Journal of Theoretical and Applied Information Technology
15th October 2019. Vol.97. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CLOUD COMPUTING BASED ATTACKS AND COUNTERMEASURES: A SURVEY

LAMYAA S. ALSALEEM[1], SARAH A. ALQAHTANI[2], SARAH F. ALHARBI[3], RACHID AGROUBA[4].
College of Computer Science and Information Technology
Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
2150006354@iau.edu.sa[1], 2150001193@iau.edu.sa[2], 2150004612@iau.edu.sa[3] , rmzagrouba@iau.edu.sa[4]

ABSTRACT
Nowadays, cloud computing and its related security issues are one of the most debated topics in today's research field. Cloud computing raises the efficiency and proposes many advantages to users, but at the same time it is still a new technology that needs a lot of enhancement in term of security. This survey presents cloud service delivery models, deployments and characteristics. Furthermore, it gives a detailed explanation on known attacks that threaten the cloud core components and how it might occur in cloud systems and discusses possible solutions to mitigate these attacks. Lastly, it summarizes the attacks and compares between the discussed solutions.

Keywords—*Cloud Computing, Attacks, Security, Network, Virtualization, Storage, Countermeasure.*

1. INTRODUCTION

Recently, cloud computing has gained a wide recognition due to the customizable services it provides. It addresses resources shortage, and promises its customers with scalable, on-demand and "pay as you go" services [1]. Not only that, it allows access from anywhere, anytime which enable users to work remotely, such features provide an enormous help for various fields, especially where employees need to work remotely and collaboratively [2]. In addition, its storage capability promises a significant advantage as compared to traditional storage medium, in terms of cost and quantity of data it holds [1]. These services are expected to be secured and protected from any type of internal or external threats. At the same time, the variety of cloud services makes the cloud computing vulnerable to many security attacks, that might target one of the cloud's core components which are: storage, application, network and virtual machines [2]. These components must be carefully protected to ensure data confidentiality, integrity and availability and that the cloud delivers its services as expected.

Thus, to maintain high security of customers' data and to reach out with various services in cloud computing, addressing the cloud core components' issues is mandatory.

Accordingly, this research aims to highlight network, virtualization and data storage attacks and compares between
its solutions to enhance the cloud security.

2. BACKGROUND

2.1 Cloud Service Delivery Models:

Cloud computing services and IT resources can be delivered to customers using many models, which are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS)

- *SaaS*: Software as a Service as the name explains, is a software offered by the provider as a service on the cloud, so that end users can access this software using the internet [3]. End Users do not need to install the software on their devices, all they need is a good internet connection to access the software service remotely [4]. In SaaS, users have no knowledge about the underlying infrastructure since it is completely handled by the cloud provider [4]. An example of software as a service is Google Apps, where users can use Google docs online without the need of installing the applications on their devices [3].
- *PaaS*: Platform as a Service can be explained as a computing environment that is equipped with tools to facilitate software development. Meaning, in SaaS, users use the software available by the cloud provider, while in PaaS, users use the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

equipped environment available by the cloud provider to develop, run and manage their own applications [4]. PaaS is so helpful when many developers located in various physical locations work together on the same development project. The underlying infrastructure such as network, databases and servers are taken care of by the provider, users are only required to choose their preferable platform [3]. In comparison to SaaS, in PaaS users have wider range of control since they are able to modify the environment setting and configuration, and they can have some control on their deployed application [4]. An example of PaaS is AWS Elastic Beanstalk, where users need to access PaaS cloud service and select the platform (Java, ASP.NET, etc.) Then start working.

- *IaaS*: Infrastructure as a Service deals with providing the computing resources, such as memory, network storage, processors, data center and virtual machines as a service [3]. In terms of control, customers have the widest control in comparison to other models, since the costumer controls the operating system used for each instance and the running services, policies and some network configuration [4]. An example of IaaS is AWS EC2 where customers need to access EC2 service then launch as many instances (virtual machines) as needed, each with the customer's preferable operating system and setting. Further, customers are responsible of creating security groups, configuring the network ports and IPs to access these instances, allocating the memory and processor for each instance and many others.

Figure 1 summarizes service delivery models, the dark blue portions are the layers handled by the provider while the user is not aware of them, the green portions are the layers where the user has some control.

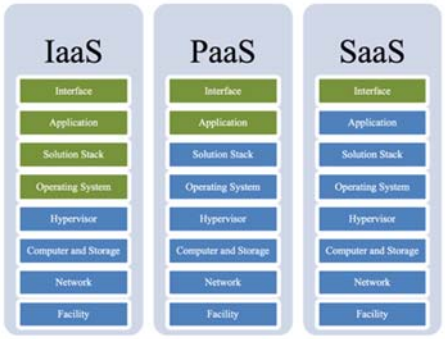

Figure 1 Cloud Service Delivery Models [3]

2.2 Cloud Deployment Models:

The National Institution of Standards and Technology (NIST) categorizes the cloud deployment models into four categories, which are: Public Cloud, Private Cloud, Community Cloud and Hybrid Cloud [5].

- *Public cloud*: Cloud computing resources are operated by cloud service provider [6] and are available for public, meaning, anyone can use the virtual machines, storage and other resources hosted and offered by the service provider [7]. The same computing resource is accessed by different customers with the help of virtualization and multi-tenancy, but that does not mean they can access each other's data, or that the data is available for public, since service providers implement access control and authentication methods [7].
- *Private cloud*: Cloud computing resources are operated and exclusively used by a single organization [4] [6] and are not available for public. The resources might be operated by a third party, but only single organization can consume the provided service [5]. The hardware required to provide the cloud computing services might be in the organization's data center or hosted off premises by a third party based on a confidentiality agreement, and the access is restricted to the organization's member only [7].
- *Hybrid cloud:* This model is a combination of the Public and Private cloud models, where a single organization can use the public cloud services for a certain type of data, while having its own private cloud for confidential and highly regulated data [4].
- *Community cloud:* Computing resources are operated and exclusively used by multiple organizations that have shared interests and are not available for public, it is only available for these organizations [6] [8].

Figure 2 Summarizes The Cloud Deployment Models.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

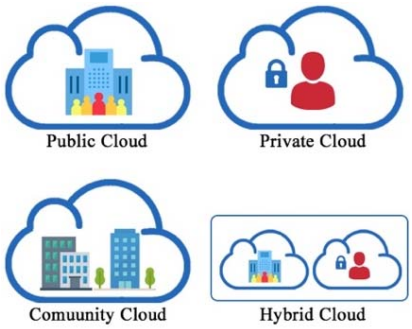

Figure 2 Cloud Deployment Models

2.3 Cloud Computing Core Components and Security Concerns:

With the development of cloud computing, there are a lot of emerging threats and attacks that raise the cloud security concerns. These attacks endanger the cloud core components, which are: Network, Hypervisor, Storage and Application [2]. This section defines these core components and briefly states each component security concerns.

- *Network*: It is the component that glues all the machines within a cloud platform together, network vulnerabilities allow the attackers to gain unauthorized access to the cloud, which might eventually affect data confidentiality, integrity and availability [2] [9].
- *Hypervisor*: Is a mechanism applied by cloud computing, it offers a pool of computing resources to be utilized by customers [10]. Virtualized environment provides efficient use of hardware and ease of installing either programs or operating systems [10]. In cloud computing the guest machine is installed by the hypervisor or so called the virtual machine monitor (VMM) which by its role, manages the computing resources of all the hosted guests as well as any communication between the hypervisor and the guest machine [10]. Hypervisor vulnerabilities reflect serious threats since hypervisor has the highest privileges, thus, if it has a vulnerability the attacker can exploit this vulnerability and run arbitrary commands to control the virtual machine [2] [9]. The below figure shows the traditional hypervisor architecture.

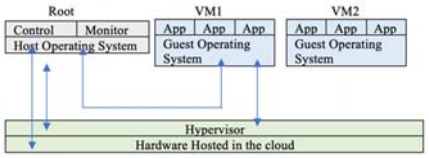

Figure 3 Hypervisor Traditional Architecture

- *Cloud storage*: It is the cloud component that is used to store, maintain and back up data, and it makes the data available through the network [3]. Data storage has many concerns that might lead to data loss, data leakage, availability problems and many other concerns [3].

2.4 Cloud Computing Characteristics:

NIST defines five essential cloud computing characteristics which are [11]:

- *On-demand self-service:* The user can request cloud computing services instantly and directly without the need of human interaction.
- *Broad network access:* Cloud computing services can be accessed from different locations and from various electronic devices, like mobile devices, tablets and laptops.
- *Resource pooling:* Consumers can dynamically use resources, like storage, memory and network bandwidth based on their needs. Whenever this resource is not in use by one consumer, it is allocated to another consumer. All resources requests are coordinated, well-structured and utilized at their most.
- *Rapid elasticity:* Cloud computing environment can adopt to customer's requirements in a flexible manner. A customer can increase the RAM allocated for a hosted server from 8 GB to 16 GB immediately and without any complexity.
- *Measured service:* Cloud computing offers a measured service for its customers, in the sense that
customers pay only for the used resource. So, if the customer used 10 Mbps in one day, the customer will be charged for the exact amount of used resources.

3. CLOUD COMPUTING ATTACKS

3.1 Network Based Attacks

Cloud computing is based on network technology, which is a core component that helps carrying out various cloud services [12]. In the same time, the fact that cloud relies primarily on network exposes

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

it to various attacks [12]. This section discusses different network attacks, which are XML-based attack, SQL Injection and Denial of Service (DoS) attacks.

3.1.1XML based attack

Simple Object Access Protocol (SOAP) is a standards-based web service access protocol used in the cloud, it allows communication between applications by exchanging SOAP messages which are in an XML format, so in short, SOAP messages are XML documents [13].

XML documents are transmitted after applying XML signature, which is achieved by asymmetric encryption using a pair of two keys, private and public keys. The sender signs the document by his/her private key, and the receiver validates the signature using the sender's public key, doing so will help the receiver validates that the message comes from a legitim source, which is known in security field as non-repudiation [14].

Usually, not the whole document is signed since that will cause performance degradation [14], instead, a hashing function is used over the whole document or parts of it to produce a unique output known as digest, this digest is unique for each message and it cannot be repeated which in turns preserves the data integrity [15].

Then, the receiver can calculate the hash over the document or parts of it and compare the generated digests to see whether there is any modification or alteration happens to the data [15].
To preserve the message authentication along with the message integrity, this digest is signed using the sender private key and sent with the message. So, the receiver calculates the received message hash value, then checks the signature element in the XML structure to decrypt the signature using the sender's public key, if decrypted correctly then it is from the intended sender. In addition, decrypting the signature will reveal the digest calculated by the sender, then the receiver compares both digests together (The one calculated by the receiver, and the one calculated by the sender), if the digests are identical then the message has not been altered [15].

To make this process achievable, there are some elements in XML documents that are specialized to store the sender's calculation of the message digest and signature [14].

In fact, SOAP messages are vulnerable to XML signature wrapping attack, which is also known as rewriting attack [13] [14].

- Wrapping attack

In this attack, the attacker makes use of the elements in the XML document. XML document is structured of various elements each has a specific id so it is possible to reference that element using its id [16].For example, there is a specific element named body that holds the body of the message, it can be written as <body id="oldBody">, then it contains the original body sent by the sender. As explained, XML document has an element that holds the signature, it has a reference to what element was signed, so if the signed element is the oldBody, then the signature will have a reference to it, this is illustrated in Figure 4.

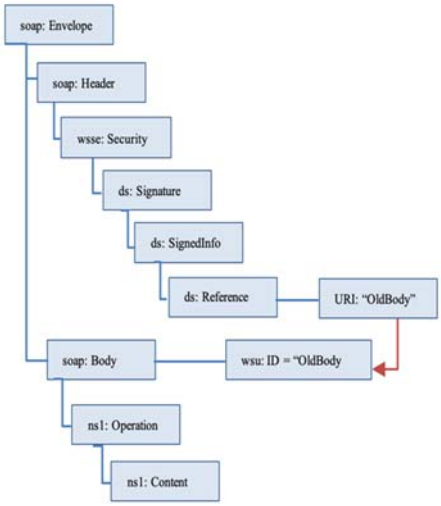

Figure 4 XML Document Structure [17]

During this attack, the attacker replaces the old body with a new crafted body that has a malicious content, this is done by changing the location of the old body and placing it in a dummy tag that is not processed by the receiver, then placing a new body in the old body location, for example <body id="newBody"> but with a malicious content [14] [13].

So, when the message is received, the receiver will check the signature element, it will look for the reference held by the signature and will find it in the dummy tag, it will do the message verification correctly. Then, it will look for the body element (the one outside the dummy tag) and will process it. Eventually, the receiver is tricked to execute the malicious content [13].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.1.2SQL injection attack

Structured Query Language (SQL) is a language used to retrieve or update data from relational databases. Nowadays, most web applications use three-tier architecture in which there is a client tier, web application tier and database tier, which are hosted in a cloud environment [18].

Web applications that deploy this architecture are vulnerable to SQL injection attack which occurs when an unauthorized user is able to enter arbitrary queries from the web application interface, and then executed in the server thus exposing confidential data or manipulating existing data stored in the database. The below Figure illustrates SQL injection in a cloud environment.

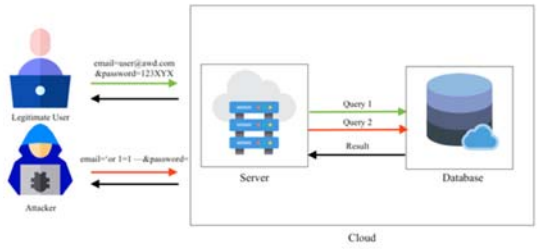

Figure 5 SQL injection Attack

This attack can cause serious harm and can ruin the reputation of the company since it affects both the integrity and the confidentiality of the stored data [19]. There are four main types of attacks associated with SQL injection [20]

1) Authentication bypass:
In this attack, the attacker is able to bypass the login page without providing a username or a password by exploiting the way the programmer wrote the code [20].

- Example:

Here is an example of a poor written SQL query that allows the attacker to bypass the authentication page:
"SELECT user FROM Users WHERE username=' $username' and password='$password' " In this way the attacker can write: **' or 1=1; --** in the username field. Here the attacker ensured that the condition will evaluate to true, by writing **' or 1=1;** and then commenting the rest of the query by writing two hyphens --. As a result, the attacker will be able to bypass the login page without providing proper authentication.

2) Leaking sensitive information

After the attacker has gained unauthorized access to the web application, the attacker can view highly confidential information stored in the database [20].

- Example:

The process performed by the attacker to expose sensitive data in the database is as follows. The first step is to find an injection point where there is an interaction with the database. Once an injection point is found, the attacker gradually exposes the number of columns, the database name, the table names, the columns names and finally the data its self. The following queries can be added by the attacker in the URL to expose each element mentioned respectively.

- Number of columns: order by 1

if the web application didn't raise an error, the number of selected columns in the query is one, otherwise it is more than one.

- Database name:

UNION(SELECT table_schema from information _schema.tables) –
As a result, the name of the web application database is exposed

- Table names:

UNION(SELECT table_name from information_s chema.tables) –
Here the attacker asks the database engine to select the names of all tables that exists in information_schema

- Columns names:

UNION(SELECT column_name from informatio n_schema.columns WHERE table_name=account s)--
Here the attacker selects column_name, which holds all the columns names of all the databases, from columns table, which holds information about all the columns in all the databases, in which the table_name equals to accounts. As a result, the three columns are exposed which are email, password and displayname.

- Data:

UNION (SELECT email from accounts) –
Now the attacker has all the necessary information, which are: database name, table name, column name, to dump the table's data. In the same manner, the attacker can write a query to retrieve users' emails, passwords, or even the displayname.

3) Loss of data integrity:
In this attack, the attacker manipulates the database content by updating an existing value or adding a new one [20].

- Example:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This attack can be established using stacked queries, the attacker first terminates the current query using a semicolon then starts a new query to update an existing value or insert a new one. Applying this concept to XYZ web application, the attacker can add the following query to the email text field:
"admin@xyz.com'; UPDATE accounts SET email='hacker@xyz.com' WHERE email = " admin@xyz.com; -- ""
This query will be executed in the database server as:
SELECT email, password FROM accounts WHERE email= 'admin@xyz.com'; UPDATE accounts SET email= 'hacker@xyz.com' WHERE email = 'admin@xyz.com'"
Here, the attacker has successfully changed the email address of the admin to his/her email address.

4) Loss of availability of data:
The attacker executes advanced query to delete highly sensitive data from the database, thus, the users will not be able to retrieve important data from the database. This attack is very dangerous and can lead to major loss to the company [20].

- Example:

Same as the previous attack, this attack can be conducted only using stacked query, in which the attacker uses SQL keyword DROP to drop a table from the database. Referring to XYZ web application, the attacker can enter the following malicious query in the email text field: "admin@xyz.com' DROP TABLE accounts; --"
The query will be executed in the database server as: " SELECT email, password FROM accounts WHERE email='admin@xyz.com'; DROP TABLE accounts; --' ; "
As a consequence, the table accounts will be dropped, and no user can login to XYZ web application.
The researchers explained the solution very well, on point illustration, but the introduction is poorly written, the ideas are scattered and the flow is not properly connected.
Error! Reference source not found. summarizes the advantages and the disadvantages of this paper.

3.1.3Denial of service attack

Denial of service attack (DoS) is one of the massive attacks that affects service availability, it is an attack designed to disrupt the cloud services and prevents legitimate users from accessing and using these services when needed [21]. Figure 6 gives an overview of one of the techniques used to cause DoS attack, where the attacker sends a huge amount of traffic to overwhelm the victim and makes it unavailable [21].

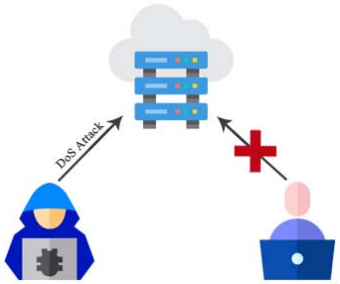

Figure 6 DoS Attack

Another approach for DoS is Distributed DoS, this approach is more intense since a group of attackers target a specific victim to make its services unavailable to legitimate users [1].

Indeed, for the attacker to stop or disrupt a service, s/he needs a huge amount of resources to attack the victim. As explained in the DDoS, the attacker needs two components, a group of compromised computers, namely a botnet, and a command-and-control server (C&C) to issue commands to botnets and by their role they attack the target and break it down. This attack becomes a way stronger in the cloud than in traditional IT system, and harder to detect and prevent since attacker has more resources [12]. Meaning the botnet and other DoS attack components might be in the cloud themselves, and by their role they attack another victim which might be in another cloud [12].
Figure 7 is a visualization of DDoS.

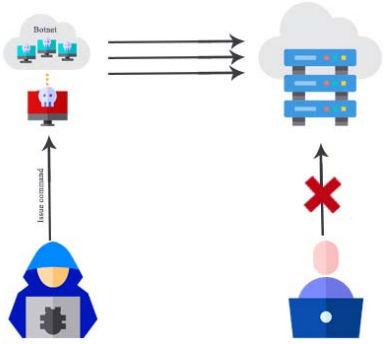

Figure 7 DDoS Attack

This section represents various DoS and DDoS attacks classified based on the network layer they target.

1) Application layer attacks:
Application layer DDoS attacks are a serious threat since it requires the monitoring system to do a deeper packets inspection and analysis to identify which packets belong to an attack and which does not, which is hard to conclude [12].

- *Back attack*

This attack mainly targets Apache web server where the attacker fills the URL with around 6000-7000 forward slashes ('/'), which eventually slows down and affects the server performance since it requires a lot of processing [22].

- *R-U-dead-yet (RUDY) attack*

R.U.D.Y is a tool used to initiate a slow rate traffic with low volume by exploiting a weakness found in HTTP protocol using websites submit forms [22]. In normal cases, websites contain submit forms where users input some data and then submit it to the server using HTTP POST method, which requires one or two packets before the connection is terminated [22]. In R.U.D.Y attack, the attacker sends legitimate yet very long "content length" header field, then it transmits the content slowly to the server, usually one byte per second. The reason behind the long content and slow transmission is to keep the connection open as long as possible [23]. By doing this repeatedly, the attacker opens many long connections until all the available connections are consumed, thus suspended from legitimate users [23].

- *Slowloris attack*

Slowloris attack is similar to R.U.D.Y attack in the sense that it produces low volume slow rate traffic [22], the attacker consumes all the available connections offered by the server, through sending partial HTTP GET requests without a termination sequence that indicates the request is complete, which forces the server to wait endlessly for the termination sequence that complete each request, which will never be sent [24]. Thus, legitimate users won't be able to open connections with the server since all the available connections are already in use. Eventually, the server services are not available for legitimate users [22] [24]. Figure 8 visualizes Slowloris attack.

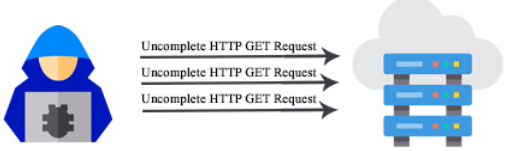

Figure 8 Slowloris Attack

2) Transport layer attacks

- *UDP storm attack*

This attack targets two victims at the same time, let's say there are two victims, A and B. The attacker will send UDP packets to A, with spoofed source IP that belongs to B. Then, A will reply to B thinking that B is the real sender, then B will reply to A and the loop continues until the connection is disrupted [22]. Such an attack can overwhelm a server if it is one of the communicating ends and make it unavailable for authorized users [22].

- *SYN flooding attack*

This attack exploits the connection-oriented feature found in Transmission Control Protocol (TCP), where it uses three-way handshake to establish a connection before transferring data between communicating hosts [25]. In normal cases, for the connection to be established, the sender sends a message with a SYN flag set, the receiver must reply with a message where SYN and ACK flags are set, then the sender will reply with a message with ACK flag set to complete the handshake [25]. During SYN flooding attack, the attacker establishes half-open connection where s/he sends a message with SYN flag set, the receiver replies with a message with SYN and ACK flags set, then the receiver waits for the sender (the attacker) to reply with the final ACK, but the attacker will never reply. In contrary, the attacker will keep sending these half handshakes to use up the kernel memory with a huge number of transmission block allocations that won't be free until its time expire [22]. Eventually, legitimate users won't be able to establish connections with the victim since the victim has no room in its buffer for them [25] [22]. Figure 9 visualizes SYN flooding attack.

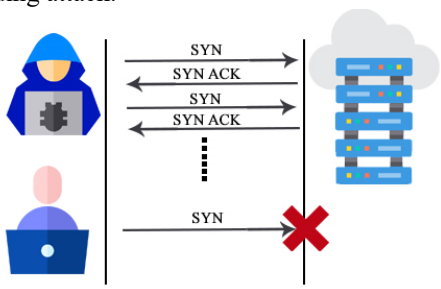

Figure 9 SYN Flooding Attack

3) Network layer attacks

- *Smurf attack*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This attack exploits the working mechanism of the Internet Control Message Protocol (ICMP) where a sender sends an ICMP echo request then gets and ICMP echo reply [22]. The exploitation is done as follows: The attacker sends an ICMP echo request with a spoofed IP address to the broadcast IP address of the network, then this request is transferred to all connected hosts, all these hosts will reply with ICMP echo reply to the spoofed victim IP, meaning, the response is amplified by the number of hosts who received the requests then reply with a huge number of echo replies, which eventually overwhelm the victim and causes it to break down [12] [22].

- *Teardrop attack*

Fragmented IP packets are reassembled at the target based on the offset field in the fragments, which helps the target to arrange the fragments correctly [26]. The fragment offset is based on the length of the previous fragment, for example, if a packet of size 2300 bytes is too large for a transmission medium, it might be fragmented to 2 fragments, one of 1500 bytes size with offset 0, which indicates that it is the first fragment, and the other one is of 800 byte size with offset 1500 (Which is the length of the previous fragment) indicating that this fragment starts at the end of the previous fragment. In teardrop attack, the attacker fragments IP packets and set incorrect offset number that causes the fragments to be overlapped when reassembled [22]. Accordingly, the receiver won't be able to reassemble or process the fragments, and simply keeps holding all the fragments that are improperly configured which consumes the available buffer space. Eventually, as the buffer space is totally consumed, there is no room for new connection from legitimate users [22].

Error! Reference source not found. summarizes the mechanism of the network-based attacks along with its categories, implication, and their solutions.

3.2 Virtualization Based Attacks

Many Virtual Machine Monitor (VMM) vendors claim that they provide 100% isolation between the host and the guest. However, this is not the case, current VMMs can contain serious vulnerabilities that can expose the host to heavy attacks [27]. This can occur since the guest has exactly the same resources as the host machine, the same number of CPUs, memory, the same patches and configuration [27]. The attacker can examine the applications located in the guest and if a vulnerability is found, the attacker can exploit this vulnerability to attack other guests or attacking the host its self, which is a critical issue that should be addressed [27]. Thus, this section discusses various attacks that can arise in cloud virtualization.

- *Guest to host attack/guest escape*

Once the attacker has found a vulnerability in the virtualization layer in combine with improper configurations of both the host and the guest, s/he can bypass the virtualization layer and access the host machine [27]. And since the host machine contains multiple guests, the attacker can control all the guest machines and monitor any interaction between the guests and the host [27]. In addition, the attacker can lunch various attacks, like, corrupting resources, memory, CPU, and launch arbitrary code [27].

- *Guest to guest attack /guest hopping*

In this attack, the attacker can inject a malware in one guest, and once s/he gets a control over the virtual machine, s/he can spread this malware to other virtual machines or attacking the virtualization layer its self [28]. Thus, controlling all the virtual machines that exist on the host machine [10]. The attacker then can monitor the usage of various resources, like, CPU, memory, etc. which affects the confidentiality of the guest machine [10]. In addition, the attacker has the ability to miniplate existing data in the virtual machines, modifying their configurations, injecting malicious code, etc. [32]. Thus, affecting the integrity and the availability of the data [32].

- *Guest mobility*

Guest machine contents are stored as files in the host machine's hard desk drive, thus, easing the process of transferring or copying the contents of one guest to another host through the network [29]. With this usability, security problems arise, if the guest is infected with malicious malware, the other host will be contaminated with the same malware [29]. Thus, the attacker will have control over multiple virtual machines on multiple hosts and possibly use the same technique to affect multiple virtual machines [29].

- *Guest denial of service attack*

In virtualization, the host machine allocates resources such as
RAM, CPU, storage and network bandwidth for each guest machine [29]. DOS attack occurs when one guest machine
occupies all the resources resulting in denying other guest machines from utilizing host's resources [29].

- *Virtual machine overflow*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In this attack, the attacker runs a malicious script on the guest machine and fills the allocated memory region with meaningless characters, exceeding the allowed boundaries for the guest machine and as a result the machine crashes [28]. After that, the attacker can access the host's memory pointer's and directing them to run the attacker's malicious script [28]. By that, the attacker can gain root access over the host machine and thus having access over all the guest machines that resides in the host machine [28].

- *Virtualization memory leak*

Each guest machine has a specific space in host's memory and if the host didn't properly free the allocated memory, a virtual memory leak can occur [30]. The attacker can exploit this vulnerability by using this allocated space to execute several attacks, like DOS and buffer overflow attack [30].

3.3 Data Storage Attacks on Cloud

Cloud storage is another core component that is managed, maintained and backed up remotely where users can access it using the network. In addition, the storage of data and its security over the cloud computing is one of the major issues. This section discusses a couple of attacks on the cloud storage.

- *Inference attack*: is a data mining technique that can be done by analyzing data in order to gain knowledge about a database or any subject illegally, without accessing it directly. An inference attack may affect the integrity of an entire database. Moreover, this sensitive information will be considered as leaked, if an attacker can infer the real value of the data at a high level. [3]
- *Pollution attack*: is one of the most dangerous threats that affect the data integrity. Where a malicious user take control of one or more storage resources to prevent the availability of data, by polluting the data or part of it. Pollution attack occurs when coding techniques are used to represent data outsourced on storage resources. In such a case, where single data items are first subdivided in parts, then encoded to preserve an adequate number of coded fragments to be placed on a set of separated storage resources; to allow the user to reconstruct the original data item from the set of coded fragments, the coded fragments must be computed in a suitable subset [31]. Using this coding techniques arise a couple of major issues:

1. It is hard to find out whether the data has been altered by an adversary storage node before the identical data item is recovered by the user, because any sequence of bits may be a legitimate coded fragment [31]
2. It is hard to distinguish between the malicious storage resource and the legitimate one. Thus, if there has been any recovered data item and it was detected as polluted, it is not easy to identify which coded fragments were polluted amongst those received by the user [31]. Table 3 summarizes the mechanism of the data storage attacks along with its categories, implication, and their solutions.

4. COUNTERMEASURES

4.1 Network Attacks Countermeasures

This section presents various countermeasures for XML, SQL and Dos attacks.

4.1.1Rewriting attack countermeasures

a. *A histogram-based method for efficient detection of rewriting attacks in simple object access protocol messages*

In the proposed solution, Nasridinov, Jeong, Byun and et al. [17] Took advantages of packets' headers to include the document structure in the the header. So, the receiver can cross check the XML structure included in the packet header with the real XML structure of the sent SOAP document. The structure label of the XML nodes is constructed and stored using Dewey labeling scheme (DLS), where each level in the XML tree structure is assigned a number, so the whole structure is a numbered-levels structure, and every node will have a series of numbers to indicate its position in the document, this series of numbers is called label. The numbered label is constructed from 3 components.

First component is the Level component, which identifies the current node level, where the level starts from 1. Secondly, Inherited label component, which is inherited value from the parent node label, to be placed in the current node label, it helps identifying the series of parents for this current node. Lastly, Sibling order component, which identifies the order of the current node in relation to its siblings in the same level, whether it is the first, second, etc. node. All these components are concatenated together using (.), so each node will have its label as: (Level component, Inherited label component, Sibling order component).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

So, if the rewriting attack occurred, where the attacker changes the location of the signed element, the receiver will know since the receiver needs to build DLS structure for the received message to compare it with the attached structure, if difference is found, then attack is detected.
Another strength in this methodology is that every time an attack is detected, its location is recorded in histogram for statistical purposes. This histogram indicates the most used locations for attack, accordingly, avoids checking all the SOAP structure and only checks the locations recorded in the histogram, which reduces overhead.

b. *Xml wrapping attack mitigation using positional token*

Kumar, Rajendran, Bindhumadhava and et al. [32] Proposed a solution that requires modification in the XML standard, the researchers suggested an algorithm for signing and verifying signature based on a Positional Token. The essence of this algorithm is to use a different attribute to refer to the signed element. They stated that using the attribute Absolute XPath with <Reference URI= the element's Absolute XPath> instead of ID with <Reference URI=ID> will indicate the position of the signed element. The absolute XPath of an element is the path from the Envelope to that element, the path states the names of the nodes from the Envelope to the signed element, for example (Envelope/Header/Body/etc.). So, the signature calculation no longer relies on the signed element alone, it considers its location. The proposed algorithm search for every element referenced using ID then extracts its Absolute XPath and uses it as a value for the URI.

The verification algorithm works in the same way, once the XML document is received, the verification algorithm checks all URIs in the signature node, it will navigate to the Absolute XPath for these URIs and calculate the digest, the digest is not for the node alone, it is for the node and its path. So if the calculated digest equals the value of the sent digest, the verification succeeds, else, the verification fails and the attack is detected, since in the case of an attack, when the verification algorithm navigates to the location specified in the URI, it won't find it since the attacker placed it in a dummy tag, which changes its location. In addition, if the attacker manually modifies the URI in the signature to point to the new location of the old body (Inside the dummy tag), the verification fails since the digest value will change, meaning, the newly calculated digest that considers the new location will not equal the sent digest.

4.1.2SQL injection attack countermeasure

D. G. Kumar and M. Chatterjee [33] proposed a methodology to detect SQL injection attacks via first removing the sent parameters from the SQL query then comparing them with the original query. This method uses a combination of static and dynamic analysis. In which the original query is referred to as static while the dynamic analysis refers to queries at the run time. An illustration with example will explain the methodology more clearly:
Once the user enters an input, it is sent to the server and the server places the parameters in the SQL query. First, a delete function is executed to remove the parameters from the SQL query then the static query is compared with this dynamic query, if they are equal then there is no SQL injection otherwise there is an injection. For instance, assuming a benign user has entered a username, "Rachid" and a password "123Rachid" the server will first place the sent parameters in the SQL query:
Q1: SELECT * FROM USERS WHERE USERNAME= 'Rachid' AND PASSWORD= '123Rachid ';
Then, the delete() function will be executed by the server:
Q1'= delete(Q1)= SELECT * FROM USERS WHERE USERNAME= '' AND PASSWORD='';
The server will compare Q1' to Q2 which is:
Q2: SELECT * FROM USERS WHERE USERNAME= ' ' AND PASSWORD= ' ';
The queries are equal so there is no SQL injection attack. The same idea is applied for the malicious user. If the malicious user enters a username "1 or 1=1 --" and a password "attacker", the query will be:
Q3 = SELECT * FROM USERS WHERE USERNAME ='1' or'1=1'—'AND PASSWORD ='attacker';
Again, running delete() function in the server will result with the following query:
Q3'= delete(Q3)= SELECT * FROM USERS WHERE USERNAME ='' or'—'AND PASSWORD='' The comparison will yield to false which means an SQL injection has occurred. The below Figure visualizes the methodology of the proposed solution.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

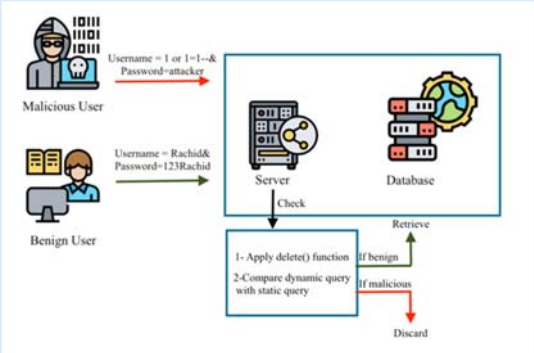

Figure 10 SQL injection Detection Methodology

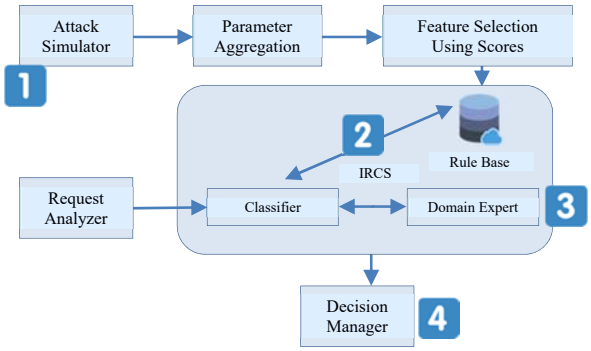

Figure 11 Architecture of intelligent rule-based classification system (IRCS)

4.1.3DoS attack countermeasure

a. *Detection of DoS attacks in cloud networks using intelligent rule-based classification system*

Rajendran, Kumar, Palanichamy, and et al. [21] Proposed a solution to detect DoS attack using machine learning classification techniques. They defined some metrics (features) to be monitored and recorded before and after the attack, which are under these categories: process usage (such as CPU usage, CPU load, etc.) memory usage (The memory space needed for the process, the memory space used by the OS and its related program in the meanwhile, etc.) and network bandwidth usage (The TCP flag set, such as close or SYN, its waiting time, etc.).

Figure 11 shows the general architecture of the proposed solution, where the attacker initiates the attack, then the selected features are collected in the parameter aggregation phase, then each of these features are ranked and given a score based on their practical importance. After the features are ranked in order, the researchers used Fisher's discriminant criterion to select the best n features from the ranked list, Fisher's algorithm helps selecting features that are clearly related to specific attack class, examples of attack class is Ping flood attack. After the selection, Fisher's selected features are compared against the rules found in the Rule Base, where each rule specify if-then condition, where a condition is explicitly specified in the (if) clause, and its action is in (then) clause. These rules are set in relation to the metrics defined in the beginning.

if the traffic selected features' match any of these rules, then the traffic is marked as suspicious or malicious, otherwise, the decision about the traffic legitimacy is left for the domain expert, where they analyze the traffic that do not have a match in the Rule Base and make a decision about it.

b. *Preventing cloud systems against DDos attack using hop count filter approach*

Zalak and Prof. Upadhyay [34] proposed a solution based on the hop count to discriminate the traffic coming from spoofed sources, which is usually done by DoS attacker to hide their identity.

The proposed solution checks the packet hops by counting it from the destination to the source using TTL, it subtracts the final TTL from the initial TTL, the initial TTL values are inferred based on the end user OS. Basically, the cloud would have a system that stores legitimate users' IPs and their hop counts, then checks whether the received data from the specific IP has a hop count that matches the stored data for that IP. If the hop counts are not equal, then the IP is spoofed, and the packets are dropped since they came from illegitimate user.

4.2 Virtualization Countermeasures

This section presents three virtualization architectures that can eliminate security issues that arise in traditional virtualization.

4.2.1No hype architecture

In this architecture, the virtualization layer is removed and replaced with a system manager that has the same functionalities as the hypervisor [35]. And since the most vulnerable element in cloud virtualization is eliminated, the architecture becomes more secure [35]. Here, the guest machines interact with the hardware directly without having to consult a virtualization layer [35]. Instead, the management software will allocate the needed resources for each guest and each guest machine will use its allocated resources

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

without intruding on other guest's resources [35]. No Hype architecture is able to manage CPU, memory, input/output device, the operations of the guest machines and ethernet switching [35]. The below Figure shows No Hype architecture.

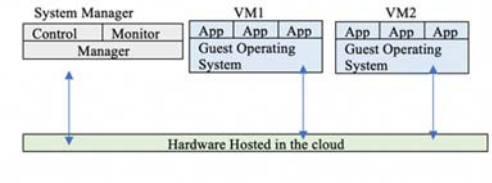

Figure 12 No Hype architecture

- CPU management: each guest has one core, so the number of guests should equal to the number of CPU cores [36].
- Memory usage: Upon user request requirements at the first stage the memory is allocated accordingly. So, if the user requested 2 GB of RAM the user will be allocated the exact amount of memory, no more no less [36].
- Input/output devices allocation: This architecture provides I/O virtualized devices by themselves like NIC, storage devices, network connections, etc. Each I/O device is related to its core with the help of Multicore Memory Controller (MMC) and Memory Management Unit (MMU). Thus, ensuring that each guest machine uses its own core and its related virtualized I/O devices [36].
- Managing guest machines operations: The lifecycle of the guest machine is managed by the system manager starting from mapping the needed resources to the guest machine, stopping or aborting its execution or even migrating the guest from one server to another [36].
- Ethernet switching: No Hyper architecture is superior over the traditional virtualization in the sense that it doesn't need a software switch to manage to manage the network traffic since the guest machines already have a direct access to the hardware [36]. Thus, enhancing the CPU performance and eliminating he burden of managing ethernet switching for large number of guests [36].

4.2.2Hyper wall architecture / hypervisor secure virtualization

This architecture is mainly based on two security aspects, confidentiality and integrity [36]. It gives its customers a feeling of security by providing authenticity, through hashing algorithms, and test assurance to ensure that the guest machine doesn't violate the security of this architecture [36]. It also plays the role of firewall by preventing malicious users from accessing restricted areas in the hardware.

This architecture depends on the hardware itself and has hardware additions in the sense that the memory portion of the guest machines are totally isolated from the hypervisor memory [36]. Therefore, users can run their programs on the virtualized environment without doubting the security of the architecture [36]. Ultimately, this architecture isolates the three main components, guest, host and the hardware completely.

4.2.3No hyper - hyper wall hybrid architecture

This architecture is a mixture of the previous two architectures, No Hyper and Hyper Wall architecture by combining their strengths and eliminating their weaknesses [36].

This architecture categorizes guest users based on their security needs. Two main categories exist, critical users, who need high security measures and common users, who need minimal security measures [36].

First of all, the user chooses the required level of security based on the two mentioned categories. This request will be transferred to the memory, which contains status table, a mapping of each user along with its category, and upon the categorization the critical users will be switched to No-Hyper sub-platform and the common users will be switched to Hyper Wall sub-platform [37]. The below Figure shows the hybrid architecture.

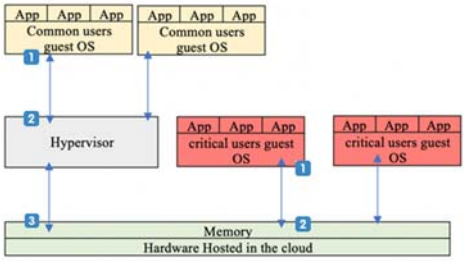

Figure 13 No Hyper - Hyper Wall Hybrid Architecture

4.3 Cloud Storage Countermeasures

4.3.1Inference countermeasures

Inference control architecture usually composed of several policy modules that regulate the security of the database. The inference control architecture typically takes a place on the server where the database resides. [38]

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 14 shows a typical architecture consists of three basic blocks, which are: ACM, Database and Inference Control Module (ICM). An Access Control Module (ACM) is always exist, as the first layer of defense, which can be a DAC or a MAC access control module, and their job begin before any of the inference control starts. by regulating the user access privileges and permissions. All the user requests to the server are directed to the ACM, to crosscheck the user with the AC database in order to confirm their validity and permission for a specific request. Then, the credentials and permissions that are regulated by the server administrator are stored in the Access Control Database. afterwards, the user's request that was confirmed by the ACM is sent to the ICM for further check of any inference channel treat. In this architecture, the ICM composed of two blocks which are: IC policy and Query Log [38]. Each request that is forwarded to the ICM for checking is firstly stored in Query Log and then examined for treats using the Inference Control Policy (ICP).

Based on the type of the ICP, different strategies of inference control such as, QSOC,QSSC, etc. are used in combination with all previously queries stored in the Query Log. If the ICP does not detect any inference treat, then the user's query is marked as safe and revealed to the user [38].

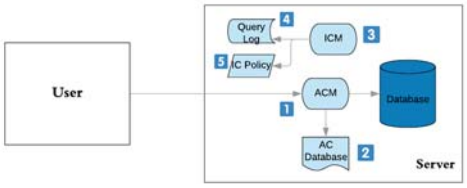

Figure 14 Traditional Inference Control Architecture

4.3.2 Pollution attack countermeasure

The proposed solution is based on a pollution detection algorithm that detects, with high probability if a set of untrusted storage resources provides at least one polluted coded fragment. This algorithm is consisting of a modified version of the Luby Transform (LT) decoding algorithm exploiting Gaussian Elimination [39]. Moreover, this solution has an identification algorithm with a high probability in identifying the storage resources that are polluters [39].

Table 1,2 and 3 summarize the attacks and their respective solutions along with the suggested countermeasures.

5. ANALYSIS

Based on the comparison on **Error! Reference source not found.** specifically rewriting attack , the histogram method proposed by Nasridinov and et al. [17] yields better results, since it is fast due to its reliance on the labels rather that string paths. In addition, it implies continuous update of the indication of compromise. All these reasons contribute to reduces the effort spent to detect wrapping attack. On contrary, Xml wrapping attack mitigation using positional token proposed by Kumar and et al. [32], requires higher of processing efforts and cost due to its linear relation to SOAP message size.

Katole and others proposed SQL injection detection system it does not prove it robustness against other types of SQL injections, such as blind SQL and common Web Application Firewall bypass, URL encoding, hex representation. Additionally, it has a considerably low detection time and doesn't apply to all databases, like NoSQL.

On the other hand, **Error! Reference source not found.** summarizes DDoS attacks countermeasures where the method IRCS proposed by Rajendran and et al. [21], reduces false positive, and is constantly updated due to the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1 Network Based Attacks and Countermeasures

Attack name	Categories	Implication	Solutions	Advantages	Disadvantages
Wrapping Attack	Wrapping Attack	Execution of malicious requests	[17]	-The histogram helps collecting data about the rewriting attack which helps reducing the cost for detecting the attack in the future -Fast processing and detection since it rely on labels rather than paths.	-As the SOAP message elements increases, the DLS grows, which increases the size of the SOAP message.
			[32]	-No extra advantages other than protecting against rewriting attack.	- Long processing time for the path string - The cost of detecting the attack is linear to the SOAP message height since each element in the SOAP message must be processed and inspected to see whether it matches the signed element's path or not. - With the addition of path strings, the SOAP message size tremendously increases.
SQL injection	Authentication Bypass Leaking sensitive information Loss of Data Integrity Loss of Availability of Data	Alters the content of the database and can deny legitimate users from using web application services.	[33]	- Does not affect the current code segment. -Can detect all types of SQL injection. -The detection rate is very high around 98.67%.	-Not applicable for all databases, like NoSQL. -Doesn't provide a proof for the rest of SQL injection attacks -The detection time is considerably slow, about 12,500 millisecond.
Dos Attack	Back Attack RUDY Attack Slowloris Attack UDP Storm Attack SYN Flooding Attack Smurf Attack Teardrop Attack	The service becomes unavailable for legitimate users	[21]	-The reduction in the false positive rate and the increase in security. -The classifier efficiency is in continuous improvement due to the knowledge provided by the domain experts	-Temporal constraints are needed to capture the dynamic nature of an attack.
			[34]	-Reduces false positive	-Limited to flooding attack -Detecting attacks for legitimate users is not considered in the proposed solution

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2 Virtualization Based Attacks and Countermeasures

Attack name	Mechanism	Implication	Solutions	Advantages	Disadvantages
Guest to Host Attack/Guest Escape	The attacker exploits an existing vulnerability in the guest machine to escape and resides in the host machine.	Running arbitrary code and corrupting computing resources.	No Hype	No Hyper architecture allocates one core to each guest machine, providing strong resource isolation and ensuring that no guest can snoop over the other. Hence, attacks like overflow are not applicable in this architecture.	No Hyper is a software-based architecture in contrast with the traditional hypervisor which is hardware-based. So, this architecture can be untrustworthy for some users since having a hardware-based is way more effective and secure.
Guest to Guest Attack /Guest Hopping	The attacker exploits an existing vulnerability in the guest machine to hop and resides in another guest machine.	Affecting the confidentiality of other guest machines by monitoring the activity and affecting the integrity by manipulating guest's files.			
Guest Mobility	The attacker injects a malicious malware inside the guest machine files, thus, if the guest files are transferred to another host, that host will be infected.	High propagation of malware and controlling large guest machines.	Hyper Wall	This architecture manages guest machines resources effectively ranging from CPU scheduling, memory scheduling, etc. Additionally, it outperforms traditional virtualization by providing high scalability and increasing the performance of the used resources.	All the components of this architecture depend on the hypervisor, single point of failure. Hence, if the hypervisor is attacked the whole system collapse.
Guest Denial of Service Attack	The attacker occupies the whole computing resources of the host machine.	Denying virtual machines from utilizing their allocated resources.			
Virtual Machine Overflow	The attacker runs a malicious script to overflow the guest memory region and access the hypervisor memory.	The attacker gains root access, thus, controlling the host and all the guest machines.	Hybrid		
Virtualization Memory Leak	This attack occurs if the host didn't properly free up the allocated memory for a gust after its usage.	The attacker can use the allocated memory region to launch attacks like DoS and buffer overflow.		This architecture categorizes each used based on his/her security needs. Therefore, lowering the cost while providing sufficient security.	The status table is filled manually which is subjected to human error.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3 Data Storage Based Attacks and Countermeasures

Attack name	Mechanism	Implication	Solutions	Advantages	Disadvantages
Inference attack	One of the data mining technique that is performed in order to gain knowledge about a database or any subject illegally, without accessing it directly.	Affects the privacy and integrity of data	[38]	No extra advantages other than eliminating from Inference attack.	There is no complete inference detection model for all database structures.
Pollution attack	The attacker take control of one or more storage resources to prevent the availability of data, by polluting the data or part of it.	Affects the integrity and availability of data.	[39]	The detection mechanism alone is not enough to identify the adversary storage nodes in order to remove them from the system. Thus, this solution proposes an algorithmic that uses both pollution detection by using rate less codes, and statistical inference to identify the adversary nodes.	-Limited to Luby Transform rate-less codes. - This solution is only exploiting coding redundancy and efficient decoding algorithms that demand the solution of systems of linear equations.

Expert Domain factor, who decide about the suspicious traffic and accordingly, improves the solution behavior. The structure of this solution makes it flexible, which enables it to include the count filter approach proposed by Zalak and Prof. Upadhyay [34]. The count hops solution has less advantages and restricted to one attack, while IRCS is more inclusive.

In addition, referring to **Error! Reference source not found.**, three solutions have been introduced, each has its own advantages and disadvantages. However, Hybrid architecture outweigh No Hype and Hyper Wall since it combines both architectures while preserving their strengths and eliminating their weaknesses. What makes Hybrid architecture effective, is it distinguishability between critical users and normal users, thus, allocating the needed resources upon user's needs.

Both Hybrid and Hyper Wall keep the hypervisor layer while No Hype eliminates it completely and depends on a system manger, software, which is not as strong as the hypervisor, hardware.

Additionally, Hybrid architecture switches critical users to No-Hype architecture, by allocating system resources to each guest, which increases the security, while common users are switched to Hyper-Wall architecture where they all share the same resources, which is subjected to threats.

Moreover, according to the analysis on

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3, each solution is focusing on a specific attack on cloud storage. The first one which is the inference attack countermeasure that is proposed by [38], it only eliminates this type of attack and there is no complete inference detection model for all database structures. The second countermeasure is for the pollution attack that is proposed by [39], this solution proposes an algorithmic that uses both pollution detection by using rate less codes, and statistical inference to identify the adversary nodes. Thus, it is suggested to combine both solutions in order to mitigate from the cloud storage attacks and preserve the integrity and availability of the data stored in the cloud.

6. CONTRIBUTION

The analysis section V. proved that each solution is not a stand-alone solution by itself, for this reason, this paper proposes a comprehensive system which merges the best solution from each and every core component in cloud computing, including network, hypervisor and data storage. The phases of the proposed system start by implementing histogram method [17] to send and receive SOAP messages and analyzing network any packets (Not necessarily packets related to SOAP messages) against predefined metrics, then classifies them using advanced classifiers. Once the attack is classified, the action to be taken is directed by the stored rules in the Rule Base. These metrics, classes and rules can be further expanded to classify rewriting attack, SQL injection or any other network attack.

If the attack is extraordinary, the packet is put on hold and transferred to Domain Expert for further analysis. If the packet is found to be malicious, the action is taken by Domain Expert and the action is stored in the Rule Base. Once the packet is found to be benign it is processed and stored in the storage medium successfully.

The second phase starts once a critical or a common user requests access to a storage medium, the request is processed in the memory by comparing the request id to a status table, if the user is found to be critical it request switches to No-Hyper sub platform, otherwise it is switched to Hyper Wall sub platform.

The third phase starts once a polluted code fragment is found in the storage medium, the algorithm runs and eliminates the storage nodes from the system. Thus, implementing such a system will address most of the significant attacks that affect the cloud computing. Figure 15 depicts the purposed system.

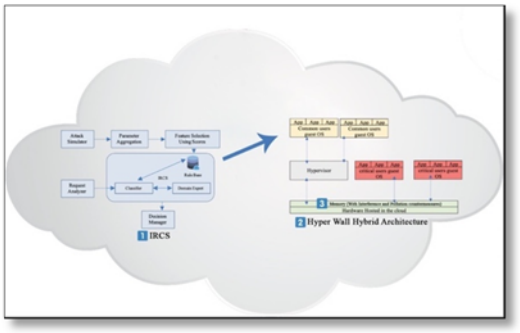

Figure 15 A secure cloud computing system

7. CONCLUSION AND FUTURE WORK

The evolution of cloud computing results a major changing in the IT field, since it brings many advantages for organizations, companies and even countries. Despite its benefits, the cloud computing is still vulnerable to many security threats, and this is what makes the adoption of cloud computing one of the major challenges in terms of security. Therefore, it is important to all stakeholders and actors to understand that security must be built at every component in the cloud computing platform as well as to learn how to mitigate its risks appropriately. This paper focused on the cloud computing core components which are Network, virtualization and data storage.

Moreover, it gave a detailed explanation about various attacks that threaten these components. As well as, it summarized these attacks and their solutions based on specified criteria. Lastly, it proposed a secure cloud computing system.

The future direction of this survey is centered around studying the integration of the best countermeasures for network, virtualization and data storage attacks, by integrating histogram-based method for efficient detection of rewriting attacks, detection of sql injection attacks by removing the parameter values of sql query, detection of DoS attacks in cloud networks using intelligent rule-based classification system for network attacks, along with hybrid architecture for virtualization attacks and inference countermeasure for data storage attacks, to produce a cloud system that is secure in each and every component and to study its efficiency and effectiveness.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES

[1] G. Somani, M. S. Gaur, D. Sanghi, M. Conti And R. Buyya, "Ddos Attacks In Cloud Computing: Issues, Taxonomy, And Future Directions," Computer Communications, Vol. 107, 2017.

[2] D. J. Prathyusha And S. Naseera, "A Study On Cloud Security Issues," Multiagent And Grid Systems, Vol. 13, No. 2, 2017.

[3] S. Singh, Y.-S. Jeong And J. H. Park, "A Survey On Cloud Computing Security: Issues, Threats, And Solutions," Elsevier, Vol. 75, 2016.

[4] S. Basu, A. Bardhan, K. Gupta, P. Saha, M. Pal, M. Bose, K. Basu, S. Chaudhury And P. Sarkar, "Cloud Computing Security Challenges & Solutions-A Survey," In 2018 Ieee 8th Annual Computing And Communication Workshop And Conference (Ccwc), Las Vegas, Nv, Usa, 2018.

[5] Nist, "Nvlpubs.Nist.Gov," 2011. [Online]. Available: Https://Nvlpubs.Nist.Gov/Nistpubs/Legacy/Sp/Nistspecialpublication800-145.Pdf. [Accessed 10 February 2018].

[6] Y. C. Liu And C. L. Li, "A Stratified Monitoring Model For Hybrid Cloud," Trans Tech Publications Ltd, Vols. 719-720, 2015.

[7] T. Diaby And B. B. Rad, "Cloud Computing: A Review Of The Concepts And Deployment Models," International Journal Of Information Technology And Computer Science, Vol. 9, No. 6, 2017.

[8] S. Goyal, "Public Vs Private Vs Hybrid Vs Community - Cloud Computing: A Critical Review," Nternational Journal Of Computer Network And Information Security, Vol. 6, No. 3, 2014.

[9] L. Coppolino, S. D'antonio, G. Mazzeo And L. Romano, "Cloud Security: Emerging Threats And Current Solutions," Elesevier, Computers And Electrical Engineering, Vol. 59, 2017.

[10] T. Yucel And K. K. Romuald, "Cloud Computing Virtualization And Cyber Attacks: Evidence Centralization," In Civil-Comp Press, Stirlingshire, Scotland, 2015.

[11] P. Mell And T. Grance, "The Nist Definition Of Cloud Computing," National Institute Of Standards And Technology Special Publication, Gaithersburg, 011.

[12] M. Masdari And M. Jalali, "A Survey And Taxonomy Of Dos Attacks In Cloud Computing," Security And Communication Network, Vol. 9, No. 16, 2016.

[13] A. A. Shaikh, "Attacks On Cloud Computing And Its Countermeasures," In 2016 International Conference On Signal Processing, Communication, Power And Embedded System (Scopes), Paralakhemundi, India, 2016.

[14] J. Kumar, B. Rajendran, B. S. Bindhumadhava And N. S. C. Babu, "Xml Wrapping Attack Mitigation Using Positional Token," In International Conference On Public Key Infrastructure And Its Applications (Pkia), Bangalore, India, 2017.

[15] G. K. Sodhi And G. S. Gaba, "An Efficient Hash Algorithm To Preserve Data Integrity," Journal Of Engineering Science And Technology, Vol. 13, No. 3, Pp. 778-789, 2018.

[16] R. Mohana And D. Dahiya, "A Proposed Soap Model Against Wrapping Attacks And Insecure Conversation," Ijcsi International Journal Of Computer Science, Vol. 10, No. 2, 2013.

[17] A. Nasridinov, Y.-S. Jeong, J.-Y. Byun And Y.-H. Park, "A Histogram-Based Method For Efficient Detection Of Rewriting Attacks In Simple Object Access Protocol Messages," Security And Communication Networks, Vol. 9, No. 6, 2016.

[18] A. Orso, "Sql Injection Attacks," In Encyclopedia Of Cryptography And Security, Boston, Springer, 2011, Pp. 51-59.

[19] S. E. A, "Framework For Data Security From Sql Injection In Cloud Computing," International Journal Of Advanced Research In Computer Science, Vol. 9, Pp. 257-262, 2018.

[20] N. Singh, "Sql Injection: Types, Methodology, Attack Queries And Prevention," In 2016 3rd International Conference On Computing For Sustainable Global Development (Indiacom), New Delhi, India, 2016.

[21] R. Rajendran, S. Kumar, Y. Palanichamy And K. Arputharaj, "Detection Of Dos Attacks In Cloud Networks Using Intelligent Rule Based Classification System," Cluster Computing, 2018.

[22] B. B. Gupta And O. P. Badve, "Taxonomy Of Dos And Ddos Attacks And Desirable Defense Mechanism In A Cloud Computing Environment," Security And Communication Networks, Vol. 28, No. 12, 2017.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[23] M. M. Najafabadi, T. M. Khoshgoftaar, A. Napolitano And C. Wheelus, "Rudy Attack: Detection At The Network Level And Its Important Features," In Flairs Conference, 2016.

[24] O. Yevsieieva And S. M. Helalat, "Analysis Of The Impact Of The Slow Http Dos And Ddos Attacks On The Cloud Environment," In 4th International Scientific-Practical Conference Problems Of Infocommunications. Science And Technology (Pic S&T), Kharkov, Ukraine, 2017.

[25] O. Osanaiye, K.-K. R. Choo And M. Dlodlo, "Distributed Denial Of Service (Ddos) Resilience In Cloud: Review And Conceptual Cloud Ddos Mitigation Framework," Journal Of Network And Computer Applications, Vol. 67, 2016.

[26] B. A. Forouzan, Data Communications And Computer Networks, Singapore: Mcgraw-Hill Education, 2013.

[27] N. O Sri, K. Tapas And V. Vedula, "A Survey On Security Aspects Of Server Virtualization In Cloud Computing," International Journal Of Electrical And Computer Engineering (Ijece), Vol. 7, No. 3 , Pp. 1326-1336, 2017.

[28] A. Tayab, Junaid, W. Talib And M. Fuzail, "Security Challenges For Virtualization In Cloud," Technical Journal, University Of Engineering And Technology (Uet) Taxila, Pakistan, Vol. 20, No. 3, Pp. 111-116, 2015.

[29] H.-Y. Tsai, M. S. A. A. Miede, Y.-L. Huang And R. Steinmetz, "Threat As A Service? Virtualization's Impact On Cloud Security," In Icde 2012 28th Ieee International Conference On Data Engineering, Washington, Dc, Usa, 2012.

[30] C. N. Modi And K. Acha, "Virtualization Layer Security Challenges And Intrusion Detection/Prevention Systems In Cloud Computing: A Comprehensive Review," Supercomputing, Vol. 73, No. 3, Pp. 1192-1234, 2016.

[31] A. Viswas V And P. Samuel, "Preventing Pollution Attacks In Cloud Storages," 2018.

[32] J. Kumar, . B. Rajendran , B. S. Bindhumadhava And N. S. C. Babu, "Xml Wrapping Attack Mitigation Using Positional Token," In International Conference On Public Key Infrastructure And Its Applications (Pkia), Bangalore, India, 2017.

[33] D. S. S. S. A. D. V. M. T. R. A. Katole, "Detection Of Sql Injection Attacks By Removing The Parameter Values Of Sql Query," In 2018 2nd International Conference On Inventive Systems And Control (Icisc), Coimbatore, India, 2018.

[34] P. Z. N And P. H. Upadhyay, "Preventing Cloud Systems Against Ddos Attack Using Hop Count Filter Approach," International Journal Of Advanced Research In Computer Science, Vol. 9, 2018.

[35] E. Keller, J. Szefer, J. Rexford And R. B. Lee, "Nohype: Virtualized Cloud Infrastructure Without The Virtualization," In Isca '10 Proceedings Of The 37th Annual International Symposium On Computer Architecture, Saint-Malo, France, 2010.

[36] M. Alouane And H. E. Bakkali, "Virtualization In Cloud Computing: Existing Solutions And New Approach," In 2016 2nd International Conference On Cloud Computing Technologies And Applications (Cloudtech), Marrakech, Morocco, 2016.

[37] K. K. P. S. A. S. V. S. C. W. Sushil Jajodia, "Secure Cloud Computing," In Secure Cloud Computing , Nca, Springer, 2014, Pp. 60-63.

[38] M. Turkanović, T. DružOvec And M. HöLbl, "Inference Attacks And Control On Database Structures".

[39] C. Anglano, R. Gaeta And M. Grangetto, "Securing Coding-Based Cloud Storage Against Pollution Attacks," 2017.