A Combination Techniques of Intrusion Prevention and Detection for Cloud Computing

Sabah M. Alturfi¹, Dena Kadhim Muhsen², Mohammed A. Mohammed², Israa T. Aziz³, Mustafa Aljshamee⁴

¹University of Kerbala, Iraq
²University of Technology, Iraq
³University of Mosul, Iraq
⁴Al-Tuff university college, Iraq

sabah.m@uokerbala.edu.iq
dena.kadhum.2015@gmail.com
mabdallazez4@gmail.com
israa_aziz@uomosul.edu.iq
mustafa.aljshamee@altuff.edu.iq

Abstract. Cloud computing provides scalable, on-demand, and highly available computing services over the Internet to both the public and organizations on a pay-per-use basis. It provides a variety of services such as networking, storage space, and applications. The key issue for cloud computing is ensuring the confidentiality and privacy of cloud resources and data. Enticing the user to purchase cloud services requires their trust which cannot be guaranteed unless the infrastructure is effectively protected because attacks at this level will threaten the whole system. To this end, we propose the Integrated Intrusion Prevention and Detection System (IIPDS) to prevent and detect different types of attacks to the infrastructure level of the cloud system. The proposed design uses Trusted Third Party (TTP) services and SSL/TLS protocols as intrusion prevention methods to secure the communication between the cloud provider and the user. It also uses multiple Intrusion Detection Systems (IDS) distributed over multiple cloud regions. The IDS system is capable of detecting known and unknown attacks using anomaly and rule-based (signature) intrusion detection techniques. The simulation results proved the efficiency of the system in detecting a wide range of attacks with low false positive alerts and low computational overhead.

Keywords. IDS, IPS, Anomaly, Signature, Integrity, Cloudsim.
1. Introduction
Cloud computing is a modern technology which provides shared computational resources over the Internet to manage, retrieve and store data. It also provides online access to different applications to clients at a comparatively low cost. The standard definition of cloud computing by the National Institution of Standards and Technology is “a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” [1].

Cloud resources are supplied by a third party through the Internet which enables clients to realize a financial benefit as it eliminates the need to implement expensive infrastructure. Likewise, organizations and businesses can benefit financially from cloud resources by only utilizing these resources when required to minimize the costs of data management. Cloud computing features are versatility, consistency, solid assurance against system assaults, rapid elasticity, fast deployment, and fast recovery from disaster at a low cost [2].

The cloud computing architecture comprises three abstract layers, which are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [1]. All layers are independent of each other. The layer which needs to be particularly mindful of security is the base layer (IaaS) in light of the fact that threats on this layer influence the other layers too. Cloud computing is configured into four models: the public cloud where one organization owns the cloud infrastructure and sells its services to the public [3], private cloud where the cloud infrastructure is owned by a private organization and its services are available to its users only [3], hybrid cloud where two or more of the cloud models (public, private, and community) are joined together under same standards and proprietary technology to enable data and application portability, and lastly community cloud. This model is provisioned for exclusive usage by the clients of specific organizations which have shared concerns (e.g., policy, vision, and security requirements) [1].

As the cloud conveys its services remotely over the Internet, three key aspects of cloud security need to be protected from distinctive attacks, specifically the confidentiality, availability and integrity of the client’s data [4]. Although numerous traditional attacks can be countered, the significant challenge for information security is intrusion detection and prevention in cloud infrastructures. Even though a large number of research studies have been conducted and various methodologies implemented to protect sensitive data, various uncharted attacks still threaten the cloud environment.

2. Cloud Security Issues
Cloud computing technology encompasses many technologies such as load balancing, networks, memory management, and operating systems. Thus, many attacks can threaten these technologies and interrupt their normal services. Cloud providers are required to guarantee the safety of the cloud resources to ensure the clients’ trust is retained. Although several features of the cloud architecture provide some security benefits which can overcome many traditional attacks, there are still some uncharted attacks that need countermeasure approaches. These features are virtualization, data segmentation and redundancy, and centralization of security [3]. While the firewall provides front-end protection against external attacks depending on the configuration policy, it cannot detect internal attacks [5]. The common security threats that target cloud resources are as follows:

2.1. Insider Attacks
Legitimate users of some organizations inside the cloud may become malicious and misuse the cloud services. Insider attacks are executed by users who have a different level of access to the organization’s systems and data. Attackers may commit fraud or destroy another user’s data which may take a longer period of time if it executed by external users. Internal Denial-of-service (DoS) is one example of this type of attack that is considered a serious trust issue for cloud systems [6].

2.2. Flooding Attacks
In a flooding attack, attackers aim to consume cloud resources and corrupt normal cloud services. They flood the cloud network or the cloud infrastructure with numerous packets using many hacked hosts called zombies. As a result, processing resources (CPUs) and network bandwidth are consumed which makes the provision of cloud services to legitimate users unviable. Any anonymous attacker can exploit the feature of deploying cloud services through the Internet and request the VM to perform DoS or Distributed Denial-of-Service (DDoS) attacks. The types of these packets can be ICMP, TCP, UDP or a mixture [7].

2.3. Port Scanning
In this type of attack, attackers target cloud servers by learning the status of the service’s port. Attackers adjust certain parameters to discover if the destination port is open, closed or filtered. More than 6000 ports are used by TCP and UDP protocols to differentiate the type of service. A port scanning attack is executed by sending crafted packets to each port one at a time in sequence or randomly. This attack does not cause direct damage to the ports but it enables attackers to deploy these attacks through open ports. There are many port scanning techniques, such as TCP scanning, ACK scanning and more [5]. Moreover, attackers can use multiple machines to scan multiple ports at the same time to increase the opportunity of finding open ports. This technique is called a distributed port scan [8].

2.4. Virtual Machine or Hypervisor Attacks
In this type of attack, hackers gain control over the host’s OS through compromising a virtual machine (VM) or hypervisor. When they control the VM, they can steal or misuse the data and applications. Zero-day attack is an example of this type of attack where attackers target VMs to control them and exploit the applications’ vulnerabilities. Many virtual server-based websites are damaged by this attack [5].

2.5. Buffer Overflow
To decrease the resistance of cloud infrastructure to random code execution, the cross-site scripting attack is widely used to exploit the applications’ vulnerabilities. When attackers control the memory area (buffer) of applications directly or indirectly, a buffer overflow attack can be deployed. This attack is performed by increasing the size of the written code over the size of memory area (written code size > buffer size). Attackers can execute this attack on the VM when the memory bound checking is not robust enough to fit the data that is copied to the buffer for execution [9].

3. Related Work
Vieiri et al. [10] introduced a Grid and Cloud Computing Intrusion Detection System (GCCIDS). This proposed design has an event auditor that plays a major role in detecting attacks that network- and host-based detection systems cannot detect. It compares the user behaviour stored in the behaviour-based database with predefined attack data stored in the knowledge-based database. When an intrusion is detected, it generates an alert and sends it out to the other nodes using middleware called Grid-M. Furthermore, to train the data and detect deviations according to the behaviour (normal or abnormal), a retropropogation algorithm is applied which is the feed-forward artificial neural network. This method has a high detection level with a low processing overhead; and it reduces the size and complexity of the captured data compared to previous techniques due to the use of many nodes. On the other hand, this model is used to detect specific types of attacks that focus on middleware layer and ignore attacks on infrastructure and software layer. Moreover, it is used for grid and cloud technologies which are different from each other. In addition, it only provides IDS.

Lo et al. [11] introduced a framework of IDS to detect DoS and DDoS attacks through the distribution of separate IDS over the cloud regions. Each IDS includes three models, cooperative, communication, and block models. Moreover, the proposed framework classifies security into three levels, slight, moderate, and serious levels, according to the matching case. When attack is detected, an alert is generated and this alert is sent to the other regions. To prove the truthfulness of the alert and to determine
whether to accept this alert or not, the majority voting process is deployed by the cooperative agent. Thus, the packets are dropped or blocked if the received alert is either serious or moderate. This framework uses the modified Snort which is differs from pure Snort by avoiding a single point of failure. Oppositely, the computational overhead is high compared to the pure Snort technique. Also, the modified Snort relates to a cooperative agent, so it cannot detect unknown attacks, similar to pure Snort. Moreover, it only provides IDS.

Dhage et al. [12] introduced distributed IDS architecture that positions mini IDS between the users and the cloud provider. This method provides an IDS instance to each user and these mini IDSs are controlled by one IDS controller. It detects suspicious patterns by deploying a set of rules that compares the normal patterns that are stored in the cloud’s log database with the user activity patterns. If intrusion patterns are detected, they are reported to the IDS controller for advanced action. This model can detects a wide variety of attacks in the cloud system due to the use of the two intrusion detection techniques, anomaly and signature-based techniques. On the other hand, the model can suffer from a single point of failure due to the use of one IDS controller. Also, it only provides IDS.

Zissis et al. [3] introduced Trusted Third Party (TTP) services to ensure confidentiality, integrity and authentication in the cloud computing environment. It calls upon the Public Key Infrastructure (PKI) which cooperates with the Single-Sign-On (SSO) mechanism and Lightweight Directory Access Protocol (LDAP). The TTP entity is operated by a third party, such as a business organization, to secure the channel of the communicating parties in the cloud environment, these being the user and the cloud provider, who both trust this third party. This third party is connected to a trust chain to review all critical transactions between the user and cloud provider. The trusted chain involves using the benefit of coupling PKI with the LDAP protocol to access the directory services in order to issue the certificates. This method provides a strong authentication and authorization process by gaining the user’s trust in the cloud’s services due to the mutual authentication mechanism. Oppositely, this method provides IPS only and it cannot detect internal attacks.

Kholdiy et al. [13] introduced the Cloud Intrusion Detection System (CIDS) that includes two IDS, Cloud-based IDS (CIDS) and Host-based IDS (HiDS). This point-to-point (P2P) architecture assists in improving the load balance between all the nodes in the system due to the absence of a central coordinator. This system uses the data stored in a behaviour-based database to detect abnormal behaviour in each node through correlating the user profile. In addition, CIDS uses the Snort technique to analyse network packets in order to detect signature attacks. The system generates alerts when intrusions are detected, then both the analyser and log summarizer analyse and summarise these alerts and send them to the cloud administrator for further action. This hybrid method provides a flexible solution for attack detection due to the P2P architecture. On the other hand, it cannot detect a wide range of distributed attacks, such as DDoS, with high false positive alerts due to the absence of a central correlation handler. Also, it provides IDS only.

4. The Proposed Design (IIPDS)

According to the security methods discussed above, the limitations of them mostly affect the performance of the cloud security system. To this end, we propose the Integrated Intrusion Prevention and Detection System (IIPDS) which aims to enhance the performance of the cloud security system by supporting the three key characteristics of the cloud which are confidentiality, integrity, and availability. It also aims to prevent hackers and insider users from deploying different types of attacks. Also, if an attack is launched, the proposed IIPDS aims to detect it and block the attacker and then informs the other regions via the following mechanisms:

1- Using a Trusted Third Party (TTP) [14] as a prevention method that ensures the confidentiality and integrity of the cloud data because the cloud data and services can be provided to the communicating parties so that their identities are verified by the trusted third party.

2- Using SSL/TLS protocols as a prevention method. These authentication protocols provide strong cryptography methods that eliminate external attacks effectively and enable authorized entities to access the cloud resource [15] [16].
3- Using anomaly and signature detection systems to ensure the availability of the cloud resources by providing 24/7 data access to cloud users as well as ensuring the integrity and confidentiality of the cloud data. These detection systems can detect internal users who misuse or compromise the cloud data. The proposed IIPDS consists of multiple IDSs distributed over multiple cloud regions to prevent any compromise attempt on the infrastructure layer (IaaS) of the cloud system. Each IDS consists of three components, detection module, storage (anomaly database and signature database) and alert handler. There are two components to the detection module: two anomaly analysers and one rule-based analyser (Snort). The incoming traffic is firstly checked by the firewall to prevent any malicious attacks defined in its policy and then it is forwarded to the two anomaly analysers. The reason for using two anomaly analysers is to provide a fault tolerant mechanism (redundant equipment) in order to protect the cloud system from suffering a single point of failure. The anomaly analyser monitors the traffic in the non-attack period and determines the threshold value of specific characteristics in a specific time window and stores this value in the anomaly database. After this, all traffic is forwarded to Snort to apply the set of rules stored in the signature database on the packets’ patterns. The aim of applying this algorithm is to detect possible intrusions and generate alerts and then to send these alerts out to the alert handler. The reason for using one rule-based analyser is that it can handle a normal traffic attack whereas the two anomaly analysers are used in a heavy attack (distributed). In the attack period, the anomaly analyser retrieves the values stored in the anomaly database to compare them with the incoming traffic. If the divergence is large, it generates an alert and sends it out to the alert handler and the packets are dropped. Next, the alert handler proves the truthfulness of the alert to reduce false positive alerts, and then informs other regions in order to avoid the same type of intrusion. A single and multiple cloud regions are shown in the below figures:

![Figure 1. Single Cloud Computing (IIPDS).](image1)

![Figure 2. Distributed Cloud Computing IIPDS.](image2)

The algorithm used in IIPDS is:

If a two-way authentication process is achieved then
Start data encryption
Transfer data
Data decryption at destination
Start anomaly detection
If anomaly attack detected then
Block attack
Alert other regions with attacker’s ID
Else
Start signature-based detection
If data pattern matches signatures stored in signature database then
Block attack
Alert other regions with attacker’s ID
Else
Process the request and respond to the sender
Else
Access denied.

4.1. Prevention System
The first component of the proposed IIPDS is a prevention system which uses two methods:
1- A Trusted Third Party (TTP) service to issue the proper authentication method[14]. Its provided by a third party who is trusted by the two communicating parties (i.e. user and cloud service provider). The third party reviews the critical communication between the cloud user and the cloud service provider and identifies any security threats. This service is connected through chains of trust and it issues a digital certificate to the known website to each participating server. The third party issues digital certificates and a public key to identify the cloud website. It works along with the Public Key Infrastructure (PKI) mechanism [3] [17].
2- A Transport Layer Security (TLS) with Secure Shell Layer (SSL) protocols as an encryption method to form a strong shell which protects the transmitted information over the network. It prevents attacks from being successful by setting suitable policies to match real attack types [15][16].

4.2. Detection System
It is important to have a complete detection system within each cloud region, so the proposed IIPDS detects a wide variety of attacks types. Two intrusion detection techniques are used in IIPDS to improve the integrity, confidentiality and availability of the cloud system, the rule-based (signature) and the anomaly intrusion detection techniques.
Snort, the rule-based network detection technique in IIPDS, is used to protect the cloud system from known attacks while the anomaly intrusion detection technique is used to find any abnormal behaviour that may be deployed by attackers on the cloud system. The signatures component is a set of rules to identify whether specific data or a particular event is good or bad. The rule-based (signature) intrusion detection method matches the network traffic with predefined signatures stored in the data base to detect a possible attack. These signatures are defined based on well-known attack strategies launched by intruders. For example, if a data packet contains a string greater than 1000 bytes and is filled with specific characters, the system will identify an attack of this type [18].

On the other hand, anomaly intrusion detection means that the IDS system has knowledge about what is a normal or abnormal system activity (or it has already been stored in the behaviour data base by the system administrator). It uses this knowledge to distinguish between normal and abnormal network traffic or system behaviour to block anomalies if they occur. These two techniques provide high detection performance and also maintain the computational overhead over the cloud regions.
Hence, IDS should be trained first to determine the normal characteristics of network traffic. Then these statistics are used to distinguish between normal and attacked signatures in the rule-based detection system and the optimal threshold for the parameters of the anomaly intrusion detection system.

4.2.1. Anomaly Intrusion Detection
Attackers adjust certain parameters or network conditions to launch their attacks with the intention to compromise cloud services. For instance, attackers who perform a port scanning attack send multiple SYN-flagged TCP requests with very low delay time to the target port and then wait for their response to check if it is open or not. If they receive (SYN, ACK) this means this port is open, if they receive (RST, ACK) this means that this port is closed [18].

Therefore, anomaly intrusion detection in IIPDS provides high availability and guarantees the integrity and confidentiality of the cloud service which is different from the methods mentioned in the previous studies. Usually, the network traffic should be trained first such as packet context, packet size, and more, to store the optimal threshold in the behaviour database. In anomaly intrusion detection, the system needs to monitor traffic behaviour, so if any inconsistent activity is observed, it generates an alert. Similar to the rule-based intrusion detection system, anomaly intrusion detection uses the behaviour database to store different types of optimal threshold values and retrieves them during the detection process to find any possible anomalies. Many parameters can be chosen to detect anomalies such as CPU and memory usage, packet size, network bandwidth, etc.

In IIPDS’s simulation, three parameters are monitored, the source IP address and the difference between the sending and receiving time interval which are extracted from the host packet and the number of packets transmitted in a specific time window. Firstly, the system checks the number of packets that are sent from the host in a specific time window. If it is greater than the optimal threshold, it checks the source IP address to find if they were sent by same host. At the same time, it checks the difference between the sending and receiving time. If the system finds that the number of packets in a specific time window exceeds the optimal threshold, it generates an alert and blocks this user. Although this process may consume processing power, the IIPDS design overcomes this weakness by using redundant components. This proposed design will reduce the time required for processing the network traffic and it protects the system from failure.

4.2.2. Rule-Based (Signatures) Intrusion Detection (Snort)
Snort is chosen to monitor the cloud’s network traffic for the following reasons:

1- It is an open source intrusion prevention and detection system that is capable of analysing real-time network traffic as well as logging packets. It enables programmers to specify and update different rules accurately according to the system requirements and the traffic conditions.

2- It is a flexible network detection system as it has a wide range of rules that inspect diverse cloud threats. For example, it has rule sets for malware-backdoor, malware-cnc, app-detect, file-multimedia, file-flash, file-pdf, blacklist, OS-windows and many others.

Snort captures incoming network packets and checks to find out if there are any intrusion patterns that might be used. The work of Snort in detecting known attacks with anomaly intrusion detection provides high security for cloud systems [18].

4.3. Alert Handler
It is known that the rule-based IDS detects low level alerts with high false positives, then, the alert handler minimizes the number of false positive alerts, especially alerts generated by Snort. It aims to group alerts into clusters and then identifies the alerts that are of interest. When an attack is detected, IIPDS generates an alert and then sends it to the alert handler to prove the truthfulness of this alert.

The IIPDS alert handler has the capability to group several alerts (five alerts) into one cluster. These alerts contain the following information (Source IP address, Receiving time, Alert generator source (i.e. Snort or anomaly detector), Attack type, Packet size).
The alert handler checks each piece of information and compares them within the group. It checks the attack type first, then it looks to find any similarities among them. If similarities are found, it generates an alert and sends it out to the other cloud regions. Moreover, in some cases, for some attack types, the alert handler generates an alert immediately without finding similarities due to the priority level. For instance, when the attack type is a DDoS, it ignores the comparison process and generates an alert to notify other cloud regions.

To reduce the processing overhead for the alert handler, priorities are attributed to the incoming alerts. Therefore, any attack with a high priority generates an alert immediately to pass it to other regions, as illustrated in the table below:

| Attack type                        | Priority |
|------------------------------------|----------|
| DoS or DDoS                        | H (high) |
| Data availability and integrity    | H        |
| attacks                            |          |
| Infrastructure attacks              | H        |
| Data confidentiality attack         | M (Moderate) |
| Confidentiality breach             | L (Low)  |

In the proposed design, the threshold is set to 3, so in the case of a password sniffer attack, for example, if the attacker repeats this attack three times within one group, the priority of the attack type changes from moderate to high.

The alert handler takes responsibility for notifying other cloud regions about this type of attack and specifies the source of attack in order to block it and prevent the user from further accessing cloud servers. The system administrator can modify and change the alert information that needs to be transmitted to other servers. This alert information includes the information needed by other servers about the type of attack and enriches their databases with new detected attacks.

### 5. IIPDS Simulation Implementation

CloudSim V3.0.3 is used to simulate the proposed design. It was developed in the cloud laboratory in a computer science and software engineering department at Melbourne university. It is free and open source, developed for academic purposes and its library is coded using the Java programming language. It is not GUI software and is integrated with Java-based Integrated Development Environments (IDEs) such as Eclips and NetBeans. In the proposed system, Eclips IDE Luna Service release 1a (4.4.1) is used. In addition, to complete and execute this code, the code should integrate with Java Development Kit (JDK) software [19].

The simulation process of IIPDS created the following entities randomly using the Math.random() function within the math class of the Java library. This function generates a real number between (0 and 0.9).

1. Power datacenters (1-10 datacenters).
2. Virtual machines (VMs) (1-1000).

Firstly, several datacenters representing cloud regions are created and started. Then the datacenter broker which acts on behalf of the user in managing VMs is started. Later, a number of VMs are created on the host within the datacenter and the host executes all the actions associated with VM management, such as creation and destruction.

IPS uses the TTP authentication service to prove the identity of the two communicating parties through using a trusted third party. This part is not implemented in this design because it requires real communicating parties (client and cloud provider). IPS starts by encrypting the packets on the source VM and decrypts them when they arrive at the datacenter to prevent certain attacks such as the man-in-the-middle attack, the password sniffing attack, and different data integrity attacks like data theft and data modification and more. It is important to choose the best encryption mechanism which should include a onetime key generation algorithm to provide a high level of transmission security, so the SSL/TLS encryption method is recommended, as discussed in the previous chapter. Due to time
limitations, we implement a Caesar encryption and decryption method which can give us an acceptable security level in cooperation with IDS.

After all the entities of IIPDS are created and started, two scenarios are implemented to detect different types of attacks and both were run 1000 times. Both scenarios utilize the IPS system to prevent possible attacks that are usually launched to steal a legitimate user’s credentials and access the cloud resources as an authorized user.

5.1. Implementation Scenarios

5.1.1. Scenario 1
The aim of scenario 1 is to detect some attack types deployed by one VM and to calculate the optimal threshold value, using one VM to send its request in the form of packets to the datacenter. The work of IDS can be described in two steps:

1- Anomaly detection: a type of attack considered to be a DoS occurs when the number of packets exceeds a predefined threshold; the packets are from the same source VM; and there is a difference in the sending and receiving time interval. When this occurs, an alert message is generated and passed to the other datacenters to block the source VM. The process of generating packets and the work of the anomaly IDS is illustrated as follows: One randomly chosen VM starts sending a random number of packets (from 1-5000 packets). The predefined threshold that represents the number of packets which will generate an attack alerted is changed five times (1000, 2000, 3000, 4000 and 5000 packets). For all these thresholds, if the number of packets is greater than the threshold, it checks the source VM to discover if they have been sent from same VM and it also checks the sending and receiving time difference to discover if it greater than 0.3 seconds.

2- If the packets display normal behaviour, they are checked by the second component to determine if it is a signature attack. In the proposed IIPDS, packet data is represented as a string and this string is fixed to specific characters except the last character, for example, “FMDNS_” where the last character is generated randomly. The signature detector calculates the probability of finding signatures that differ in content length. This probability is also changed five times (90%, 80%, 70%, 60% and 50%). In the database file, several strings are stored and one of them is similar to the packet data but the last character is specified. If the packet data matches one string stored in database, it fires a rule. Then it generates an alert message and passes it to the other datacenters and blocks the source VM.

5.1.2. Scenario 2
In this scenario, the simulation is run to detect a DDoS attack which is launched by multiple zombies which have been hacked by attackers to affect the cloud services and exhaust its resources. The simulation uses multiple VMs that send packets to the associated datacenter. The work of the IDS in this scenario differs from scenario one in the following three ways: firstly, the attack type is considered to be a DDoS in addition to a signature attack. Secondly, the system counts the number of packets for each VM and compares it to the thresholds mentioned in the previous scenario. Thirdly, the difference between the sending and receiving time is changed to 0.5 seconds.

5.2. Simulation Results
The simulation is run 1000 times for each scenario and the threshold value regarding the number of packets is changed five times (each threshold value is simulated 200 times).

5.2.1. Scenario 1 Results
Figure 3 shows the simulation results for scenario 1. As shown in the graph, when the threshold increases, the number of attacks decreases sharply. Also, the number of detected anomaly attacks is greater than the number of detected signature attacks.
Print screen 1 shows the simulation result for the normal operation of the cloud system. Even though the number of packets is larger than the threshold, the alert is not generated because the detection of anomalies depends on the second parameter which is the difference between the sending and receiving time. This time difference is specified to be 0.3 seconds.

Print screen 2 shows the detection of an anomaly attack of the proposed design where the number of packets sent by one VM is greater than the detection threshold, whereby an alert is generated and transmitted to the other datacenters after its truthfulness has been proven in order to destroy the source VM. In the figure below, the number of packet sent by VM1 is 5603 which is greater than the threshold which is 1000 packets.
Print screen 2. Anomaly Attack Detection.

Print screen 3 shows the detection of a signature attack where the number of packets is very small compared to the threshold. An alert is generated due to the fact that the probability of the packet data having the same signature content is greater than the threshold probability of the signature detection system. The content of the packet data is compared with the signatures stored in the signature database.

Print screen 3.

Signature Attack Detection.

Print screen 4 shows the probability of the signature matching that was adjusted to reduce the false positive alerts. In this simulation, it was 80%, so when the probability of matching for VM number 613 exceeds this percentage, the detection system fires the rule and an alert is generated. In this scenario, the probability of signature matching was adjusted five times (90%, 80%, 70%, 60% and 50%) to determine the optimal threshold.

Print screen 4. Probability of Signature Matching.

5.2.2. Scenario 2 Results
Figure 4 shows the simulation results for scenario 2. The major function of this scenario is to prove the ability of the system to detect a DDoS attack. The performance of this scenario is the same as scenario 1. When the threshold increases, the number of attacks decreases sharply. Also, the number of detected anomaly attacks is greater than the number of detected signature attacks which is launched one time only.

![Figure 4. Simulation Results for Scenario 2](image)

Print screen 5 shows the normal operation of the cloud system. Although the number of packets is 5741 which is much higher than the threshold of 2000, an alert is not generated because the difference between the sending and receiving time is greater than 0.5 seconds and also these packets have been distributed evenly between the host VMs.

![Print screen 5. Normal Operation of the Cloud System.](image)

Print screen 6 shows an anomaly attack detection launched by multiple VMs. This DDoS attack attempts to interrupt the normal operation of the cloud system by exhausting its resources. As shown, the number of packets that is sent by multiple VMs is 2991 and the threshold is 2000. In this case, an alert is generated as the time difference between sending and receiving the packets is less than the threshold of 0.5 seconds.

![Print screen 6. Anomaly Attack Detection.](image)

Print screen 7 shows the signature attack detection that was launched by VM number 215. The system was able to detect the signature attack by calculating the probability of the signature existing by
comparing the packet data content with the signatures stored in the signature database. It is important to adjust the threshold of probability carefully to reduce the false positive alerts.

Print screen 7. Signature Attack Detection.

Print screen 8 shows the probability of signature matching, which was adjusted five times (90%, 80%, 70%, 60% and 50%) to determine the optimum threshold to reduce the false positive alerts. In this simulation, the probability of matching for VM number 565 was 70%. When this percentage was exceeded, the detection system fired the rule and generated the alert.

Print screen 8. Probability of Signature Matching.

5.3. Results Discussion

In this simulation, three parameters are taken into consideration to discover abnormal behaviour of the network packets that are included in the host packet class of CloudSim. The first parameter is the source VM which is equivalent to the source IP address and the second parameter is the internal receiving time which is calculated as the difference between the sending and receiving times. The internal receiving time is equivalent to the time difference between two packets received by the cloud resources which is normally in milliseconds. The third parameter is the number of packets received from the source VM which is equivalent to the number of packets sent by the source machine to the target cloud resource. In a real cloud environment, the cloud administrator has a variety of parameters from which to choose such as packet header size, source port, protocol type, sequence number and more.

The simulation results show the efficiency of the attack prevention and detection of the proposed IIPDS for the cloud system. The cloud administrator must carefully choose the parameters to monitor for different attacks, depending on the historical data stored in the database and the new types of attacks detected by the system. In this design, anomaly attack detection relies on three parameters, source IP address, number of packets, and receiving time interval. The threshold was adjusted five times to discover the optimum threshold to maximize system performance. The results show that when the
threshold is small, the number of false positive alerts increases, whereas the false negative alerts increase sharply when the threshold value is set too high. Therefore, adjusting the threshold depends on the nature of the traffic and results in high system performance. Also, the results show that the number of detected anomalies is much larger than the number of detected signatures. Similar to the anomaly detection system, the adjustment of signature probability plays a major role in detecting signature attacks. Snort uses the signature database to match the content and fire the rules. So, if the database is not updated regularly, the system will miss some new types of signature attacks. To sum up, choosing the parameters carefully as well as enriching the signature database with up-to-date signatures will increase the detection of anomaly attacks.

6. Conclusion

Cloud computing is an emerging technology that provides on-demand services to both the public and organizations. Users benefit from these virtualized resources by minimizing their cost and effort. It uses remote servers to store, manage, and process data over the Internet. Due to the sharing of resources (multitenancy), the cloud environment faces many threats deployed by cyber criminals to hack the data and interrupt the cloud services. These attacks target different cloud parts, for example, SQL Injection and Cross-Site Scripting (SSX) attacks target cloud websites and the man-in-the-middle attacks, network sniffing, network flooding and DNS attacks are examples of network attacks.

As a result, different security mechanisms are designed to prevent such attacks. One of these mechanisms is using virtualization in the cloud which provides a better isolation level than a traditional OS. Furthermore, two techniques are used to protect cloud resources, the Intrusion Prevention system (IPS) which prevents possible attacks in advance and the Intrusion Detection system (IDS) which detect attacks and blocks them.

Thus, we proposed IIPDS as a complete protection system for cloud computing. IPS employs two techniques, TTP and SSL/TLS protocols. TTP uses a third party who is trusted by the user and cloud provider to issue security certificates to verify the mutual identity of the communicating parties. SSL/TLS protocols are used as main authentication technique of the cloud system. They secure the communication by providing a secure cryptographic method. IDS in the proposed design uses two techniques to detect intrusions. The first technique is anomaly intrusion detection which aims to detect abnormal behaviour of the network traffic. Different parameters need to be monitored by the proposed design to detect threats such as the number of packets, packet size, the type of protocol, etc. In our simulation for IIPDS, three parameters are specified, source IP address, the difference between sending and receiving time, and the number of packets. When the value of a parameter exceeds the optimal threshold value, an alert is generated and is transferred to the alert handler. The alert handler is used to reduce the false positive alerts by checking some information to prove the truthfulness of the alert.

The second technique of IDS is using Snort for rule-based intrusion detection. Snort is flexible open source software used to detect different types of attacks, such as the port scanning attack and more. Snort analyses network traffic after having been checked by the anomaly detection technique. If the probability of matching the packet data with the signatures stored in the signature database satisfies a predefined probability, it fires a rule and generates an alert and passes this to the alert handler.

Our experimental results demonstrate the efficiency and feasibility of our proposed IIPDS for the following reasons. Firstly, it prevents a single point of failure due to the redundancy of components. Moreover, it reduces the processing overhead due to the distributed system and the use of two techniques, IPS and IDS, with two sub-techniques for each. Lastly, it has the capability of detecting and blocking a wide range of attacks. Also, this proposed design shows the optimum threshold value which reduces the number of false positive alerts when it is adjusted carefully.

References

[1] Y. Jadeja and K. Modi, "Cloud Computing - Concepts, Architecture and Challenges," in International Conference on Computing, Electronics and Electrical Technologies (ICCEET), Kumaracoil, 2012.
[2] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," Journal of Network and Computer Applications, vol. 34, no. 1, pp. 1-11, January 2011.

[3] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," Future Generation Computer Systems, vol. 28, no. 3, p. 583–592, March 2012.

[4] C. Modi, D. Patel, B. Borisaniya, A. Patel and M. Rajarajan, "A survey on security issues and solutions at different layers of Cloud computing," J Supercomput, pp. 561-592, 13 October 2012.

[5] C. Modi, D. Patel, H. Patel, B. Borisaniya, A. Patel and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 42-57, 2013.

[6] Z. M. Yusopa and J. Abawajyb, "Analysis of Insiders Attack Mitigation Strategies," in International Conference on Innovation, Management and Technology Research., Malaysia, 2013.

[7] E. P. Krishna and T. Siva, "Controlling various network based ADoS Attacks in cloud computing environment: By Using Port Hopping Technique," International Journal of Engineering Trends and Technology (IJETT), pp. 2099-2104, May 2013.

[8] H. Mohamed, L. Adil, T. Saida and M. Hicham, "A collaborative intrusion detection and Prevention System in Cloud Computing," in AFRICON, Pointe-Aux-Piments, 2013.

[9] A. Kundu and E. Bertino, "A New Class of Buffer Overflow Attacks," in International Conference on Distributed Computing Systems, Minneapolis, MN, 2011.

[10] K. Vieira, A. Schuler, C. B. Westphall and C. M. Westphall, "Intrusion Detection for Grid and Cloud Computing," IT Professional Magazine, vol. 12, no. 4, pp. 38-43, July-August 2010.

[11] C.-C. Lo, C.-C. Huang and J. Ku, "A Cooperative Intrusion Detection System Framework for Cloud Computing Networks," in International Conference on Parallel Processing Workshops (ICPPW), San Diego, CA, 2010.

[12] S. N. Dhage, B. B. Meshram and R. Rawat, "Intrusion Detection System in Cloud Computing Environment," in ICWET '11 International Conference & Workshop on Emerging Trends in Technology, Mumbai - India, 2011.

[13] H. A. Kholidy and F. Baiardi, "CIDS: A framework for Intrusion Detection in Cloud Systems," in Ninth International Conference on Information Technology- New Generations, Las Vegas, NV, 2012.

[14] F. C. Chong, "Trusted third party authentication for web services". United States Patent US 7,900,247 B2, 1 Mar 2011.

[15] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol.," http://tools.ietf.org/html/rfc5246, 2008.

[16] A. Freier, P. Karlton and P. Kocher, "The Secure Sockets Layer (SSL) Protocol Version 3.0," https://www.rfc-editor.org/rfc/rfc6101.txt.pdf, 2011.

[17] R. S. NEJKAR and G. A. PATIL, "TRUSTED THIRD PARTY SERVICE FOR SECURE CLOUD DATA," International Journal of Research in, vol. 2, no. 6, pp. 203-211, 2014.

[18] B. Caswell and J. Beale, Snort 2.1 Intrusion Detection, Second Edition, USA: Syngress, 2004.

[19] S. K. Garg and R. Buyya,"NetworkCloudSim: Modelling Parallel Applications in Cloud Simulations," in Fourth IEEE International Conference on Utility and Cloud Computing, Victoria, NSW, 2011.