A Framework to Automate Cloud based Service Attacks Detection and Prevention

P Ravinder Rao[1]
Research Scholar in Koneru Lakshmaiah Education Foundation
Dept. of Computer Science and Engineering-India

Dr. V.Sucharita[2]
Supervisor in Koneru Lakshmaiah Education Foundation
Dept. of Computer Science and Engineering
Narayana Engineering College-India

***Abstract*—With the increasing demand for high availability, scalability and cost minimization, the adaptation of cloud computing is also increasing. By the demand from the data, consumer or the customers of the applications, the service providers or the application owners are migrating all the applications into the cloud. These migrations of the traditional applications and deploying new applications are benefiting the consumers and the service providers. The consumers are getting the higher availability of the applications and in the other hand, the consumers of the applications are getting benefits from of the cost reduction by optimal scalability and deploying additional features with the least cost, which intern providing the better customer satisfaction. Nevertheless, this migrations and new deployments are attracting the attention of the hackers and attackers as well. In the recent past, several attacks are reported on various popular services like search engines, storage services, and critical application ranging from healthcare to defence. The attacks are sometimes limited to the data exploration, where the attackers only consume the data and sometimes the attackers destroy crucial services. The major challenge in detecting these attacks is mostly identifying the nature of the connection request. Also, identifying the attacks are not sufficient in providing the security for the cloud services and must be deployed as security as a service in the applications or the services or in the data centre as automatic and continuous measures. Various research endeavours have shown critical enhancements in the on-going past for recognizing the security attacks. Nonetheless, these attempts have not provided any solution in preventing the security attacks. Also, the existing methods as mentioned are not automated and cannot be included in the services. Thus, this work provides a unique automated framework solution for detecting the application traffic pattern and generates the rule sets for detecting any anomalies in the request types. The major outcome of this work is to identify the attack types and prevent further damages to the cloud services with a minimal computational load. The additional benefits from this work are the preventive measure for popular attack types. The work also demonstrates the ability to detect a new type of attacks based on traffic pattern analysis and provides preventive measures for making the cloud computing application hosting industry a safer place.**

Keywords*—Data breach; HoA; insider threat; malware injection; ACS; insecure APIs; DoS; automated attack detection; automated prevention; characteristics based detection

I. INTRODUCTION

The remote attacks in the cloud computing environments are generally carried out by executing malicious commands through the connection requests to the virtual machines of the cloud services. The work by Z. Su et al. [1] has demonstrated the effects of the attacks and damage situations on the services. As also demonstrated by the A. Stasinopoulos et al. [2], the attackers can deploy powerful commands to permanently damage the authentication protocols and can obtain access to any of the cloud services. The attacks are not limited to the single applications. Any attacks on the data centre authentication, such as the SSH key based authentications, can generate access viability to all the applications hosted on that datacentre. The analysis report from AWS, Analysis of SSH Attacks on Amazon EC2 [3], is a significant proof of the collateral damage.

The best possible way of preventing these attacks on the security protocols are making the network architecture virtual and continuously changing. Also, the pattern of the connection requests must be analysed in order to make an early prediction of the possible attacks. The pattern of the connection requests must be also analysed against the application type for stopping the algorithm making false detection of the attacks.

In this direction of research, a number of research outcomes are presented by various researchers. The outcome from G. Badishi et al. [4] has demonstrated the strategy for detecting DoS attacks on the cloud networks and the preventive measure. The enhancements of the previously reported work are again enhanced by Q. Jia et al. [5] in the year of 2013. Regardless to mention the works of W. G. Morein et al. [6] and A. Stavrou et al. [7] also must be considered as popular solutions to the DoS attacks on cloud services. Nevertheless, these outcomes are majorly focused on the DoS attacks and do not address other types of attacks.

Thus the demand from the research and application industry on cloud computing is to provide a generic solution for detection and prevention of all major types of attacks on cloud and also build the capability to detect newer types of attacks. Henceforth, this work objectifies these challenges as deliverable outcomes.

The rest of the work is elaborated as, in the Section–II the detailed review of the literature is carried out with the limitations, in the Section–III the analysis of the attack characteristics are performed, further, the deployment of the security measure as preventive actions are elaborated in the Section–IV, Section–V discusses about the automatic detection and prevention framework components with details, the driving algorithm of this work is elaborated in the Section–VI, the comparative analysis is carried out in the Section–VII, the obtained results are discussed in the Section–VIII and the final conclusion of this work is presented in the Section–IX.

II. Outcomes from the Parallel Research Works

The attacks on cloud services, networks, resources and infrastructure are not recent. A number of attacks are reported every year violating the security policies, destroying the resources and making application data visible over the networks. However, the number of attacks has increased in the recent years. As a counter measure the number of researches is also carried out in the recent past. Nonetheless, all these attempts do not solve all attack types and have specific limitations and advantages. In this section of the work, the outcomes from the parallel researches are discussed.

It is often identified that, the security attacks are caused due to misconfiguration of the load balancing or the routing algorithms. The work by B. Abali et al. [8] has elaborated the misconfiguration and correction strategies of routing algorithms on cloud networks. Considering this phenomenon, the work by F. Araujo et al. [9] elaborates the concept of misdirecting the attackers. This policy cannot prevent the attacks, but can cause significant delay in the attacks. Yet another violation of the security is the attacks on the resources of the infrastructures. The recommendation from A. Brzeczko et al. [10] is a well-accepted solution securing the infrastructure on cloud using adaptive models.

As mentioned by T. E. Carroll et al. [11], the network configurations and analysis of the network traffic can lead to a high success rate in detecting the attacks. Nevertheless, this detection must be backed up with a suitable prevention mechanism. Also, the data access pattern can be an elaborative evidence for data breaches as suggested by L. Cheng et al. [12]. The improvements over the standard network architectures were able to resist maximum attacks on the cloud services. The work by A. Chowdhary et al. [13] suggested the recent improvements by deploying the SDN strategies.

The attacks on the frameworks are also been reported in the year of 2017 as the report from “The Apache Struts Project Management Committee” is published [14]. This indicates the mandate of including the security as a service component to all deployable frameworks on the cloud.

The mobile cloud computing agents, in spite of the location hiding policies, are not safe from the attacks. The work by D. Evans et al. [15] elaborates the attack types on the mobile cloud agents and few counter measures. The complexity of this solution is the increasing load on the routing algorithms. This problem was well addressed by A. Gupta et al. [16] with the tree based routing algorithm. Nonetheless, the reductions of the routing complexity of the requests have imposed few limitations such as region specificity of the agents. However, the work by V. Heydari et al. [17] could successfully address this problem. This solution was backed up by the works from J. B. Hong et al. [18] and J. H. Jafarian et al. [19].

Finally, as discussed in the work by N. Virvilis et al. [20] a good number of further researches are required to make the cloud services more secure and more so, provide a generic solution to address all attack types under a single framework.

Henceforth, these works identifies the challenges in the existing solutions and provide the novel solution, which is discussed in the further sections of this work.

III. Attack Types and Characteristics Identification

The individual attack types are the key point of effective detection of the attacks and further providing the preventions. Thus in the section of the work, the attack types are analysed with the proposed characteristics metric.

A. *Data Breaches*

The data breaches are the first types of attacks can be encountered on the cloud environments. Various studies have shown that this type of attack was encountered even before the cloud computing paradigm came into existence. During the data breach attack, the sensitive data is exposed to unauthorised access. This attack types can be identified if there is a high volume of data transfer in the network, which is unusual for the regular traffic. Also, the unusual access restrictions for any user profiles can be a significant hint of data breaches.

B. *Hijacking of Accounts (HoA)*

The second types of attack are the hijacking of the accounts or HoA. During this attack the user is often signed out of the portal and cannot regain access to the system. During this attack the hacker can obtain sensitive information from the accounts or can perform random tasks, which will be vulnerable to the application or the data. If any use in the system loses access to the resources or the account, then it is a clear indication of HoA.

C. *Insider Threat*

The third type of the attack can be most unlikely to happen, but with the deep drive into the security aspects reveal that this type of attack can happen. During this type of attacks, the attacker may make unauthorised access requests multiple times.

D. *Malware Injection*

The fourth type attack is the malware injection attack. This type of attack is usually introduced in the network by deploying a false instance in the cloud data centre. This instance eventually hampers the network and service functionalities. A sudden chance in the network architecture of unusual routing of the requests can be a hint for malware injection.

E. *Abuse of Cloud Services (ACS)*

The fifth type of attack is the ACS attack. This type of attacks is eventually generated by the legal users by hosting illegal applications of the contents on the cloud. The detection of this type of attacks is limited to the report from the victim. Also, this type of attacks can be detected by validating the application hosting rules from every country and then matching with the application characteristics.

F. *Insecure APIs*

The sixth type of attack is the insecure API attack. The API based access can be highly beneficial and at the same time highly risky for the hosted applications. Due to the vulnerable nature for authentication or the access or the effects on the access request encryption. The insure API access can be identified by analysing unauthorized access request and violation of encryptions.

G. *Denial of Service (DoS)*

The last popular attack type is the DoS attacks. This attack can make permanent damage to the applications by making the applications or part of the applications unavailable to the users. The detection of the DoS attacks can be carried out by identifying the random unavailability of the resources.

Henceforth with the detailed understanding of the attack types, in this section of the work, the characterization of the attacks are also formulated [Table 1].

TABLE I. ATTACKS AND CHARACTERISTICS

Attack Types	**Attack Characteristics**							
	High Data Transfer	**Access Restrictions**	**Resources Access Restrictions**	**Unauthorized Access Request**	**Architecture Change**	**Unusual Routings**	**NDA Violation**	**Encryption Violation**
Data Breaches	Yes	Yes				Yes		
Hijacking of Accounts		Yes	Yes	Yes				
Insider Threat	Yes			Yes			Yes	
Malware Injection					Yes	Yes		
Abuse of Cloud Services						Yes	Yes	
Insecure APIs				Yes				Yes
Denial of Service Attacks			Yes		Yes			

Further the detection of the characteristics from the client access requests must be performed; hence the first proposed algorithm for request characterization is furnished here.

```
Algorithm-1: Attack Detection Based on Characteristics (ADBC)

Step - 1.   Access the client connection request
Step - 2.   For each connection requests
            a.   Check for High Data Transfer
                      i.   If true then mark as T1
            b.   Check for Access Restrictions
                      i.   If true then mark as T2
            c.   Check for High Resources Access Restrictions
                      i.   If true then mark as T3
            d.   Check for Unauthorized Access Request
                      i.   If true then mark as T4
            e.   Check for Architecture Change
                      i.   If true then mark as T5
            f.   Check for Unusual Routings
                      i.   If true then mark as T6
            g.   Check for NDA Violation
                      i.   If true then mark as T7
            h.   Check for Encryption Violation
                      i.   If true then mark as T8
Step - 3.   End
Step - 4.   If T1 & T2 & T6 are True
            a.   Then mark as Data Breach
Step - 5.   If T2 & T3 & T4 are True
            a.   Then mark as HoA
Step - 6.   If T4 & T7 are True
            a.   Then mark as Insider Threat
Step - 7.   If T5 & T6 are True
            a.   Then mark as Malware Injection
Step - 8.   If T6 & T7 are True
            a.   Then mark as ACS
Step - 9.   If T4 & T8 are True
            a.   Then mark as Insecure APIs
Step - 10.  If T3 & T5 are True
            a.   Then mark as DoS
Step - 11.  Report the attack type
```

Furthermore, in the next section of the work, the proposed prevention model is elaborated.

IV. Security Policy Management

The proposed attack detection algorithm can identify the attack types and can further enable the security policy management protocols to be implemented. The detection of the attacks can temporarily relieve the network from the attackers, but it cannot prevent from the damage. Thus in this section of the work, the security policy management and deployment algorithm must be elaborated.

Though the applicability of the policies significant depends of the characteristics of the attacks and the predefined measures for prevention must be furnished first. Hence the preventive measures are elaborated first in this section of the work [Table 2].

TABLE II. Attacks and Prevention Measures

Attack Type	Preventive Measure
Data Breaches	• Match traffic pattern and disconnect the clients with high data requests • Restore the security access points • Update routing table
Hijacking of Accounts	• Restore the security access points • Update resource graphs • Disconnect the IP address with unauthorized requests
Insider Threat	• Match traffic pattern and disconnect the clients with high data requests • Disconnect the IP address with unauthorized requests • Match NDA and terminate application
Malware Injection	• Update Architecture graphs • Update routing table
Abuse of Cloud Services	• Update routing table • Match NDA and terminate application
Insecure APIs	• Disconnect the IP address with unauthorized requests • Update session keys, public and private keys
Denial of Service Attacks	• Update resource graphs • Update Architecture graphs

Further, the security policy management and deployment algorithm is elaborated here:

```
Algorithm–2: Security Policy Management & Deployment (SPMD)

Step - 1.   If T1 & T2 & T6 are True
            a.   Match traffic pattern and disconnect the clients
                 with high data requests
            b.   Restore the security access points
            c.   Update routing table
Step - 2.   If T2 & T3 & T4 are True
            a.   Restore the security access points
            b.   Update resource graphs
            c.   Disconnect the IP address with unauthorized
                 requests
Step - 3.   If T4 & T7 are True
            a.   Match traffic pattern and disconnect the clients
```

```
                  with high data requests
            b.    Disconnect the IP address with unauthorized
                  requests
            c.    Match NDA and terminate application
Step - 4.   If T5 & T6 are True
            a.    Update Architecture graphs
            b.    Update routing table
Step - 5.   If T6 & T7 are True
            a.    Update routing table
            b.    Match NDA and terminate application
Step - 6.   If T4 & T8 are True
            a.    Disconnect the IP address with unauthorized
                  requests
            b.    Update session keys, public and private keys
Step - 7.   If T3 & T5 are True
            a.    Update resource graphs
            b.    Update Architecture graphs
Step - 8.   Deploy the combined security policy
```

Further the complete automated framework as proposed in this work is elaborated in the next section of the work.

V. Proposed Automated Framework

As discussed in the previous sections of this work, a number of research attempts are carried out for detecting and preventing the attacks on the cloud and cloud services. The existing solutions are limited for two major reasons:

- The parallel research outcomes are not applicable to be deployed as security as a service. Thus cannot be incorporated within the services hosted on the cloud. To make the detection and prevention methods coupled into the services, the framework must be automated. The proposed framework in this work is automated can detect random attack events.
- Also, the parallel research outcomes are focused on single attack types. Thus detection of the newer attack types cannot be detected and prevented. In order to achieve this goal, the proposed framework is designed to be characteristics based, so that any new attack can be detected based on the violation of the normal application or request properties.

Henceforth, the proposed automated characterization based framework is elaborated here in this section of the work [Fig. 1].

Further in the next section of the work, the elaboration on the process flow and the algorithms is furnished.

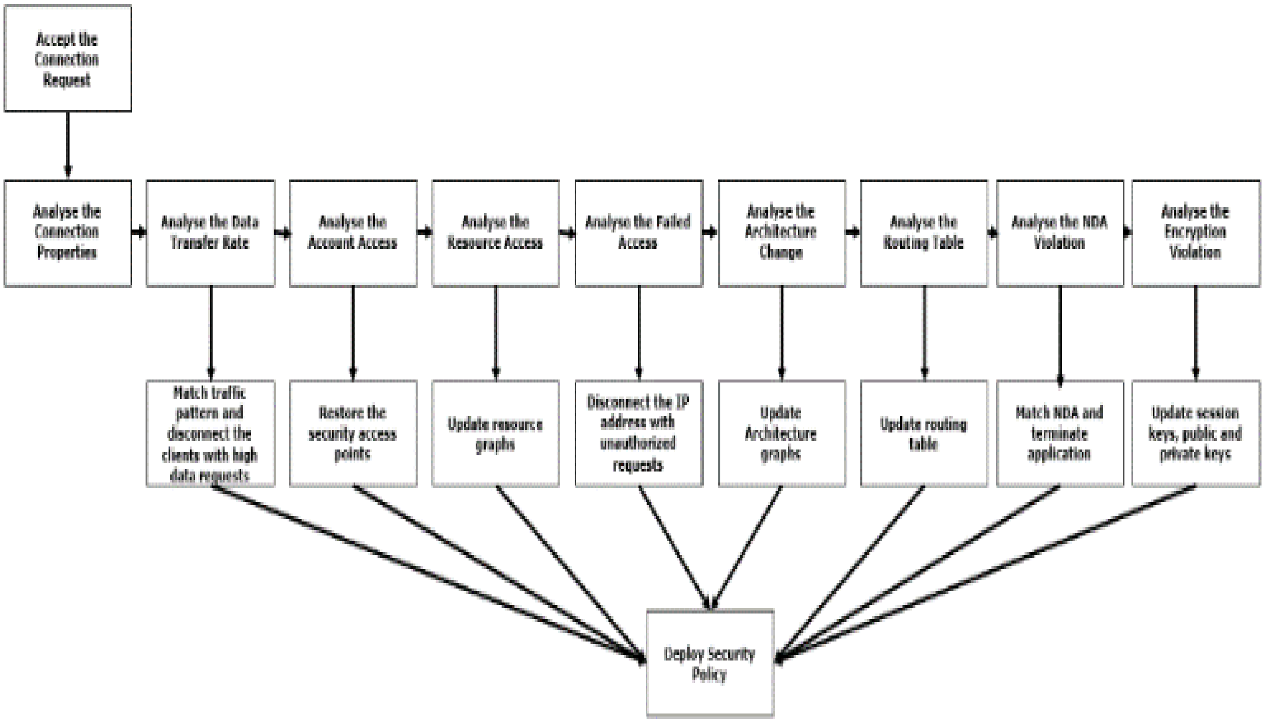
Accept the Connection Request
Analyse the Connection Properties
Analyse the Data Transfer Rate
Analyse the Account Access
Analyse the Resource Access
Analyse the Failed Access
Analyse the Architecture Change
Analyse the Routing Table
Analyse the NDA Violation
Analyse the Encryption Violation
Match traffic pattern and disconnect the clients with high data requests
Restore the security access points
Update resource graphs
Disconnect the IP address with unauthorized requests
Update Architecture graphs
Update routing table
Match NDA and terminate application
Update session keys, public and private keys
Deploy Security Policy

Fig. 1. Proposed Automatic Framework for Detection and Prevention of the Cloud Attacks

VI. Proposed Automated Algorithm

The success of any framework can be measured based on the driving algorithm. Thus in this section of the work, the automated algorithm running behind the framework is elaborated:

Algorithm–3: Automatic Detection and Prevention of the Cloud Attacks (ADPCA)

```
Step - 1.  Accept the client request
Step - 2.  Extract the IP address of the client
           a.  For each IP in the connection request
                 i.  Extract the header
                ii.  Extract the IP table from the header
               iii.  Identify the source IP
           b.  End
Step - 3.  Extract the location from IP address
           a.  For each IP as source
                 i.  Map the IP segments with Geolocation API
                ii.  Extract the location
           b.  End
Step - 4.  Identify the data transfer rate for the connection
Step - 5.  If the data transfer rate > Network standard transfer rate
           a.  Disconnect the clients with high data requests

Step - 6.  If the connection request terminates
           a.  Check for connection duration
           b.  If connection duration < Network standard connection duration
                 i.  Restore the connection
           c.  Else if connection request failed > 5 times
                 i.  Disconnect the IP address with unauthorized requests
Step - 7.  Identify the resource access by the connection
           a.  If the connection resoruce access <> Network standard resource graphs
                 i.  Update the resource graphs
Step - 8.  Identify resource updates
           a.  If new resource included <> Resource graph pattern
                 i.  Terminate the resource
           b.  Else if existing resource terminated <> Resource graph pattern
                 i.  Restore the resource
Step - 9.  Identify the routing pattern
           a.  If routing pattern <> routing table
                 i.  Update routing table
Step - 10. If the location policy <> Connection policies
           a.  Terminate connection
Step - 11. Identify the resource access time stamps
           a.  If restricted resource access time stamp = Recent time stamp
                 i.  Update session keys and Public-Private key pairs
Step - 12. Repeat Step - 4 to 11.
```

Henceforth, the comparative analysis is presented in the next section of this work.

VII. Comparative Security Analysis

In order to claim the superiority of the proposed method, there must be a comparative analysis. Hence, in this section of the work, the comparative analysis is carried out [Table 3].

Thus it is natural to realize that the proposed framework and the algorithms are significantly better performing compared to the other parallel research outcomes.

The ranking analysis is also visualized graphically [Fig. 2].

Hereafter, with the comparative analysis, the results are discussed in the next section of the work.

TABLE III. Comparative Analysis & Ranking

Solution Name	**New Attack Detection Capabilities**	**Accuracy**	**Scalability**	**Security As A Service Applicability**	**Time Complexity**	**Ranking**
CTBM	No	High	Yes	No	High	4
SPRT	No	Modarate	No	No	High	6
BOT	No	Low	No	No	Modarate	3
Attack	No	High	Yes	No	High	2
MVLVAL	No	Modarate	No	No	Modarate	5
NICE	No	High	Yes	No	High	7
Proposed ADPCA Algorithm	Yes	High	Yes	Yes	Low	1

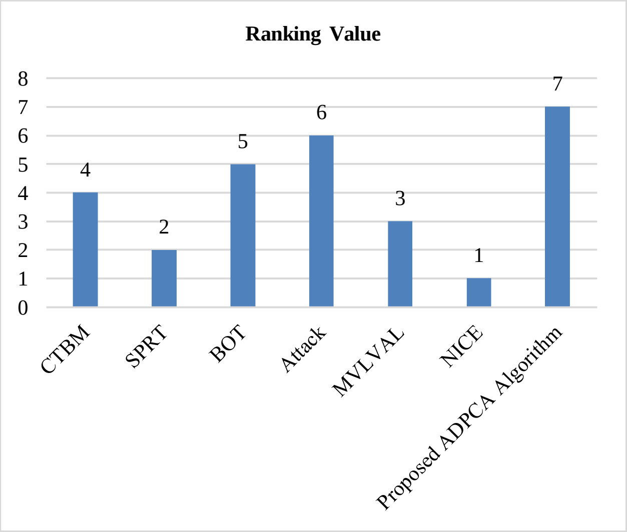

Fig. 2. Ranking Analysis–Comparative Analysis.

VIII. Results and Discussions

The results from the proposed automated framework are highly satisfactory. The obtained results from the proposed framework for each component are discussed in this section.

A. IP Address Extraction from Connection Requests

Firstly, the IP address extraction process results from each connection requests are presented. The IP address extraction is one of the core components of the framework [Table 4].

Also, the success rate for the overall execution duration is analysed [Table 5].

The results are analysed graphically here [Fig. 3].

TABLE IV. IP Address Extraction from Connection Request

Connection Sequence	Detected IP Address	Status of the IP Extraction Process
Seq - 1	202.198.31.26	Success
Seq - 2	216.194.164.12	Success
Seq - 3	199.163.30.63	Success
Seq - 4	219.239.174.162	Success
Seq - 5	219.245.8.10	Success
Seq - 6	207.109.235.76	Success
Seq - 7	221.103.131.23	Success
Seq - 8	206.178.5.105	Success
Seq - 9	220.122.165.101	Success
Seq - 10	215.180.174.50	Success

TABLE V. IP Address Extraction from Connection Request Success Rate

Number of Connection Requests	Number of IP Addresses Extracted Successfully	Success Measure (%)
1035	1035	100

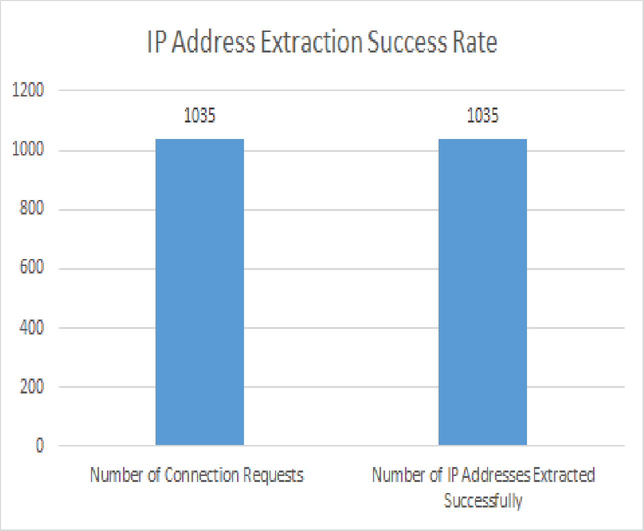

Fig. 3. IP Extraction Accuracy.

Thus the extraction of the IP address component demonstrates 100% accuracy.

B. Location Extraction from IP Address

Detection of the location is also one of the prime components of this framework, as the NDA or Non-Disclosure Agreement violations can be detected based on this factor. The detection results are elaborated here [Table 6]

Also, the success rate for the overall execution duration is analysed [Table 7].

The results are analysed graphically here [Fig. 4].

TABLE VI. Location Detection

IP Address	Actual Location	Detection Location	Location Detection Status
202.198.31.26	Oceania	New Zealand	Success
216.194.164.12	North America	United States	Success
199.163.30.63	Europe	Czech Republic	Success
219.239.174.162	Asia	Japan	Success
219.245.8.10	Asia	Japan	Success
207.109.235.76	North America	United States	Success
221.103.131.23	Asia	Japan	Success
206.178.5.105	North America	Canada	Success
220.122.165.101	Asia	Japan	Success
215.180.174.50	North America	United States	Success

TABLE VII. Location Detection Success Rate

Number of IP Addressed Tested	Number of Locations Detected Successfully	Success Measure (%)
1035	1035	100

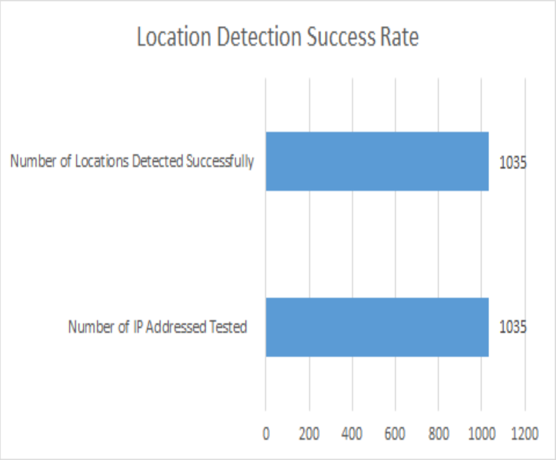

Fig. 4. Location Detection Accuracy.

C. Data Transfer Rate Identification & Validation for Attacks

Based on the application type the threshold can be set for the data transfer rate. The connections violating the predefined transfer rates can be identified as attacks. The results from this component are furnished here [Table 8].

The analysis result is visualized graphically here [Fig. 5].

D. Connection Duration Identification & Validation for Attacks

The duration for the connection indicates the significance of the attacks. In case of a standard application type, the connection duration can be predetermined and in case of over timing of any connections can be a potential attack. The results from this component are elaborated here [Table 9].

The results are analysed graphically [Fig. 6].

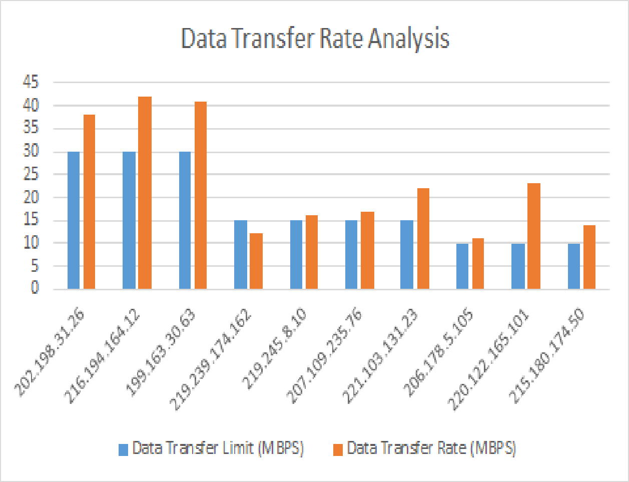

Fig. 5. Data Transfer Rate Analysis.

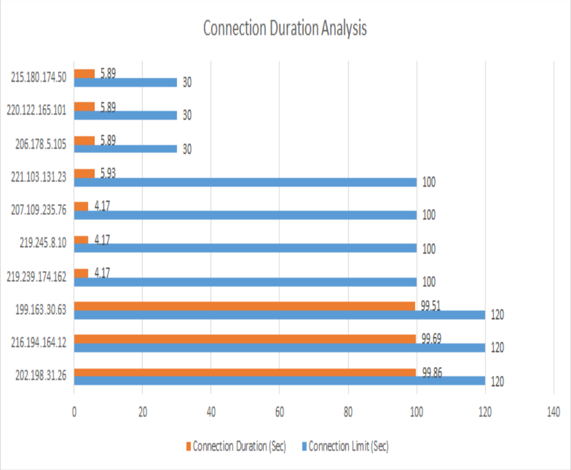

Fig. 6. Connection Duration Analysis.

E. Resource Access Time Stamp Validation & Validation of Attacks

The resource access time stamps for the allowed or for the restricted resources can be a deterministic factor for detection of the attacks. The resources which are identified by the service as restricted, having most recent time stamp can be a strong witness of the attacks. The results from this module are elaborated here [Table 10].

Hence, it is natural to realize that the proposed automated framework can identify and prevent the attacks with 100% accuracy.

Further, with the detailed presentation and discussion on the results, this work presents the final conclusion of this work in the next section.

TABLE VIII. DATA TRANSFER RATE & ATTACK IDENTIFICATION

Application Type	Data Transfer Limit (MBPS)	Connection From	Data Transfer Rate (MBPS)	Analysis
Data Storage	30	202.198.31.26	38	High Volume Transmissions
		216.194.164.12	42	Attack
		199.163.30.63	41	Attack
Email Application	15	219.239.174.162	12	Normal
		219.245.8.10	16	Normal
		207.109.235.76	17	Normal
		221.103.131.23	22	Attack
Dash Board Application	10	206.178.5.105	11	Normal
		220.122.165.101	23	Attack
		215.180.174.50	14	High Reads

TABLE IX. CONNECTION DURATION ANALYSIS & ATTACK IDENTIFICATION

Application Type	Connection Limit (Sec)	Connection From	Connection Start Time Stamp	Connection End Time Stamp	Connection Duration (Sec)	Analysis
Data Storage	120	202.198.31.26	17:34.3	27:09.5	99.86	Normal
		216.194.164.12	17:35.3	27:09.5	99.69	Normal
		199.163.30.63	17:36.3	27:09.5	99.51	Normal
Email Application	100	219.239.174.162	26:46.0	27:10.0	4.17	Early Disconnect - Attack
		219.245.8.10	26:46.0	27:10.0	4.17	Early Disconnect - Attack
		207.109.235.76	26:46.0	27:10.0	4.17	Early Disconnect - Attack
		221.103.131.23	26:46.0	27:20.1	5.93	Early Disconnect - Attack
Dash Board Application	30	206.178.5.105	26:46.2	27:20.1	5.89	Early Disconnect - Attack
		220.122.165.101	26:46.2	27:20.1	5.89	Early Disconnect - Attack
		215.180.174.50	26:46.2	27:20.1	5.89	Early Disconnect - Attack

TABLE X. CONNECTION DURATION ANALYSIS & ATTACK IDENTIFICATION

Resource Type	Connection From	Access Time Stamp	System Time Stamp	Analysis
Restricted	202.198.31.26	26:46.2	26:46.2	Attack
Restricted	216.194.164.12	26:46.2	26:46.2	Attack
Unrestricted	199.163.30.63	26:51.6	26:51.6	Normal
Unrestricted	219.239.174.162	26:51.6	26:51.6	Normal
Restricted	219.245.8.10	26:51.6	26:51.6	Attack
Unrestricted	207.109.235.76	26:57.3	26:57.3	Normal
Restricted	221.103.131.23	26:57.3	26:57.3	Attack
Unrestricted	206.178.5.105	26:57.3	26:57.3	Normal
Restricted	220.122.165.101	27:02.5	27:02.5	Attack
Unrestricted	215.180.174.50	27:06.0	27:06.0	Normal

IX. CONCLUSION

The notoriety of the distributed computing not just pulled in the application designers, server farm proprietors and the purchasers, yet additionally pulled in a colossal number of attackers. The attacker tries to gain access to the cloud services, cloud networks, resources and the data. These unauthorized accesses lead to huge losses. In order to detect and prevent the attacks, a number of research attempts are carried out. The parallel research outcomes fail in making the detection process automatic and most of the cases struggle to detect newer types of attacks. Thus this work proposes a framework to analyses the connection types in order to detect standard attack types and the newer attacks as well. The characterization makes the proposed method significantly better than the other research outcomes. Also this work demonstrates a significant property of the proposed framework as the proposed framework can be automated for detection of the attacks and can eventually be integrated as security as a service for the other services hosted on the cloud environments for making the cloud computing a better dimension for the application service industry.

REFERENCES

[1] Z. Su and G. Wassermann, “The essence of command injection attacks in Web applications,” in Proc. Conf. Rec. 33rd ACM SIGPLAN-SIGACT Symp. Principles Programm. Lang. (POPL), New York, NY, USA, 2006, pp. 372–382.

[2] A. Stasinopoulos, C. Ntantogian, and C. Xenakis, “Commix: Detecting and exploiting command injection flaws,” Dept. Digit. Syst., Univ. Piraeus, Piraeus, Greece, White Paper, Nov. 2015.

[3] An In-Depth Analysis of SSH Attacks on Amazon EC2. Accessed: Feb. 1, 2017. [Online]. Available: https://blog.smarthoneypot.com/in-depth-analysis-of-ssh-attacks-on-amazon-ec2/

[4] G. Badishi, A. Herzberg, and I. Keidar, “Keeping denial-of-service attackers in the dark,” IEEE Trans. Depend. Sec. Comput., vol. 4, no. 3, pp. 191–204, Jul. 2007.

[5] Q. Jia, K. Sun, and A. Stavrou, “MOTAG: Moving target defense against Internet denial of service attacks,” in Proc. 22nd Int. Conf. Comput. Commun. Netw. (ICCCN), Jul. 2013, pp. 1–9.

[6] W. G. Morein, A. Stavrou, D. L. Cook, A. D. Keromytis, V. Misra, and D. Rubenstein, “Using graphic turing tests to counter automated DDoS attacks against Web servers,” in Proc. 10th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2003, pp. 8–19.

[7] A. Stavrou, A. D. Keromytis, J. Nieh, V. Misra, and D. Rubenstein, “MOVE: An end-to-end solution to network denial of service,” in Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS), San Diego, CA, USA, 2005, pp. 81–96.

[8] B. Abali and C. Aykanat, “Routing algorithms for IBM SP1,” in Proc. 1st Int. Workshop Parallel Comput. Routing Commun. (PCRCW), 1994, pp. 161–175.

[9] F. Araujo, K. W. Hamlen, S. Biedermann, and S. Katzenbeisser, “From patches to honey-patches: Lightweight attacker misdirection, deception, and disinformation,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2014, pp. 942–953.

[10] A. Brzeczko, A. S. Uluagac, R. Beyah, and J. Copeland, “Active deception model for securing cloud infrastructure,” in Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS), Apr. 2014, pp. 535–540.

[11] T. E. Carroll and D. Grosu, "A game theoretic investigation of deception in network security," in Proc. 18th Int. Conf. Comput. Commun. Netw. (ICCCN), Washington, DC, USA, Aug. 2009, pp. 1–6.

[12] L. Cheng, F. Liu, and D. D. Yao, "Enterprise data breach: Causes, challenges, prevention, and future directions," Data Mining Knowl. Discovery, vol. 7, no. 5, p. e1211, 2017.

[13] A. Chowdhary, S. Pisharody, and D. Huang, "SDN based scalable MTD solution in cloud network," in Proc. ACM Workshop Moving Target Defense (MTD), New York, NY, USA, 2016, pp. 27–36.

[14] The Apache Struts Project Management Committee. (2017). Apache Struts Statement on Equifax Security Breach. [Online]. Available: https://blogs.apache.org/foundation/entry/apache-struts-statement-on-equifax

[15] D. Evans, A. Nguyen-Tuong, and J. Knight, Effectiveness of Moving Target Defenses. New York, NY, USA: Springer, 2011, pp. 29–48.

[16] A. Gupta, A. Kumar, and M. Thorup, "Tree based MPLS routing," in Proc. 15th Annu. ACM Symp. Parallel Algorithms Archit. (SPAA), New York, NY, USA, 2003, pp. 193–199.

[17] V. Heydari, S.-I. Kim, and S.-M. Yoo, "Scalable anti-censorship framework using moving target defense for Web servers," IEEE Trans. Inf. Forensics Security, vol. 12, no. 5, pp. 1113–1124, May 2017.

[18] J. B. Hong and D. S. Kim, "Assessing the effectiveness of moving target defenses using security models," IEEE Trans. Depend. Sec. Comput., vol. 13, no. 2, pp. 163–177, Mar. 2016.

[19] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "An effective address mutation approach for disrupting reconnaissance attacks," IEEE Trans. Inf. Forensics Security, vol. 10, no. 12, pp. 2562–2577, Dec. 2015.

[20] N. Virvilis, B. Vanautgaerden, and O. S. Serrano, "Changing the game: The art of deceiving sophisticated attackers," in Proc. 6th Int. Conf. Cyber Conflict (CyCon), Jun. 2014, pp. 87–97.