Attacks on cloud computing and its countermeasures

Conference Paper · October 2016

DOI: 10.1109/SCOPES.2016.7955539

Asma Shaikh
Marathwada Mitra Mandal's College of Engineering

Attacks on Cloud Computing and its Countermeasures

Mrs. Asma A. Shaikh, ME Computer Engg.
Assistant Professor, Department of MCA,
Marathwada Mitra Mandal's College of Engineering,
Pune , Maharashtra, India
asmamokashi@mmcoe.edu.in

***Abstract*— Now with advent of internet, the Cloud Computing has been revolutionized by opening new horizon at a global level with promising opportunities. Cloud Computing is the important buzzword in the *today's* world of Computer.**

Cloud is an International Collection of Hardware and Software from hundreds of thousands of private and public computer network. Cloud is global platform that permits digital information to be shared and distributed at very less cost and very fast to use.

□ ith the rise of popularity, opportunities and its public connectivity via the Internet it is the ne□t leading edge for tro□an, viruses, worms, hackers and cyber□terrorists to start probing and attacking. □iruses, worms, hackers and cyber attacks will enlarge because organized criminals, terrorist and hostile nations would see this as a new frontier to try to steal confidential information, interrupt services and route damage to the enterprise cloud computing network.

□his paper will discussed the different attacks like □ eb Security □ttack, □rowser Security □ttack, Cloud □ alware In□ection □ttack, □looding □ttack and their countermeasure.

Keywords*— *SOAP, TLS, Cloud Computing.

I. INTRODUCTION

Cloud computing is a structure or model which is available everywhere and provide convenient and on-demand network access to a shared pool of computing resources (e.g., networks, servers, storage, applications, and services) that can be speedily provisioned and released with least management endeavor or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models." [7]

Cloud computing characteristics:

A] On-demand self-service: Cloud customers can demand computing capabilities such as network storage, hardware and software.

B] Broad network access.: The cloud capabilities are available over the network and are accessed by customers using platforms (e.g.: laptop, PDA).

C] Resource pooling. : Cloud provider's computing resources are club together to support multiple users model

D] Rapid elasticity : The capabilities can be rapidly and elastically demanded . The capabilities are appeared to be infinitely available to the customers and can be purchased at any time.

E]Measured service :Cloud system automatically controls and optimizes the resources usage by leveraging metering capability to the specific type of service.

TABLE I Cloud services delivery model

Sr. No.	Name of Service	□□amples
1	Software As Service	Google Docs MobileMe Zoho
2	Platform As Service	Microsoft Azure Google App Engine Force.com
3	Infrastructure As Service	Amazon EC2, S3 Rack space Moss Offering Sun's Cloud Services Terre mark cloud Offering

Security Attacks on Cloud:

1. Web Security Attack
2. Browser Secuity Attack
3. Cloud Malware Injection Attack
4. Flooding Attack

II. SECURITY ATTACK ON CLOUD

□. *Web Security Attack*

SOAP, Simple Object Access Protocol provides lightweight and simple mechanism for transferring structured and typed information between terminals in a decentralized, distributed environment using XML. SOAP does not itself define any application semantics structure such as a programming framework.

1. XML Signature attack

A well known type of attacks on protocols using XML Signature for authentication or integrity protection is XML

978-1-5090-4620-1/16/$31.00 ©2016 IEEE

Signature Element. This of course applies to Web Services and therefore also for Cloud Computing.

Figures 1 and 2 show a simple example for a wrapping attack to illustrate the concept of this attack. The first figure presents a SOAP message sent by a valid client. The SOAP body contains a request for the file “me.jpg” and was signed by the sender. The signature is enclosed in the SOAP header and refers to the signed message fragment using an XPointer to 1. If an attacker eavesdrops such a message, he can perform the following attack. The original body is moved to a newly inserted wrapping element inside the SOAP header, and a new body is created. This body contains the operation the attacker wants to perform with the original sender’s authorization, here the request for the file “tender.doc”. The resulting message still contains a valid signature of a legitimate user, thus the service executes the modified request.

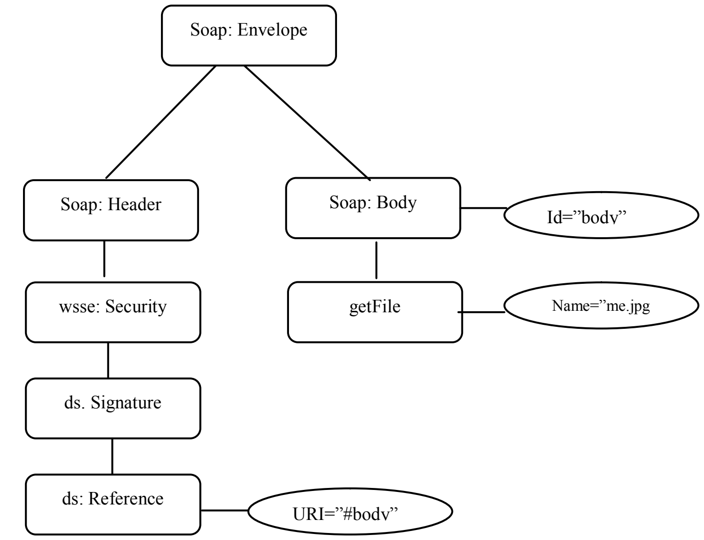

Fig. 1.Example SOAP message before attack[1]

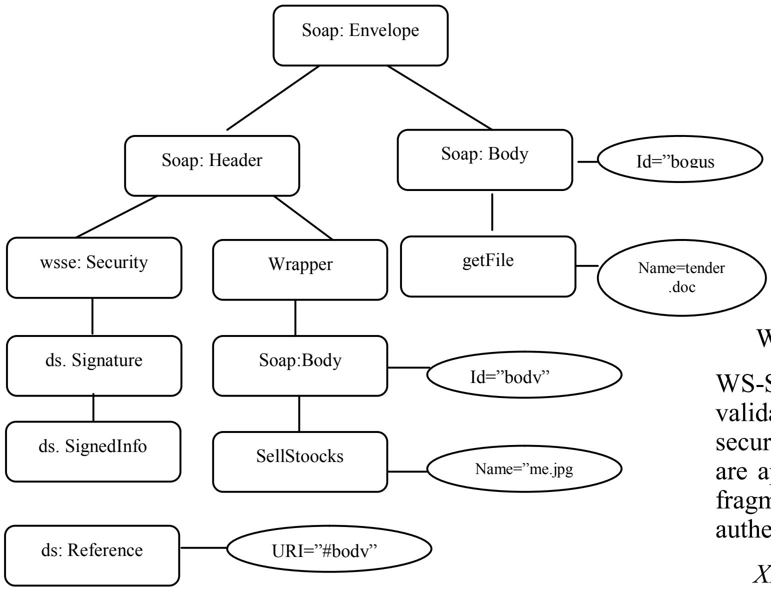

Fig 2: Example SOAP message after attack[1]

Figures 1 and 2 show a simple example for a wrapping attack to illustrate the concept of this attack. The first figure presents a SOAP message sent by a valid client. The SOAP body contains a request for the file “me.jpg” and was signed by the sender. The signature is enclosed in the SOAP header and refers to the signed message fragment using an XPointer to 1. If an attacker eavesdrops such a message, he can perform the following attack. The original body is moved to a newly inserted wrapping element inside the SOAP header, and a new body is created. This body contains the operation the attacker wants to perform with the original sender’s authorization, here the request for the file “tender.doc”. The resulting message still contains a valid signature of a legitimate user, thus the service executes the modified request.

COUNTERMEASURE:

1. The possible countermeasure would be using a combination of WS- Security with XML signature to sign particular element and digital certificated such as X.509 issued by trusted Certificate Authorities (CAs).

2. The web service server side should create a list of elements that is used in the system and reject any message which contains unexpected messages from clients.

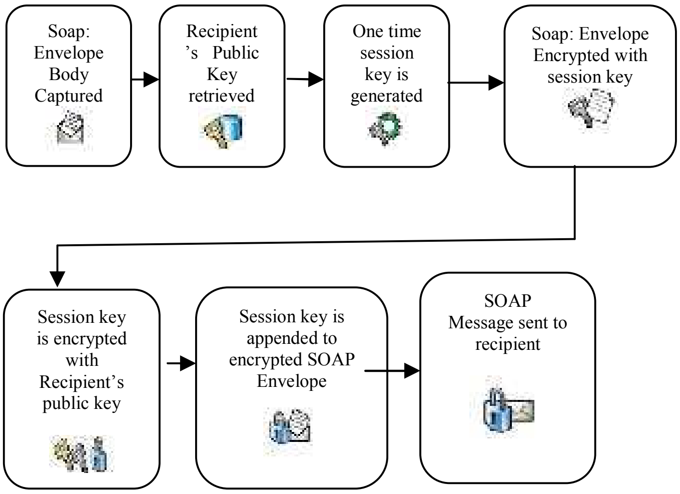
Soap: Envelope Body Captured
Recipient ’s Public Key retrieved
One time session key is generated
Soap: Envelope Encrypted with session key
Session key is encrypted with Recipient’s public key
Session key is appended to encrypted SOAP Envelope
SOAP Message sent to recipient

Fig 3. Countermeasure for XML signature Attack

WS-Security:

WS-Security defines how to provide reliability, secrecy and validation for SOAP messages. It defines how existing XML security standards like XML Signature and XML Encryption are applied to SOAP messages. XML Signature allows XML fragments to be digitally signed to ensure integrity or to proof authenticity.

XML SIGNATURES:

The XML Signature element has the following structure:

```
<Signature>
<SignedInfo>
<CanonicalizationMethod
Algorithm="..."/>
<SignatureMethod Algorithm="..."/>
<Reference URI="..." >
<DigestMethod Algorithm="...">
<DigestValue>...</DigestValue>
</Reference>
</SignedInfo>
<SignatureValue>...</SignatureValue>
</Signature>
```

The signing process works as follows: For every message part to be signed a Reference element is created and this message part is canonicalized and hashed. The resulting digest is added into the Digest Value element and a reference to the signed message part is entered into the URI attribute. Finally the SignedInfo element is canonicalized and signed. The result of the signing operation is placed in the Signature Value element and the Signature element is added to the security header.

XML Encryption:

It allows XML parts to be encrypted to ensure data confidentiality. The encrypted XML part is replaced by an Encrypted Data element containing the cipher text of the encrypted data as content. XML Encryption defines an Encrypted- Key element for key transfer purposes. The most common application for an encrypted key is a hybrid encryption: an XML part is encrypted with a randomly generated symmetric key, which itself is encrypted using the public key of the message recipient. In SOAP messages, the Encrypted Key element must appear inside the security header.

In addition to encryption and signatures, WS Security defines security tokens suitable for transportation of digital identities, e.g. X.509 certificates.

B. *Browser Security Attack :*

Transport Layer Security TLS offers lots of different options for key agreement, encryption and authentication of network terminals, but most frequently the following configuration is used:

The Web server is configured with a X.509 certificate that includes its domain name. This certificate must be issued from a “trusted” certification authority (CA), where “trusted” means that the root certificate of this CA is included in nearly all Web browsers.

During the TLS Handshake, the server sends this certificate to the browser. The browser checks that the certificate comes from a “trusted” CA, and that the domain name in the certificate matches the domain name contained in the requested URL. If both checks succeed, the browser continues loading the Web page.

In a Cloud, computation is done on remote servers. The client PC is used for taking input data from user and return output only If we take into account TLS, which is used for authentication and authorization of commands to the Cloud. This style has been observed during the last few years, and has been categorized under different names: Web applications, Web 2.0, or Software-as-a-Service (SaaS). Modern Web browsers with their AJAX techniques (JavaScript, XMLHttpRequest, Plugins) are ideally suited for Input and output operations. Web browsers can not directly make use of XML

Signature or XML Encryption: data can only be encrypted through TLS, and signatures are only used within the TLS handshake. For all other cryptographic data sets within WS-Security, the browser only serves as a passive data store.

.

COUNTERMEASURE:

Security Assertion Markup Language(SAML)is an XML base open standard data format to exchange authentication & authorization data. From previous work, we identified four methods to protect SAML tokens with the help of TLS.

A] In this approach, the SAML token is sent inside an X.509 client certificate. The SAML token thus replaces other identification data like distinguished names. The certificate has the same validity period as the SAML token.

X.509 Certificate
SAML TOKEN

Fig 4. Wrapping SAML token inside X.509 Certificate

B] In this approach, TLS with client authentication is used, but the client certificate does not transport any authorization information. Instead, the SAML token is bound to the public key contained in this certificate.

C] Whereas the previous approaches relied on the server authenticating the client, in this approach we strengthen the client to make reliable security decisions. This is done by using the server’s public key as a basis for decisions of the Same Origin Policy, rather than the insecure Domain Name System.

D] By binding the token to a certain TLS session, the server may deduce that the data he sends in response to the SAML token will be protected by the same TLS channel, and will thus reach the same (anonymous) client who has previously sent the token.

C. *Cloud Malware Injection Attack*

A main job of a Cloud Computing system consists in maintaining and coordinating instances of virtual machines (IaaS) or explicit service implementation modules (PaaS).

A client request any service from the cloud computing, cloud system determine and instantiate a instance of requested service implementation type. Then those instances are communicated and send to requested client. For the identification purpose some metadata is maintained. This metadata covers all WSDL web service description document related to specific service implementation for the PaaS case of Web service provided by the cloud.

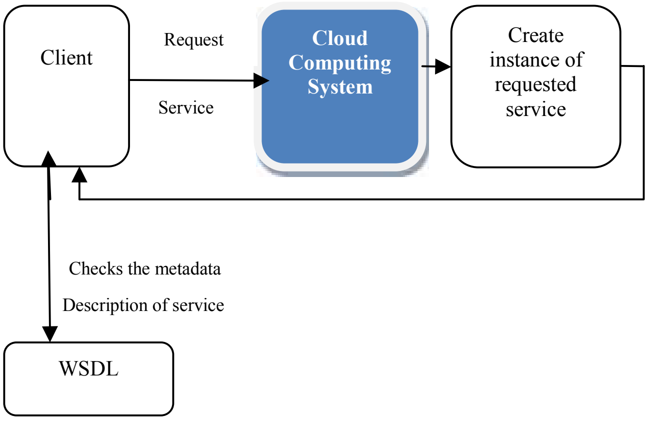

Fig 5. Cloud Computing Before Attack

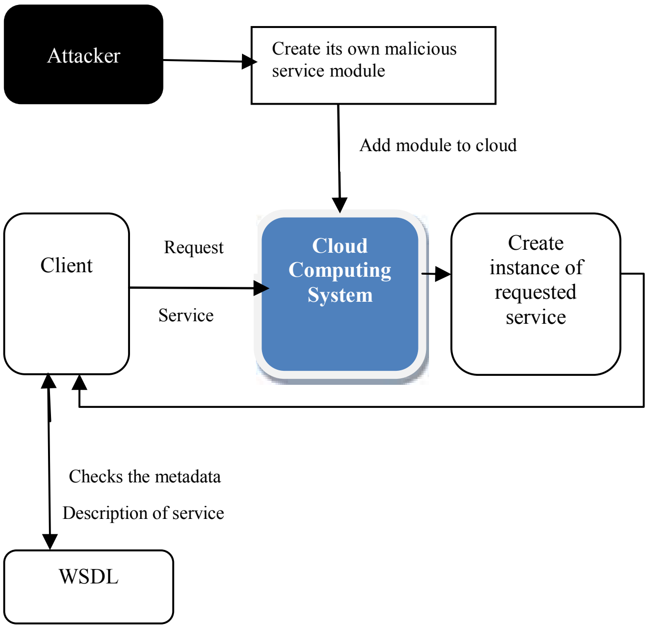

Fig 6. Cloud Computing After Attack

The Web Service description document itself (the WSDL file) should not only be present within the service implementation instance, but also be provided by the Cloud system in order to deliver it to its users on demand. To invoke service of cloud system, any user checks the metadata description of service to determine appropriateness of a service for a specific purpose.

Additionally, these descriptions also represent some preliminary service identifiers, as assembly service implementations with identical WSDL descriptions provide the same functionality. Thus, these metadata should be stored outside of the Cloud system, resulting in a necessity to maintain the correct association of metadata and service implementation instances.

A first significant attack attempt aims at injecting a malicious behaviors implementation on virtual machine into the Cloud system. Such kind of Cloud malware could serve any particular purpose the adversary is interested in, ranging from eavesdropping via subtle data modifications to full functionality changes or blockings. This attack requires the adversary to create its own malicious service implementation module (SaaS or PaaS) or virtual machine instance (IaaS), and add it to the Cloud system. Then, the adversary has to trick the Cloud system so that it treats the new service implementation instance as one of the valid instances for the particular service attacked by the adversary. If this succeeds, the Cloud system automatically redirects valid user requests to the malicious service implementation, and the adversary's code is executed.

COUNTERMEASURE:

The possible countermeasure for this type of attack could be performing a service instance integrity check for incoming requests. A hash value can be used to store on the original service instance's image file and compare this value with the hash values of all new service instance images. As a result of using the hash values, an attacker is required to create a valid hash value comparison in order to trick the cloud system and inject a malicious instance into the cloud system.

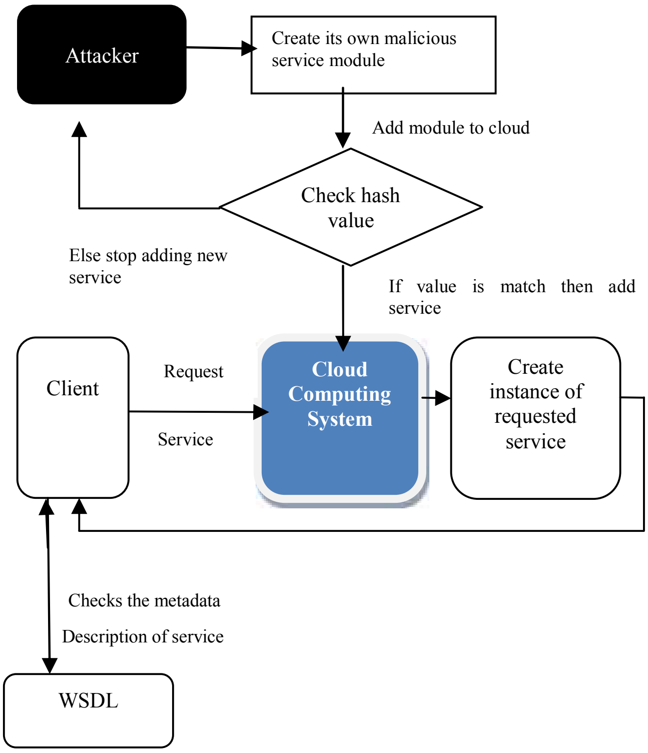
Attacker
Create its own malicious service module
Add module to cloud
Check hash value
Else stop adding new service
If value is match then add service
Client
Request
Service
Cloud Computing System
Create instance of requested service
Checks the metadata
Description of service
WSDL

Fig 7. Countermeasure

D. Flooding Attack

One of the common characteristics of the cloud system is to provide dynamically scalable resources.

Once there are more requests from clients, cloud system automatically scale up by starting up new service instances in order to support the clients' requirements.

On the other hand, this also can be a severe vulnerability of flooding attack such as DoS, which, basically, is an action of sending a large number of nonsense requests to a certain service.

COUNTERMEASURE:

Even though it is difficult to completely prevent DoS attacks, installing a firewall or intrusion detection system (IDS) is able to filter malicious requests from attacking the server.

III. SUMMARY

Sr. No.	**Name of the Attack**	**Countermeasures**
1	XML Signature attack	1. A combination of WS-Security with XML signature 2. The web service server side should create a list of elements that is used in the system and reject any message which contains unexpected messages from clients.
2	**Browser Security Attack**	1.TLS with x509 certificate 2.SAML with Public key 3.Using server's public key 4.Session binding
3	***Cloud Malware Injection Attack***	To perform a service instance integrity check for incoming requests by hash value
4	**Flooding Attack**	Installing a firewall or intrusion detection system (IDS) is able to filter malicious requests from attacking the server.

IV. CONCLUSION

In this paper, we find issues of cloud computing security, XML Signature Element Wrapping, Browser Security, Cloud Malware Injection Attack and Flooding Attacks, and its potential countermeasures. Attacker always tries to find new way to attack cloud so it appears that the security concern of cloud computing is most important in this era.

REFERENCES

[1] Meiko Jensen, Jorg Schwenko, Nils Gruschka, Luigi Lo Iacono , "On Technical Security Issues in Cloud Computing " in 2009 IEEE International Conference on Cloud Computing

[2] M. McIntosh and P. Austel, "XML signature element wrapping attacks and countermeasures," in SWS '05: Proceedings of the 2005 workshop on Secure web services. ACM Press, 2005, pp. 20–27.

[3] Google, "Browser security handbook," 2009. [Online] Available: http://code.google.com/p/browsersec/

[4] D. Kormann and A. Rubin, "Risks of the passport single signon protocol," Computer Networks, vol. 33, no. 1–6, pp. 51–58, 2000.

[5] Farzad Sabahi , " Cloud Computing Threats and Responses" in 978-1-61284-486-2/111$26.00 ©2011 IEEE

[6] Ram Kumar Singh, Aniruddha Bhattacharjya "Security and Privacy Concerns in Cloud Computing" International Journal of Engineering and Innovative Technology (IJEIT) Volume 1, Issue 6, June 2012 ISSN: 2277-3754

[7] Peter Mell , Timothy Grance , "The NIST Definition of Cloud Computing " Special Publication 800-145 NIST

[8] Madhu Chauhan, Riidhei Malhotra, Mukul Pathak and Uday Pratap Singh, "Different aspects of Cloud Security" in International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2, Mar-Apr 2012, pp.864-869.