National College of Ireland

Joint Honeypot Networks and Hybrid Intrusion Detection System for Mobile Cloud Computing

MSc Internship
Cybersecurity

Surya Prakash Subramaniam Govindaraj
x18149090

School of Computing
National College of Ireland

Supervisor: Imran Khan

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Surya Prakash Subramaniam Govindaraj
Student ID:	x18149090
Programme:	CyberSecurity
Year:	2019-2020
Module:	MSc Internship
Supervisor:	Imran Khan
Submission Due Date:	29/01/2020
Project Title:	Joint Honeypot Networks and Hybrid Intrusion Detection System for Mobile Cloud Computing
Word Count:	3837
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	29th January 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Joint Honeypot Networks and Hybrid Intusion Detection System for Mobile Cloud Computing

Surya Prakash Subramaniam Govindaraj
x18149090

Abstract

The growing smartphone technology and emerging mobile cloud technology are the latest wireless technology. Mobile cloud computing has many of the advantages that look forward to the future and it's also simple for hackers to take full control of many other users Privacy of Data. While data security is expected to be secured, the main drawback for users when the computer is connected to the internet it's not that difficult for an intruder to engage in a data theft on the required target. So, for providing better security the combination of Hybrid Intrusion Detection System (HyInt) and Honeypot networks is thus implemented into Mobile Cloud Environment with the significant purpose of mitigating unidentified and known attacks in order to provide security. Execution of the research work provides a pure perspective of the security and quality products of the algorithm that was not included in the previous research work. As part of the research work, intensive statistical analysis was performed to prove the consistency of the proposed algorithm. The implementation and evaluation outcome offers clear potential for any further research work in the cloud-based Intrusion Detection System. The implemented algorithm can be used for high-security cloud environment that is developed for army and banking purposes to monitor the network's activities effectively.

Keywords – Hybrid Intrusion Detection System, Honeypot Networks, Signature and Anomaly based detection, Mobile Cloud Computing, Performance.

1 Introduction

Mobile Virtualization is the most highly developed feature arising all over in today's world, and its uses for smartphones are increasing day by day. The mobile user is continuously increasing as it allows the work to be simple and faster, where it provides the latest technology that is rapidly growing and allows the user to access all the apps via the network from anywhere in the world. Mobile cloud computing has a major advantage where the use of Mobile Cloud Computing (MCC) is very versatile and we can access the data and share information anywhere in the world unless we are connected to the internet, It also offers cost-effectiveness where use and maintenance becomes comparatively low and real-time data availability, where all user information is available in real-time on our mobile device when connected to the network from which we can update and access the data via cloud-based online services, as well as data backup when uploaded to the cloud for security purposes. Despite all the hype of the MCC, it lacks the major disadvantage of privacy and security which contributes to trustworthy problems for consumers and businesses as the innovation is evolving in the world the hacker's increase day by day.

Similarly, the companies are also implementing new things and methods for protection where the cloud computing services are available on the pay to go to secure the cloud environment.

How Hybrid Intrusion Detection System (HyINT) and Multi-Honeypot Network (MHN), when implemented together provide better security features in Mobile Cloud Computing?

The implementation of Honeypot networks is used to achieve more defense in depth protection and total security of the cloud environment, the implementation of honeypot networks is used to achieve more defense in depth protection and total security of the cloud environment, the analysis of attack approaches is identified in the honeypots network as necessary for countermeasures. Many harmful threats such as DDOS, XSS injection, SQL injection cannot be prevented entirely but can be avoided. Where there are several ways to protect it from hackers, but IDS is the most critical and common way to detect any malicious code in a network where it plays a crucial part in securing the cloud environment from the attackers [1]

2 Related Work

In this section we provide the background and the work related to the proposed solution in this paper. Where we discuss about the Mobile Cloud Computing and its security issues.

2.1 Emphasis of Mobile Cloud Computing

Mobile Cloud Computing is the present and trending technology all over the world, and it has various benefits, which is very useful in the way of enriching the user experience. [2] From which it has specific functions such as storage, smartphone mobility anywhere via wireless or internet access and its service is simply pay as you go. Similarly, as resulted by the Juniper Research is the growing use of mobile computing, which notes that the public and private sector demand for cloud-based mobile applications which is expected to increase to 9.5 billion dollars as predicted by 2014, yet hopefully in the near future it will increase more than that. Similarly applications for smartphones have become numerous in past few years with applications in different categories such as entertainment, social media, online streaming, banking, news, and so on the main cause behind this is that the mobile computing is capable of providing the subscriber with a resource where and how it is required purely on the basis of user organization. As shown in an analysis reported by International Data Corporation (IDC) in 2009, where 74% of IT administrators and Chief Information Officer (CIOs) find that user privacy concerns are the major risks that stopped most organizations from jumping into virtualization. There are 3 fundamental principles which mainly benefits in mobile computing like technology, hardware and communications. Where hardware consists of devices such as smartphones, portable devices which can be used by clients. However, with the wireless network's rapid progress, consumers are gradually embracing PDAs. [3] More than 2.4 billion consumers will use a portable device to arrive at cloud computing platform for 2015 during the Allied Business Intelligence report. Similarly, Google highlights certain cloud-based products for consumers and companies, where it has a necessary item for mobile phones which is currently trending all over the world known as Android OS also it has various applications like google maps, streets, etc. Similarly, Google has launched an emerging technology

known as Google Stadia which is a cloud-based gaming service it does not require any hardware as it just needs an internet connection to connect. [4]

The below Figure 1 shows a design of the MCC process, the core techniques used in the technology industry like parallelization model, virtualization and mass production are the three primary techniques for cloud computing.

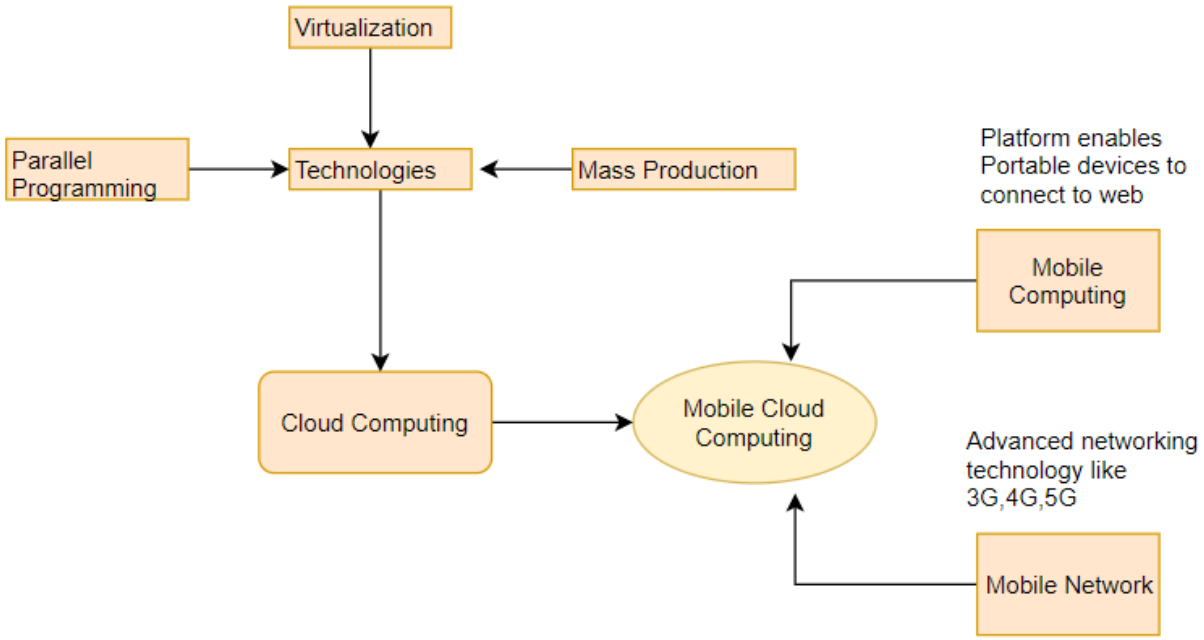

Figure 1: Architecture of Mobile Cloud Computing

2.2 Importance of Cloud Security

Mobile devices are prone to several external threats where they can cause unknown attacks as people use mobile phones in cloud environments, whereas information privacy and authentication should be known to regular users and software developers where if they are aware of the outcomes of the privacy there will not be any problems with the hackers. People nowadays don't know the usage of technology and the advanced features in their smartphones and their PDA's. Through various security features including through app installation such as anti-virus's mobile protection can be achieved. [1] [5]

Security frameworks for Mobile Cloud Computing (MCC) is classified into two groups, application security and data security frameworks, storing data on a database in a virtual environment without revealing any details is more difficult for mobile users. An authentication method is used to verify that if a user transfers a file to a cloud server for sharing with different clients, similarly it should also be checked that perhaps the user accessing the file is a trustworthy client, scalability is the capability of a network which helps to interact with clients in an impeccable manner. [19] Similarly, the latest security technologies for online services should be introduced such as VPN usage, encryption of password, authentication and entry command can thus provide uninterrupted services against various attacks like DOS attacks and data theft. [23] Therefore, when such attacks occur the cloud services must provide a backup and restore service that can improve

customer trust. The below table shows the recent security issues and current approaches as follows in Figure 2

Security Issues		Current Approaches
Mobile Cloud	Platform Reliability	Authentication and access control, Privacy and data protection.
	Privacy and Data Protection	Key management and data encryption. Integrating the current security technologies.
Mobile Terminal and Network	Malware software	Detection and prevention CloudAV
	Software Vulnerabilities	Installing the system patches, checking software legitimacy and integrity.
	Information Leakage	Data Encryption and Security Protocol

Figure 2: List of Security Issues

2.3 Potential of Intrusion Detection System in Cloud

An intrusion is any attack which might compromise a device or network's CIA, and there are many possibilities of intruder attacks the most common is (DOS) attacks Denial of Service, when this attempt occurs legitimate users cannot access internet-based services. [6] In the virtual environment, the intruder can send repeated attempts to authenticate VMs via cyborgs, thus overloading their availability to legitimate users. The implementation of Intrusion Detection and Prevention Systems (ID/ PS) that are still accessible could not achieve the necessary level of protection and performance. Pandeeswari and Kumar (2016) have applied a Fuzzy Mean Clustering-based ANN that detects breaches in the cloud, where IDS usually operates in the above methods and implemented on end host cloud servers. [5] [7]

By using authentication techniques, potential ransomware will prevent the use of conventional HIDS based on signature matching methods. By testing the controlled computer with the aid of the security process, complex evaluation based on existing IDS can be prevented. [8] Signature matching approaches require proper monitoring, later another level of protection (Modi and Patel, 2013) connects modern NIDS tools with traditional anomaly detection method which detect cyberattacks into a network. Similarly, some services like Snort IDS which is active by cloud protection, fail to recognize VM attacks target from individual residents to different on a physical server. The below Figure 3 shows the different types of Cloud IDS. [9] The Hybrid Intrusion Detection system is

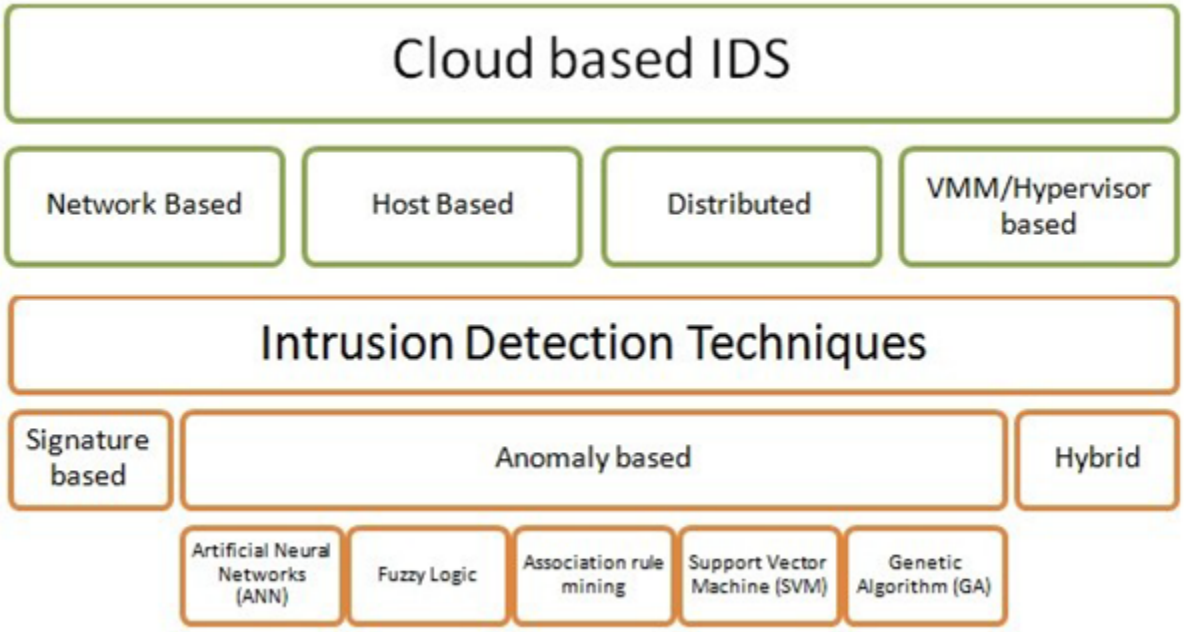

Figure 3: Structure of Intrusion Detection System in Cloud environment

the effectiveness of IDS, which can be significantly enhanced by combining signature-based techniques with anomaly-based techniques. The resilience to new unknown attacks that benefit from the existing knowledge already generated by known attacks. X. Wang et al suggested a methodology that relates to the central management approach, even though it has the drawbacks of all strategies that use centralized control in a distributed environment [10], Similarly, Modi et al initiated a method for a stepwise detection of intrusion. It originally pre-processes packets and transmits them to signature-based IDS after comparing them with patterns that have already been found, Hybrid IDS is more beneficial in terms of vulnerability security and also performance. The main constraints to previous solutions were that they could not be fully designed to handle new types of attacks, where this is also a time-consuming task that requires too much time to examine suspicious attacks [11]

2.4 Honeypot uses of Intelligence

In network security, honeypots are a sophisticated idea, Such a system aims to gather information about intrusion attempts. The level of interaction varies from minimum interaction honeypots, emulating only the communication layer, to strong interaction honeypots, running a real operating system. One of the main reasons for using a cloud services takes advantage of lower IT infrastructure and company costs, and it is to collect high and low communication honeypots used in a cloud environment to evaluate the attacks, they must verify that the distributed packets are legitimate once they are transferred to HoneyCY as their transition to the cloud [12]. Similarly, it is made up of 3 design layers where in this HoneySrv collects honeypie devices and information gathered, also HoneyVm analyzes collected malware. Brown et al listed numerous virtualization systems involved in honeypot sensors, and Saadi et al provided IDS focused on a smartphone device with a mixture of honeypots such as Honeycomb, HoneyNet and HoneyD. [13] [14].

The below Figure 4 shows the architecture of the Honeypot function.

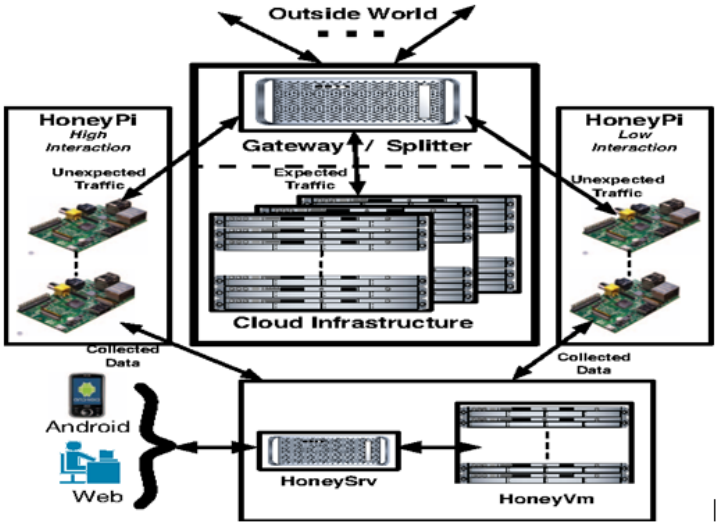

Figure 4: Structure of Honeypot

3 Methodology

This process is focused on more improvisation from the existing scenario, which is as follows in the proposed architecture. The ideas which are majorly recommended to the protection of the Cloud Service can be defended by the joint combination of Hybrid Intrusion Detection System and Multi Honeypot Networks, where it identifies for any intruder alerts and slows down the attackers. These can be accomplished by the Implementation of Hybrid Intrusion Detection System (HyInt) in a Cloud Service, also the deployment of Honeypot Networks and creating a certain rules of regulations to be followed for the precautions of any intruder alerts, if in such case of removing a appropriate malware data from log files and conducting a binary analysis of dynamic malware. [15]

3.1 Hybrid Intrusion Detection System

In this approach it has the advantages of the combination of both anomaly and signature-based intrusion detection systems as it can find any unknown attacks also it has the knowledge of known attacks. Also, the main gain of this process is which can be functional in both cloud and grid computing environments also it has the least false positive rate. Similarly, Arshad et al, gave an abstract model that satisfies all kinds of possible solutions for the requirements necessary. Where in this a system that has minimum human contact with improved response times, although in the real system it will be much more intricate and problematic to implement. There are some techniques which are implemented in the step-wise intrusion detection system, at first it will track any packets which is transferred to them with a signature-based IDS known as SNORT, and it compares whether it already exists, similarly in case if the matching attempt is failed a verdict tree algorithm will be used for anomaly-based intrusion detection. Next to the final, a set of possible innovative

rules are created, and the signature database is updated, which makes the Hybrid IDS by the way of more efficient in terms of security alongside vulnerabilities and performance. [20]

Throughout this Hybrid Intrusion Detection System, it follows an algorithm that implements 3 methods, such as Anomaly analysis, misuse analysis phase and authentication phase, that checks if the user is signed up or not, similarly if the user is not recognized it will increase the user's warning. Secondly, it will process misuse analysis, which validates user login credentials and MAC Address. [16] Similarly, the below algorithm from Figure 5 to Figure 8 represents the structure of the design to be followed by the application where it denotes the process to be running in a secured way and the Figure 9 represents the flowchart of the application running.

A. HIDS (Host-based Intrusion Detection System)Algorithm

Input: Incoming Requests made by user

Output: Alert for abnormal request along with details of every request made

```
1) Begin
2) Initialize, number of users of the
   cloud (n)
3) For each user k←1 to n
   {
   a) Initialization and execution of
      authentication_stage();
   b) Initiating
      Signature_analysis_stage();
   }
4) End
```

Figure 5: HIDS Algorithm

B. Authentication Phase:

Input: Login credentials, MAC Address

Output: Authenticated user gets registered

```
1) Begin
2) For each n users of the cloud
   {
   a) Get email id 'email_id' and
      password 'pwd'
   b) Auto-retrieve MAC address 'mac_id'
      of system
   c) If (email_id is valid && pwd is
      valid && mac_id is valid )
   d) Then, successfully register the
      user
   e) Else,
      {
      i) Stop authentication_stage();
      ii) Report admin of abnormal
          activity detection.
      }
   }
```

Figure 6: Authentication Phase

C. Signature Analysis Phase:

Input: email_jd, pwd, mac_jd

Output: Examine and report about abnormal requests along with displaying details of each request.

```
1) Begin
2) Examine all incoming requests;
3) Fetch email_id, pwd & mac_id;
4) Extract protocol 'ptcl' & port number
   'port_no' for all incoming requests;
5) If(email_id is invalid && pwd is
   invalid && mac_id is invalid)
   {
   a) Then, Request may be an intrusion;
   b) Alert admin of possible intrusion
      detection;
   }
6) If(ptcl is unknown && port_no is
   unknown)
   {
   a) Then, Request may be an intrusion;
   b) Alert admin of possible intrusion
      detection;
   }
7) If(signature_analysis phase
   successful) Then
   {
   a) Start new anomaly_analysis stage;
   }
8) Else, Report Admin for monitoring and
   controlling activities in network;
9) End;
```

Figure 7: Signature Analysis Phase

D. Anomaly analysis Phase:

Input: Incoming Requests made by user

Output: Flaunt Bandwidth and Speed of the incoming request

```
1) Begin
2) Set threshold value of speed 'τr' and
   bandwidth 'τb';
3) For each incoming request i←1 to 10
   a) Extract speed of request 'rspeed'
      & bandwidth of request 'bdw' &
      display them;
4) If (rspeed<= τr) Then,
   a) Request may be intrusion;
   b) Mark & display request;
5) Else, If (bdw<= τb) Then,
   a) Request may be intrusion;
   b) Mark & display request;

6) Calculate average of all request speed
   'avg_speed' ;
7) Calculate average of all bandwidth
   'avg_bdw';
8) If(avg_speed > τr) Then, τr← avg_speed;
9) Elseif(avg_bdw > τb) Then, τb←avg_bdw;
10)   Repeat for all Incoming requests;
11)   End;
```

Figure 8: Anomaly Analysis Phase

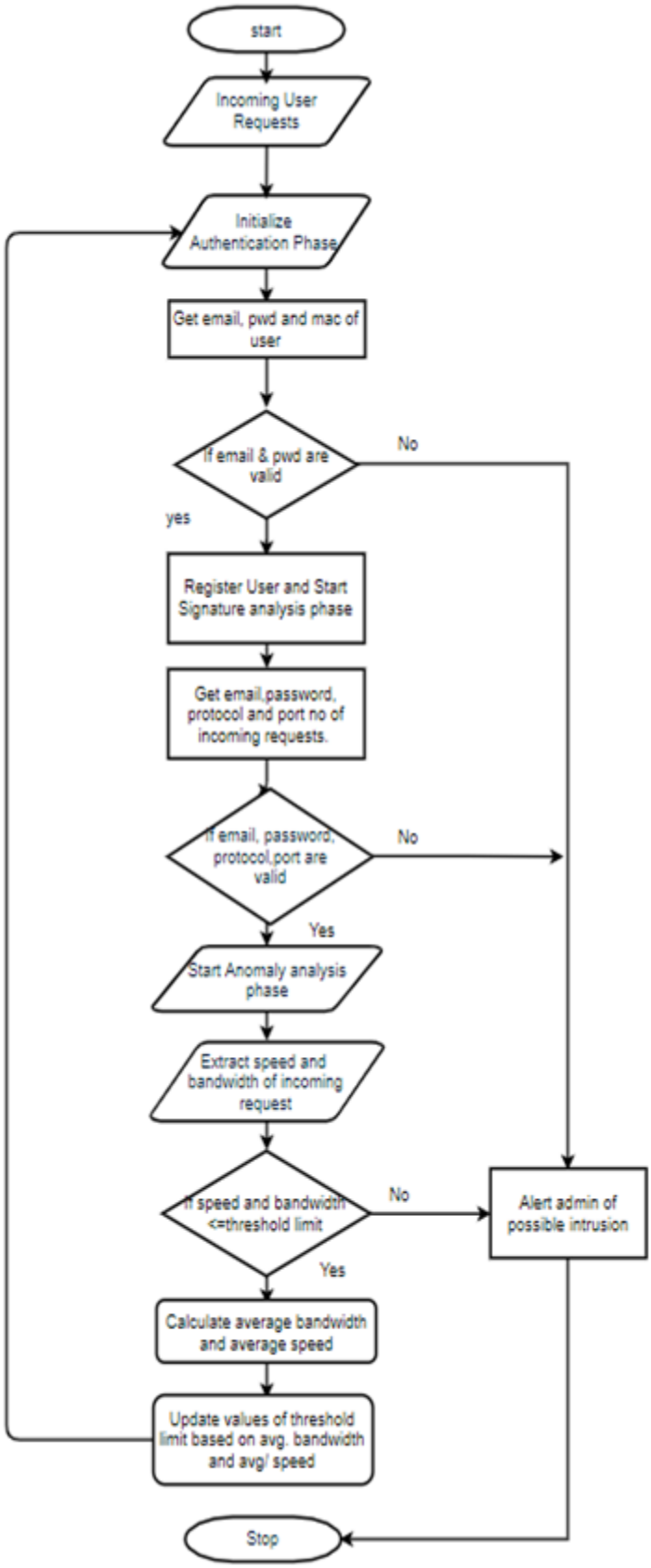

Figure 9: Flowchart of Hybrid Intrusion Detection System

3.2 Multi Honeypot Network

Honeypot is a decoy system that ensures exploits through the simulation of one or more vulnerable hosts where the intruder gives an easy-going goal. Where it uses specific features to lure an intruder into attacking its host system when monitoring the device activity and actions of all, also making documents of these attacks. Similarly, when the thumbprint pattern of incidents and unidentified activities occur after the honeypot has been activated, then it can track hosts to view the activity and identify if the activity is an unknown attack or not [18]. The virtual environment is subject to various attacks from outside and inside due to its various types of use and traffic because most of the attacks are from outside, thus the collection of details such as target IP address, network types, ports used, operating system and device vulnerabilities must be identified before to take immediate mitigation steps. Similarly, cloud service should not be stopped from being used, but we also use maximum and minimum interface for honeypot networks as a way of achievement. Methodology discusses the dynamic analysis of the samples obtained from the honeypot networks, then the samples running in a specific sandbox environment with the goal of achieving this model feature, through the device impact of the attack, the data acquired can be further calculated and make the signature module available. Signature module includes two sections namely Rule Generator and Rule Updater that has some rules. [17]

3.3 Rules of the Procedure

The honeypot database maintains attack information in the form of operations and logs, and then the rule generator runs a shell script to generate specific rules from the information collected. In this, there are various kinds of attacks but the 2 most common web-based attacks are SQL injection and Cross-Site Scripting (XSS). Now, HyInt can find these types of attacks, but often because of avoiding techniques and the lack of information on unknown frameworks which can be inserted through these attacks. Similarly, the second kind of intrusion depends on suspicious binary files and SSH attacks, when this kind of attack occurs, the data can be initiated as Dionaea, cowire and glastopf in the honeypot database. [21] [22]

4 Design Specification

The architecture of the proposed system is based on the Hybrid Intrusion Detection System and Honeypot networks, where its architecture follows the application process, which is deployed in a virtual environment where the data is collected. It follows the common network, throughout this we have combined 2 methods proposed to prove a strong security as Hybrid Intrusion Detection System (HyInt) which can prevent unidentified and identified attacks by using a proposed algorithm and the second method is honeypot network, which is capable of luring the attackers or trying to delay themselves off which further makes them fall into a loop when the attackers try to harm the network. Where they also enforce the Sandbox environment and verification unit, which manages to recognize the honeypot networks. Similarly, in this application, any violation is typically reported either to an administrator or by using Security Information and Event Management (SIEM) system, where it combines the outputs from multiple sources and uses alarm filtering techniques to distinguish any intrusion from false alarms. Where in this,

a honeypot network which is attached in a system is implemented as a decoy to lure the cyber attackers and also to detect, or study hacking attempts in order to gain any unauthorized access to information systems. Where the project application is implemented in a local server which was developed using Java Programming language, client-side scripting in HTML, JavaScript and CSS and the database in MySQL. Where the software application needed for this project is Workbench Eclipse Kepler, server deployment in Tomcat 7.0, this project was implemented in Windows 10 OS with 8 GB of RAM, 1 TB of Hard Disk Space and GPU support.

5 Implementation

The architecture diagram for this process is in the below Figure 10, also the algorithm for this implementation process is based on the list below as the proposed system is followed by that algorithm.

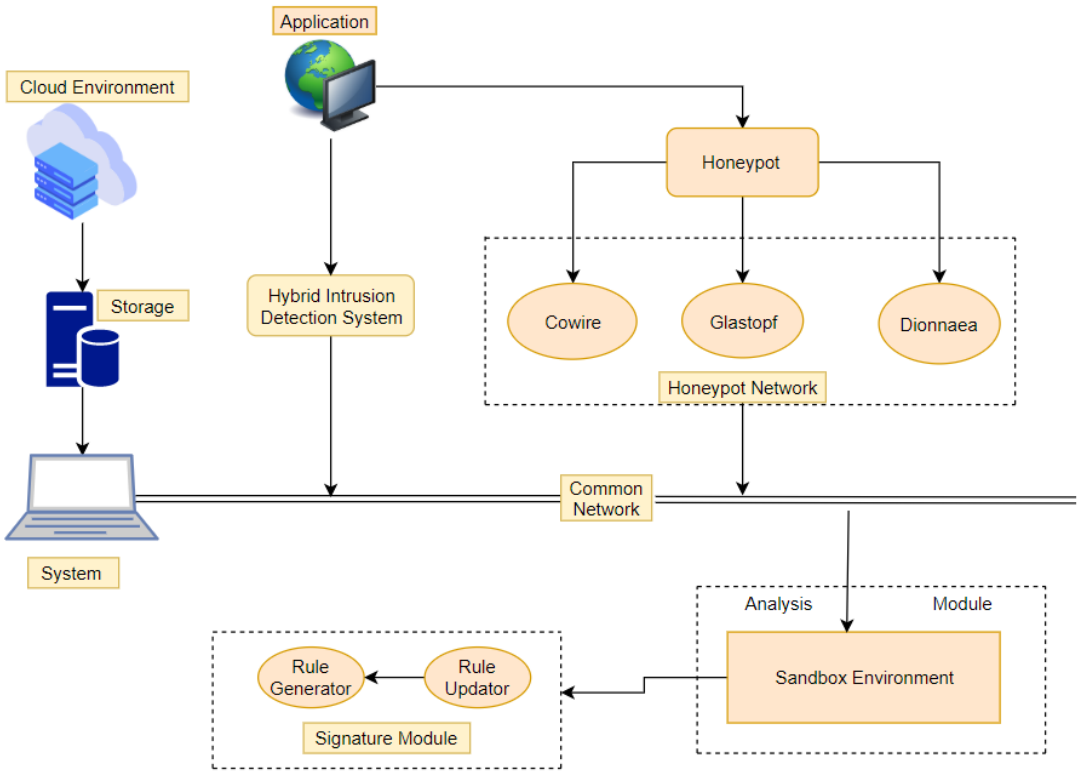

Figure 10: Architecture Diagram of Proposed System

The below algorithm from Figure 11 to Figure 14 is implemented using Java programming language where SQL server as back end. Considering the authentication phase as per the algorithm is implemented, where the nature of the cloud may vary. An N number of users will be generated, and user information will be stored in the database, they are denoted as registered users. Similarly, the application will automatically obtain the physical address of the system through which the authentication takes place throughout the registration process. The authorized client has access to the cloud for legal purposes

only through that specific system. If any request is received other than the registered user's MAC address, an intrusion alert would be sent to the administrator.

Algorithm 1: HIDS Algorithm
Input: Incoming Request
Output: Show all of the requested information and illustrate the unusual request

1. Start
2. {
3. Initialize the number of cloud users;
4. For all the "n" users of the cloud
5. {
6. Initialization and execution of Registration phase ();
7. Initialization and execution of Misuse_check_phase ();
8. Initialization and execution of Anamoly_check_phase ();
9. }
10. End

Figure 11: HIDS Algorithm

Algorithm 2: Registration Phase
Input: Username, Password, Mac Address
Output: Registering authenticated user

1. Start
2. For all the n users of the cloud;
3. {
4. Get valid e-mail-id and password 'eid', 'pid';
5. Auto fetch the Mac address of the system mac ();
6. If the value of email id eid is valid && password pid is valid and
7. If mac address mac is valid then the registration is successfully done;
8. Else
9. {
10. Stop the registration process;
11. Alert the admin for abnormal activity;
12. }}
13. End

Figure 12: Registration Phase Algorithm

Algorithm 3: Misue_Check_Phase
Input: Email-id, password and mac address
Output: Scan, display the requested details and highlight any abnormal activity

1. Start
2. {
3. Scan incoming requests;
4. Read email id eid, password pid and Mac address mac;
5. Read protocol pt and port number pb for incoming request;
6. If email-id, password and mac address is not valid then,
7. The request may be an intrusion alert;
8. Highlight the request and alert admin;
9. If protocol pt and port number pb is not known, then
10. The request may be an intrusion;
11. Highlight the request and alert admin;
12. If misuse_check _phase is success, then
13. Shift to next phase, Anomaly_check_phase;
14. Else
15. Alert admin to monitor the activities of network;
16. }
17. End

Figure 13: Misuse Check Phase

Algorithm 4: Anomaly_Check_Phase
Input: Incoming Request
Output: Shows the speed and bandwidth of the application

1. Start
2. {
3. Set threshold value of request speed rs
4. Set threshold value of bandwidth bw;
5. All incoming request “i”
6. Up to the value of “I” is 10
7. Scanning the speed of the request rs and displays it;
8. Scanning the bandwidth of the request and displays it;
9. If the request speed is less than or equal to threshold value of speed request, then
10. The request can be an intrusion;
11. Highlight the request in display;
12. Else
13. Check the bandwidth of the request speed;
14. If the bandwidth value of greater than threshold the bandwidth bw, then
15. Request can be an intrusion;
16. Highlighting the request in display;
17. Find the average value for request speed rs, then
18. Find the average value for bandwidth bw;
19. If the average request speed rs is greater than request threshold rt, then
20. Update threshold value rt with average value rs;
21. Else
22. Shift to bandwidth criteria;
23. If average bandwidth bw is less than bandwidth threshold bwt, then
24. Update threshold value bwt with average value bw;
25. Repeat for all incoming requests;
26. } End

Figure 14: Analysis Check Phase

6 Evaluation

The performance of a proposed intrusion detection framework was tested using the JMeter testing tool also another testing tool like Vega and Nmap. Similarly, with the help of this JMeter, the intrusion detection system has been fetched for 5 users, 10 users and 50 users. A system's overall performance is evaluated using factors such as estimated response time, the transmission of data throughput. Therefore, other parameters provided by JMeter, such as Median and JMeter's response time is in milliseconds, the below table represents the performance of Intrusion Detection of the users. The homepage of the intrusion detection system "login page" is that where both the user and administrator will sign in.

Performance for 5 Users

Label	Samples (Count)	Response Time(seconds)	Throughput(sec)	Data Transfer (KB/sec)
Home Page	5	0.750	4.2883304	7.411119
Login	5	1.561	2.4502058	6.854123
Admin	5	1.108	4.1044482	24.141
Total	15	0.370	7.6423061	24.457

Figure 15: Performance of 5 Users

Performance for 10 Users

Label	Samples (Count)	Response Time(seconds)	Throughput(sec)	Data Transfer (KB/sec)
Home Page	10	0.412	6.156615	10.421
Login	10	1.321	2.970431	7.4185
Admin	10	0.789	3.487248	20.101
Total	30	0.851	7.910298	26.198

Figure 16: Performance of 10 Users

Performance for 50 Users

Label	Samples (Count)	Response Time(seconds)	Throughput(sec)	Data Transfer (KB/sec)
Home Page	50	25.49	9.1089466	15.424
Login	50	80.75	2.8036936	6.9875
Admin	50	68.42	2.4963757	14.821
Total	150	58.45	7.2446655	24.129

Figure 17: Performance of 50 Users

The Figures above from Figure 15 to Figure 17 depicts the graphical representation of the response time, where it shows the fast response time and the performance does not become slow when the registration of users increases. Since before the chart generated from those in JMeter has legibility problems, the parameters obtained from the above tables are defined as a chart.

6.1 Performance for 5 Users

When JMeter is retrieved from the Index page from the Home Page with 5 virtual users, authentication page and administrator page the above table in figure.7 is obtained a result along with graph.

6.2 Performance for 10 Users

When JMeter is retrieved from the Index page from the Home page with 10 virtual users, authentication page and administrator page the above table in figure.8 is obtained a result along with graph.

6.3 Performance for 50 Users

When JMeter is retrieved from the Index page from the Home page with 50 virtual users, authentication page and administrator page from the above table in figure.9 is obtained a result along with graph.

6.4 Summary of Scanning

Similarly, we checked the scan report of the software tool Vega, which shows some of the technical faults for the corrections, where it shows the main drawback of SQL Injection it is the primary cause of the application. So, we need to make a prepared statement for the application to run the interface.

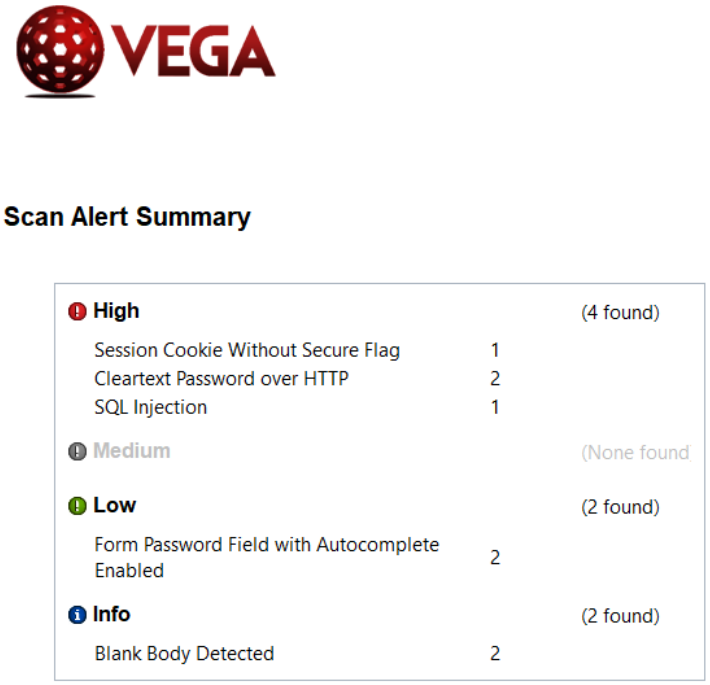

Figure 18: Scan Graph

6.5 Discussion

The registration and login time test were carried out to verify that the proposed plan was accurate and reliable than the existing method. Each client will have a MAC address in the past method, which would be a complicated task at the time of authentication and would be signed up in the database when the user is registered. Here the 5 users are registered, authorized to log in and the test was successful. It also estimates the activity of the response to this latency of the application to identify any unusual signal, which might be an intruder. Similarly, the suggested algorithm structure can be extended according to the complexity of the network in which the program is being implemented. The application built is essentially an internet-based platform executed in Java and HTML, the framework has Mysql server backend support, which includes a list of registered users. Various server execution work was done using AWS that are recognized attackers and identification based on their several interventions. New types of threats are not regarded due to the difficulty of duplicating patterns of attack. It is quite explicit that the suggested approach is successful because the algorithm can improve the quality of its function and then satisfy the complex nature of the Hybrid Intrusion Detection method.

7 Conclusion and Future Work

The proposed methodology can be designed for a heavily secure cloud environment, like the cloud that is being developed for defensive purposes and informative purposes to observe the network's actions expertly. The algorithm's performance in terms of computation and data consumption is stable. Similarly, it is also possible to implement the proposed algorithm using free software such as PHP, Python and can be deployed in the open-source clouds such as Open Stack, Cloud9.

The algorithm's performance can be increased by adding many more parameters to detect an Intruder alert in a network. Similarly, the algorithm's efficiency enhanced by changing the values based on the deadline. The output of the proposed algorithm will remain strong, even though it increases the number of users. The identification of the proposed algorithm anomaly intrusion alert may be further enhanced in future by alerting the user with an E-mail when an intruder tries to break the application, Similarly trap the intruder in a more complex honeypot and study further to prevent the attacks.

References

[1] Noor, T.H., Zeadally, S., Alfazi, A. and Sheng, Q.Z., 2018. Mobile cloud computing: Challenges and future research directions. Journal of Network and Computer Applications, 115, pp.70-85.

[2] S. Muhseen and A. Elameer, "A Review in Security Issues and Challenges on Mobile Cloud Computing (MCC)", 2018 1st Annual International Conference on Information and Sciences (AiCIS), 2018. Available: 10.1109/aicis.2018.00035

[3] Roman, R., Lopez, J. and Mambo, M., 2018. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. Future Generation Computer Systems, 78, pp.680-698.

[4] Gai, K., Qiu, M., Tao, L. and Zhu, Y., 2016. Intrusion detection techniques for mobile cloud computing in heterogeneous 5G. Security and Communication Networks, 9(16), pp.3049-3058

[5] M. Mollah, M. Azad and A. Vasilakos, "Security and privacy challenges in mobile cloud computing: Survey and way ahead", Journal of Network and Computer Applications, vol. 84, pp. 38-54, 2017. Available: 10.1016/j.jnca.2017.02.001

[6] Y. Mehmood, M. Shibli, U. Habiba and R. Masood, "Intrusion Detection System in Cloud Computing: Challenges and opportunities", 2013 2nd National Conference on Information Assurance (NCIA), 2013. Available: 10.1109/ncia.2013.6725325

[7] S. Alonso-Monsalve, F. García-Carballeira and A. Calderón, "A heterogeneous mobile cloud computing model for hybrid clouds", Future Generation Computer Systems, vol. 87, pp. 651-666, 2018. Available: 10.1016/j.future.2018.04.005

[8] AyeThu, A. (2013). Integrated Intrusion Detection and Prevention System with Honeypot on Cloud Computing Environment. International Journal of Computer Applications, 67(4), pp.9-13.

[9] Sahu, M. and Pandey, U. (2018). Mobile Cloud Computing: Issues and Challenges. 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN).

[10] Mishra, P., Pilli, E., Varadharajan, V. and Tupakula, U. (2017). Intrusion detection techniques in cloud environment: A survey. Journal of Network and Computer Applications, 77, pp.18-47.

[11] Chattopadhyay, N., Bhattacharya, S., Ghosh, R. and Paal, A. (2018). Data Intrusion Detection with basic Python coding and prevention of other intrusive manifestation by the use of intrusion application. 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON).

[12] Gjermundrød, H. and Dionysiou, I., 2015, December. CloudHoneyCY-An Integrated Honeypot Framework for Cloud Infrastructures. In 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC) (pp. 630-635). IEEE

[13] Beham, M., Vlad, M. and Reiser, H. (2013). Intrusion detection and honeypots in nested virtualization environments. 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN).

[14] Brown, S., Lam, R., Prasad, S., Ramasubramanian, S. and Slauson, J., 2012. Honeypots in the Cloud. University of Wisconsin-Madison.

[15] Gamlo, A., Zhang, N. and Bamasag, O. (2017). Mobile Cloud Computing: Security Analysis. 2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud).

[16] R. Kumar and D. Sharma, "HyINT: Signature-Anomaly Intrusion Detection System", 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2018. Available: 10.1109/icccnt.2018.8494088

[17] V. Mahajan and S. Peddoju, "Integration of network intrusion detection systems and honeypot networks for cloud security", 2017 International Conference on Computing, Communication and Automation (ICCCA), 2017. Available: 10.1109/ccaa.2017.8229911

[18] L. Dongxia and Z. Yongbo, "An Intrusion Detection System Based on Honeypot Technology", 2012 International Conference on Computer Science and Electronics Engineering, 2012. Available: 10.1109/iccsee.2012.158

[19] H. Suo, Z. Liu, J. Wan and K. Zhou, "Security and privacy in mobile cloud computing", 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), 2013. Available: 10.1109/iwcmc.2013.6583635

[20] El-Sofany, H. and Abou El-Seoud, S. (2019). A Novel Model for Securing Mobile-based Systems against DDoS Attacks in Cloud Computing Environment. International Journal of Interactive Mobile Technologies (iJIM), 13(01), p.85.

[21] D. Fraunholz, M. Zimmermann and H. Schotten, "An adaptive honeypot configuration, deployment and maintenance strategy", 2017 19th International Conference on Advanced Communication Technology (ICACT), 2017. Available: 10.23919/icact.2017.7890056

[22] H. Wafi, A. Fiade, N. Hakiem and R. Bahaweres, "Implementation of a modern security systems honeypot Honey Network on wireless networks", 2017 International Young Engineers Forum (YEF-ECE), 2017. Available: 10.1109/yef-ece.2017.7935647

[23] D. Chen and H. Zhao, "Data Security and Privacy Protection Issues in Cloud Computing", 2012 International Conference on Computer Science and Electronics Engineering, 2012. Available: 10.1109/iccsee.2012.193