experiment result to simulate the API exhaustion attack. This is because during the API exhaustion attack an anomaly with the largest drops is seen. This shows that the difference of data density value during the attack is extreme with value of 0.86. Moreover, it also shown the methodology of detection is accurate as it did not count or regards normal operation or background traffic as an anomaly. Hence, for the purpose of detecting anomalies of such attack, we did not require to use a secondary source of contextual information to distinguish or differentiate between anomalies caused by normal operations and an attack.

By putting a threshold value on the feature density value differences, we able to implement an anomaly detection system which automatically flag an anomaly of API exhaustion during on-going attack, as an example, we can set a threshold value to 0.5; hence, if the difference of the density value is more than > 0.5 is

calculated, it will be automatically flagged as malicious and cloud admin will be alerted. Meanwhile, Anomalies cause by normal activity will not be flagged as malicious as most anomalies caused by normal activity have a density difference value of less than < 0.5. This will lower the probability of causing a false alarm and producing an anomaly detection method which is more accurate for the cloud environment.

V. CONCLUSION

As a conclusion, malicious parties can exploit the vulnerabilities of the cloud API in cloud management software such as OpenStack. The attack on API which provides authentication services could lead to data breach and privacy issues while API exhaustion attack could disrupt the availability of the API services. This paper has demonstrated that critical password was exposed in plaintext while performing eavesdropping on the token packet of the authentication API. Moreover, this paper also demonstrated that by sending large amount of API request until the cloud controller node exhaust all of it resources can cause denial of service, thus disrupting the availability of the API services. Such attack can be detected by identifying anomalies on the system parameters using the AD3 algorithm. In the future, we want to explore the novel method to automatically mitigate such attack after it has been detected.

ACKNOWLEDGEMENT

This research was supported by Faculty of Computer and Mathematical Sciences, UiTM Shah Alam which provides us with a grant to conduct this research. We also would like to extend our gratitude to fellow colleague from Computer Technology and Networking department for comment and support needed for this research.

REFERENCES

[1] Weins, K. (2016). "*Cloud Computing Trends: 2016 State of the Cloud Survey*". [online] Rightscale.com. Available at: http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2016-state-cloud-survey.

[2] Nanos, I., Manthou, V. and Androutsou, E. (2018). "*Cloud Computing Adoption Decision in E-government. Operational Research in the Digital Era – ICT Challenges*", pp.125-145.

[3] Subramanian, N. and Jeyaraj, A. (2018). "*Recent security challenges in cloud computing*". Computers & Electrical Engineering, 71, pp.28-42.

[4] Almutairy, N., Al-Shqeerat, K. and Al Hamad, H. (2019). "*A Taxonomy of Virtualization Security Issues in Cloud Computing Environments*". Indian Journal of Science and Technology, 12(3), pp.1-19.

[5] Satya Suryateja, P. (2018). "*Threats and Vulnerabilities of Cloud Computing: A Review*". International Journal Of Computer Sciences And Engineering, 6(3).

[6] Basu, S., Bardhan, A., Gupta, K., Saha, P., Pal, M., Bose, M., Basu, K., Chaudhury, S. and Sarkar, P. (2018). "*Cloud computing security challenges & solutions-A survey*". 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC).

[7] Hay, R. (2018). "*Rethinking the Cloud After Recent Microsoft Azure Outages*." [online] IT Pro. Available at: https://www.itprotoday.com/performance-management/rethinking-cloud-after-recent-microsoft-azure-outages [Accessed 21 Mar. 2019].

[8] Hillsman, M. and Leong Sun, Y. (2018). 2018 "*OpenStack User Survey Report*". [online] OpenStack. Available at: https://www.openstack.org/user-survey/2018-user-survey-report/ [Accessed 21 Mar. 2019].

[9] Ismaeel, S., Miri, A., Chourishi, D. and Dibaj, S. (2015). "*Open Source Cloud Management Platforms: A Review*". 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing.

[10] Kumar, R., Gupta, N., Charu, S. and Jain, K. (2019). "*Open Source Solution for Cloud Computing Platform Using OpenStack*". International Journal of Computer Science and Mobile Computing, 3(5).

[11] Albaroodi, H., Manickam, S. and Singh, P. (2014). "*Critical Review Of Openstack Security: Issues And Weaknesses*". Journal of Computer Science, 10(1), pp.23-33.

[12] Joshi, B., Vijayan, A. and Joshi, B. (2012). "*Securing cloud computing environment against DDoS attacks*". 2012 International Conference on Computer Communication and Informatics.

[13] Ahmed, R., Hussain, M., Rahmani, T. S., Mansoor, A., & Ali, M. L. (2018). "*Minimization of Security Issues in Cloud Computing*". Journal of Information Communication Technologies and Robotic Applications, 3-40. Retrieved from http://jictra.com.pk/index.php/jictra/article/view/48

[14] Somorovsky, J., Heiderich, M., Jensen, M., Schwenk, J., Gruschka, N. and Lo Iacono, L. (2011). "*All your clouds are belong to us*". Proceedings of the 3rd ACM workshop on Cloud computing security workshop - CCSW '11.

[15] Cui, B. and Xi, T. (2015). "*Security Analysis of Openstack Keystone*". 2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing.

[16] Developer.openstack.org. (2016). "*OpenStack Docs: OpenStack APIs*". [online] Available at: http://developer.openstack.org/api-guide/quick-start/api-quick-start.html#openstack-api-quick-guide.

[17] Goasguen, S. (2013). "*Intro to CloudStack API*". [online] Slideshare.net. Available at: http://www.slideshare.net/sebastiengoasguen/intro-to-cloudstack-api.

[18] Heczko, A. (2015). Bug #1509986 "*Security vulnerability: OpenStack APIs and Horizon...*" : Bugs : Fuel for OpenStack. [online] Bugs.launchpad.net. Available at: https://bugs.launchpad.net/fuel/+bug/1509986.

[19] Cve.mitre.org. (2012). CVE -CVE-2013-0247. [online] Available at: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-0247.

[20] Fulmer, J. (2012). Siege Home. [online] Joedog.org. Available at: https://www.joedog.org/siege-home/

[21] Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A. and Rajarajan, M. (2013). "*A survey of intrusion detection techniques in Cloud*". Journal of Network and Computer Applications, 36(1), pp.42-57.

[22] Watson, M., Shirazi, N., Marnerides, A., Mauthe, A. and Hutchison, D. (2014). "*Towards a Distributed, Self-organising Approach to Malware Detection in Cloud Computing*". Self-Organizing Systems, pp.182-185.

[23] Angelov, P. and Yager, R. (2011). "*Simplified fuzzy rule-based systems using non-parametric antecedents and relative data density*". 2011 IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS).

[24] Wodehouse, C. (2019). "*SOAP vs. REST: A Look at Two Different API Styles***"**. [online] Upwork. Available at: https://www.upwork.com/hiring/development/soap-vs-rest-comparing-two-apis/ [Accessed 17 Apr. 2019].

[25] Ariffin, M., Marnerides, A. and Mauthe, A. (2017). "*Multi-level resilience in networked environments: Concepts & principles***"**. 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC).

[26] Ali, M., Zolkipli, M., Zain, J., & Anwar, S. (2018). "*Mobile Cloud Computing with SOAP and REST Web Services***"**. Journal Of Physics: Conference Series, 1018, 012005. doi: 10.1088/1742-6596/1018/1/012005

[27] Mohd Ali, F., & Hadzril Wan Ismail, W. (2011). "*Network security threat assessment model based on fuzzy algorithm***"**. 2011 IEEE International Conference On Computer Science And Automation Engineering. doi: 10.1109/csae.2011.5952688

[28] Abu Othman, N., Mohd Ali, F., & Mohd Noh, M. (2014). "*Secured web application using combination of Query Tokenization and Adaptive Method in preventing SQL Injection Attacks"*. 2014 International Conference On Computer, Communications, And Control Technology (I4CT). doi: 10.1109/i4ct.2014.6914229

[29] Martins, André & Figueiredo, Mário & Aguiar, Pedro & Smith, N.A. & Xing, E.P.. (2015). AD3: *"Alternating directions dual decomposition for map inference in graphical models***"**. Journal of Machine Learning Research. 16. 495-545.

[30] Naveed, Q. N., Qureshi, M. R. N. M., Shaikh, A., Alsayed, A. O., Sanober, S., & Mohiuddin, K. (2019). *"Evaluating and ranking cloud-based e-learning critical success factors (CSFs) using combinatorial approach***."** *IEEE Access*, *7*, 157145-157157.