ISSN 2278-7763

An analysis of Security Attacks on Cloud wrt SaaS

Ms. Shaheen Ayyub, Mr. Praveen Kaushik
Research Scholar MANIT, Bhopal India, Assisstant Prof. MANIT,Bhopal,India
shaheenayyub@gmail.com, pk_kaushik@rediffmail.com

Abstract - *With the development and application of Cloud Computing in recent years, Cloud Storage, as the module which provides data storage service in the Cloud Computing architecture, has become the kernel component of Cloud Computing. This is one of the advantages of cloud computing to create and store data at remote servers. But this advantage implicitly contains drawback of data security and privacy vulnerabilities. Many algorithms and methodologies are there by which data security in cloud computing can be achieved but at the same time it possesses many security risks. In this paper we identify different security attacks on cloud. More specifically this paper presents an elaborated study of SaaS components' security and determines vulnerabilities and countermeasures.*

Keywords:- Cloud ,Cloud Computing, Cloud Services, Cloud Security, Saas, Security Attacks

1. Introduction

Key to the definition of cloud computing is the "cloud" itself. For our purposes, the cloud is a large group of interconnected computers. These computers can be personal computers or network servers; they can be public or private. For example, Google hosts a cloud Google's cloud is a private one (that is, Google owns it) that is publicly accessible (by Google's users). Cloud computing is the use of hardware and software resources that are delivered as a service over a network. Clouds allow users to pay for whatever resources they use, allowing users to increase or decrease the amount of resources requested as needed. The framework design and characteristics of Cloud Computing imposes a number of security benefits, which include centralization of security, data and process segmentation, redundancy and high availability. While many traditional threats are encountered effectively, a number of security challenges and unchartered risks have been introduced to the clouds. There are many queries that arise as to whether a cloud is secure enough. Considering intruders there
are many kinds of possible attacks , such as Denial of service attacks, Side Channel attacks Authentication attacks, Man-in-the-Middle Cryptographic attacks, Wrapping attacks, Malware-Injection attacks, Flooding attacks, Browser attacks, and also Accountability checking problems**.** There is a critical need to securely store, manage, share and analyze massive amounts of complex (e.g., semi-structured and unstructured) data to determine patterns and trends in order to improve the quality of healthcare, better safeguard the nation and explore alternative energy and to provide solutions to detect top attack types using machine learning techniques. In this paper attempts are made to identify and analyze different types of attacks in cloud computing environment.

2. Evolution of Cloud Services

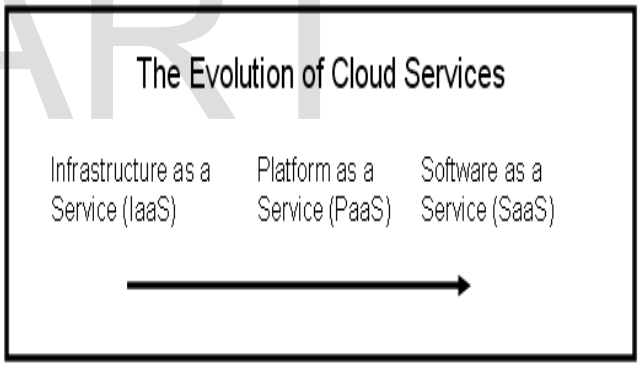

Fig.1 Cloud Services

2.1 Infrastructure as a Service (IaaS): As the name implies we are buying infrastructure. It means that we own the software and are purchasing virtual power to execute it, as needed. Here cloud providers deliver computation resources, storage and network as internet-based services. This service model is based on the virtualization technology.

Characteristics of IaaS[2]

- Resources are distributed as a service.
- Allows for dynamic scaling.
- Has a variable cost, utility pricing model.
- Generally includes multiple users on a single piece of hardware.

ISSN 2278-7763

2.2 Platform as a Service (PaaS): Here cloud providers deliver platforms, tools and other business services that enable customers to develop, deploy, and manage their own applications, *without installing any of these platforms or support tools on their local machines.* Services provided by this model include all phases of the system development life cycle (SDLC) and can use application program interface (APIs), website portals, or gateway software.

Characteristics of PaaS[2]

- Services to develop, test, deploy, host and maintain applications in the same integrated development environment. All the varying services needed to fulfill the application development process
- Web based user interface creation tools help to create, modify, test and deploy different UI scenarios
- Multi-tenant architecture where multiple concurrent users utilize the same development application
- Built in scalability of deployed software including load balancing and failover
- Integration with web services and databases via common standards
- Support for development team collaboration – some PaaS solutions include project planning and communication tools
- Tools to handle billing and subscription management

2.3 Software as a Service (SaaS): where cloud providers deliver applications hosted on the cloud infrastructure as internet based service for end users, without requiring installing the *applications on the customers' computers.* This model is designed to provide everything and simply rent out the software to the user.

Characteristics of SaaS[2]

- Web access to commercial software.
- Software is managed from a central location.
- Software delivered in a "one to many" model.
- Users not required handling software upgrades and patches.
- Application Programming Interfaces (APIs) allow for integration between different pieces of software

3. *Cloud Security Structure with SaaS*

In the SaaS model, enterprise data is stored at the SaaS provider's data center, along with the data of other enterprises. Moreover, if the SaaS provider is leveraging a public cloud computing service, the enterprise data might be stored along with the data of other unrelated SaaS applications. The cloud provider might, additionally, replicate the data at multiple locations across countries for the purposes of maintaining high availability[7]. Most enterprises are familiar with the traditional on-premise model, where the data continues to reside within the enterprise boundary, subject to their policies. Consequently, there is a great deal of discomfort with the lack of control and knowledge of how their data is stored and secured in the SaaS model. There are strong concerns about data breaches, application vulnerabilities and availability that can lead to financial and legal liabilities.

The following figure(2) illustrates the layered stack for a typical SaaS vendor and highlights critical aspects that must be covered across layers in order to ensure security of the enterprise data.

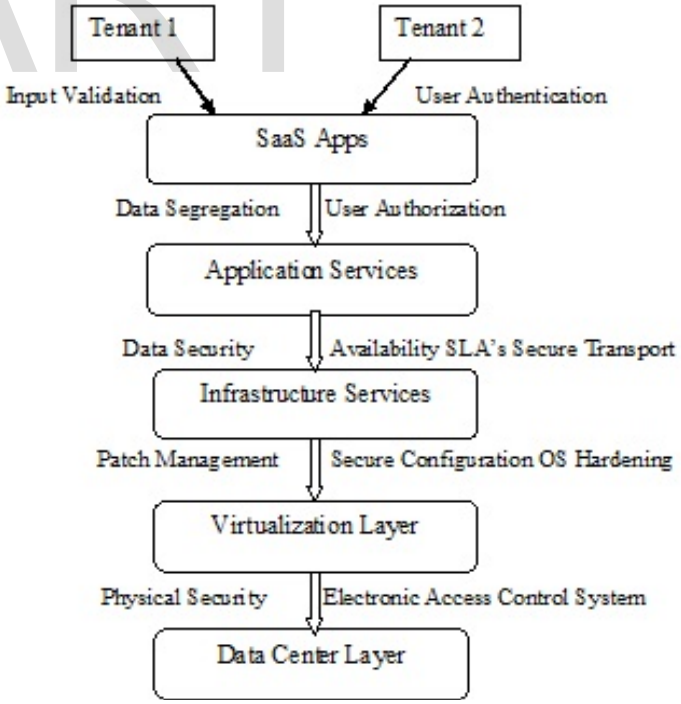

Fig 2: Simple Cloud Security Structure [7]

ISSN 2278-7763

4. Attacks associated with SaaS model

In the SaaS model enforcing and maintaining security is a shared responsibility among the cloud providers and service providers. The SaaS model inherits the security issues associated with IaaS and PaaS model. The most common attacks associated with SaaS model in a public cloud infrastructure are divided into the following four groups :

(i) Availability,
(ii) Data Security,
(iii) Network Security,
(iv) Identity Management

Availability based attacks are:
1) Denial of Service
2) Account Lockout
3) Buffer over Flow

Data Security based attacks are:
1) Cross Site Scripting
2) Access Control Weakness
3) Privilege Escalation

Network Security based attacks are:
1) Network Penetration
2) Session Hijacking
3) Data Packet Interception

Identity Management based attacks are:
1) Authentication Weakness
2) Insecure Trust

(i) Availability based attacks

(a) Denial of Service: which means many nodes systems attacking one node all at the same time with a flood of messages. It is the matter of argument for professionals that the cloud is more vulnerable to DoS attacks, because it is shared by many users, which makes DoS attack much more damaging. When the Cloud Computing operating system notices the high workload on the flooded service, it will start to provide more computational power means more service instances, more virtual machines to cope with the additional workload. Thus, the server hardware boundaries for maximum workload to process do no longer hold [3].In that sense, the Cloud system is trying to work against the attacker by providing more computational power, but actually—to some extent—even supports the attacker by enabling him to do most possible damage on a service's availability, starting from a single flooding attack entry point [3]. The Denial of Service (DoS) attack is focused on making unavailable a resource (site, application, server) for the purpose it was designed. There are many ways to make a service unavailable for legitimate users by manipulating network packets, programming, logical, or resources handling vulnerabilities, among others. If a service receives a very large number of requests, it may stop providing service to legitimate users. In the same way, a service may stop if a programming vulnerability is exploited, or the way the service handles resources used by it.

Sometimes the attacker can inject and execute arbitrary code while performing a DoS attack in order to access critical information or execute commands on the server. Denial-of-service attacks significantly degrade service quality experienced by legitimate users. It introduces large response delays, excessive losses, and service interruptions, resulting in direct impact on availability.

(b) Account Lockout: In an account lockout attack, an attacker attempts to lock out user accounts by purposely failing the authentication process as many times as needed to trigger the account lockout functionality. This in turn prevents even the valid user from obtaining access to their account. For example, if an account lockout policy states that users are locked out of their accounts after three failed login attempts, an attacker can lock out accounts by deliberately sending an invalid password three times. On a large scale, this attack can be used as one method in launching a denial of service attack on many accounts. The impact of such an attack is compounded when there is a significant amount of work required to unlock the accounts to allow users to attempt to authenticate again. EBay is a classic example of how attackers exploit account lockout bugs. Account lockout bugs exist by design. Locking an account after a certain number of failed login attempts protects users from the risk of credential brute-force/dictionary attacks. To fully avoid this vulnerability, the application must apply an account lockout policy that doesn't block a legitimate client from using the application, which is insecure because it opens the application to brute-force/dictionary attacks. Therefore, account lockout bugs can be seen as a necessary side effect of implementing credential brute-force protection. A possible defense against account lockout attacks is to set a loose account lockout policy, but to enforce strong passwords.

(c) Buffer over Flow: A buffer overflow is exactly what it sounds like; the attacker overflows a buffer in the program. In particular, the stack is where a function stores the return address of the function that

ISSN 2278-7763

called it. If a buffer is located on the stack, then an overflow may allow the attacker to overwrite the return address, which will allow the attacker to take over the program. Taking over the program like send mail by the attacker means that the hacker can effectively grant themselves root access to a remote machine.

Consider the following program:

```
int add()     {
1: int x = read ();
2: int y = read ();
3: return x + y;
        }
int read()    {
4: char arr[8];
5: int val;
6: scanf("%s", arr);
...
  }
```

Just before line 6 is executed, the stack, in memory will look something like this: (Fig 3)

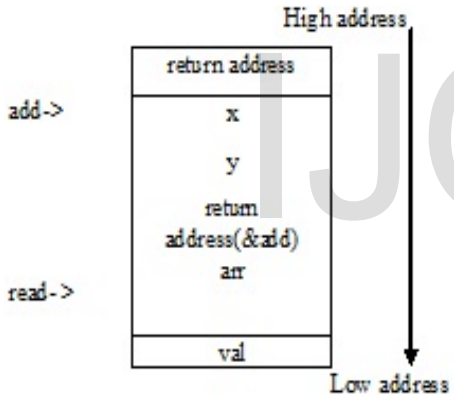

Fig 3: The stack before overflow

Note that arr is a stack allocated character array, with space for only 8 characters. The call to scanf will copy characters from the console into arr. This is fine as long as the user enters only 7 or fewer characters (like"n" or"hello"), but what if the user enters the string "MMMMMMMMMMMM"? Then the buffer is overflowed, and the stack will look like: (Fig 4)

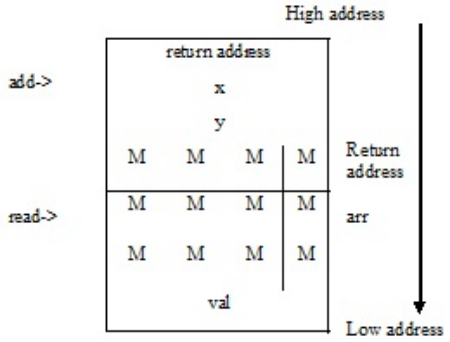

Fig 4: The stack after overflow

At this point, all you've done is just crash the program.

(ii) Data Security based attacks

(a) Cross Site Scripting: Cross Site Scripting is one of the most common application level attacks that hackers use to sneak into web applications today. Cross site scripting is an attack on the privacy of clients of a particular web site which can lead to a total breach of security when customer details are stolen or manipulated. Unlike most attacks, which involve two parties – the attacker, and the web site, or the attacker and the victim client, the CSS attack involves three parties – the attacker, a client and the web site. The goal of the CSS attack is to steal the client cookies, or any other sensitive information, which can identify the client with the web site. With the token of the legitimate user at hand, the attacker can proceed to act as the user in his/her interaction with the site – specifically, impersonate the user.

There are two ways for users to become infected by XSS attacks. Users are either tricked into clicking on a specially crafted link (Non-Persistent Attack) or, unknowingly attacked by simply visiting a Web page embedded with malicious code (Persistent Attack). It's also important to note that a

User's Web browser or computer does not have to be susceptible to any well-known vulnerability. This means that no amount of patching will help users, and we become solely dependent on a website's security procedures for online safety. Browser vendors, software developers and information security professionals working with Web applications are the key to stopping this entirely preventable attack

(b) Access Control Weakness: Access controls are a critical defense mechanism within the application

ISSN 2278-7763

because they are responsible for making the decision of whether it should permit a given request to perform its attempted action of access the resources that it is requesting. When they are defective, an attacker can often compromise the entire application, taking control of administrative functionality and accessing sensitive data belonging to every other user. Access control attacks are among the most commonly encountered categories of web application vulnerability. There are two types of access controls firstly Vertical access control and the secondly Horizontal access control.Vertical Access Controls allow different types of users to access different parts of the application's functionality. Horizontal Access Controls allow users to access a certain subset of a wider range of resources of the same type. Access controls are broken if any user is able to access functionality or resources for which he is not authorized.

(c) Privilege Escalation: A privilege escalation attack is a type of network intrusion that takes advantage of programming errors or design flaws to grant the attacker elevated access to the network and its associated data and applications.
Not every system hack will initially provide an unauthorized user with full access to the targeted system. In those circumstances privilege escalation is required. There are two
kinds of privilege escalation: vertical and horizontal.

- Vertical privilege escalation requires the attacker to grant himself higher privileges. This is typically achieved by performing kernel-level operations that allow the attacker to run unauthorized code.
- Horizontal privilege escalation requires the attacker to use the same level of privileges he already has been granted, but assume the identity of another user with similar privileges. For example, someone gaining access to another person's online banking account would constitute horizontal privilege escalation.

(iii) Network Security based attacks

(a) Network Penetration: It is an attack on a computer system with the intention of finding security weaknesses, potentially gaining access to it, its functionality and data. The process involves identifying the target systems and the goal, then reviewing the information available and undertaking available means to attain the goal.
(b) Session Hijacking: Session Hijacking is when a hacker takes a control of a user session after the user has successfully authenticated with a server. It involves an attack identifying the current session IDs of a client/server communication and taking over the client's session. Session Hijacking involves the following three steps to an attack:

(i) Tracking the Session: In this the hacker identifies an open session and predicts the sequence number of the next packet.

(ii) Desynchronizing the Connection: The hacker sends the valid user's system a TCP reset or finish packet to cause them to close their session.

(iii) Injecting the Attacker's Packet: The sends the server a TCP packet with the predicted sequence number and the server accept it as the valid user's next packet.

(iv) Identity Management based attacks

(a) Authentication weakness: Authentication is a weak point in hosted and virtual services and is frequently targeted. There are many ways by which a user can be authenticated. The mechanisms used to secure the authentication process and the methods used are a frequent target of attackers. If we talk about Saas, IaaS and Paas architecture, there is only IaaS which provides this type of protection and data encryption.

(b) Insecure Trust: Identity management refers to the process of representing and recognizing entities as digital identities in computer networks. Different identity management models will have different trust requirements. Since there are costs associated with establishing trust, it will be an advantage to have identity management models with simple trust requirements.

5. Conclusion

This paper explains SaaS specific security challenges and how contemporary security testing can ensure that the challenges are met. Just as there are advantages to cloud computing, there are also several key security issues to keep in mind. Security of any cloud-based services must be closely reviewed to understand what protections our information has. There is also the issue of availability. This availability could be jeopardized by a denial of service or by the service provider suffering a failure or going out of business.

ISSN 2278-7763

6. References

1) Mohamed Al Morsy, John Grundy and Ingo Müller "An Analysis of The Cloud Computing Security Problem**"** In Proceedings of APSEC 2010 Cloud Workshop, Sydney, Australia, 30th Nov 2010.

2) "Understanding The Cloud Computing Stack SaaS, Paas, IaaS", Diversity Limited, 2011 Non-commercial reuse with attribution permitted.

3) "Overview of Attacks on Cloud Computing **"** ISSN: 2277-3754 International Journal of Engineering and Innovative Technology (IJEIT) Volume 1,Issue 4, April 2012

4) Audun Jøsang1, John Fabre2, Brian Hay2, James Dalziel3 , Simon Pope1,"Trust Requirements in Identity Management" Distributed Systems Technology Centre Australasian Information Security Workshop 2005 (AISW 2005), Newcastle, Australia. Conferences in Research and Practice in Information Technology, Vol. 44.

5)Jayanti Vemulapati, Neha Mehrotra & Nitin Dangewal, "SaaS Security Testing: Guidelines & Evaluation Framework",11th Annual International Software Testing Conference-2011.

6) Maneesha Sharma, Himani Bansal, Amit Kumar Sharma, " Cloud Computing: Different Approach & Security Challenges", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012 421

7) Pradnyesh Rane, "Securing SaaS Applications: A Cloud Security Perspective for Application Providers",Information system Security.

