Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 09, September-2016

Security in Web Services- Issues and Challenges

Aruna. S
M.Tech Computer Science and Engineering
School of Computing Science and Engineering
VIT University, Vellore

***Abstract*— This paper focuses on the major issues and challenges involved in handling and implementing security in web services. Like any software or web application, web services are also prone to security issues related to authentication, availability and integrity. New and challenging problems related to security arise due to the distributed nature of the web services and their cross platform access and also during service composition. As the web services provide access to the data in an autonomous way, the confidentiality and authenticity of the data transmitted through them attains more importance. In the recent years, many technologies and standards have emerged in order to handle the security issues related to web services. However new threats and attacks related to web services are also coming to forefront. Therefore, a study on the existing standards and protocols for security is carried out. The challenges that arise are also discussed.**

Index Terms*— *Composition, REST, SAML, SOAP, XML, WS-Security

1. INTRODUCTION

A web service is a collection of open protocols and standards used for exchanging data between applications or systems. They are self-contained, modular applications that can be described, published, located and invoked over a network, generally the World Wide Web [IBM]. These applications can be local, distributed, or web-based. Web services are built on top of open standards such as TCP/IP, HTTP, Java, HTML, and XML. They provide a way of communication between the applications using SOAP, XML, and WSDL. Web Services essentially involve the three roles of the Service Oriented Architecture: service provider, service requester and service broker. A service provider could be an industry, business or a company capable of providing service. A requester also could be a company or a business that is in need of the service, where as the broker is a place, entity or a system that helps both service provider and service requester to discover each other. Web services are both platform and language independent service components can be exposed using a standard Web Services Description Language (WSDL) and registered at Universal Discovery Description and Integration (UDDI) registries. The Universal Description, Discovery and Integration (UDDI) specification provides a set of services that assist in discovering or inquiring about the availability of Web services. Web Services Description Language (WSDL), is "an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information". This WSDL file can be sent directly to perspective users, or published in the UDDI registries. Simple Object Access Protocol (SOAP) is used for communication among different Web Services. It is a lightweight protocol for exchanging structured information in the implementation of web services. SOAP is defined independently of the underlying messaging transport mechanism in use. It allows the use of many alternative transports for message exchange. Despite this general transport independence, most first-generation Web services communicate using HTTP, because it is one of the primary bindings included within the SOAP specification. This paper focuses on the security issues related to web services. The essential security requirements of any web based application; the threats faced by web services, major attacks are discussed in elaborate detail. The existing technologies for implementing the concepts are also highlighted.

2 RELATED WORK

Web services involve combining several systems and resources. To provide an end-to-end security in such a dynamic environment requires the combination of several technologies. In 2012 a survey on the major issues concerning web service security was published [1]. The attacks on web services can range from denial of service to malicious code injection and session hijacking. Sindhu et.al [2] has provided solutions for WS-Address spoofing and SOAP action spoofing with respect to a dynamic composition scenario. However such solutions should ne converted to APIs in order to extend the functionality to XML as well. Azzam Mourad [3] et.al has proposed a new approach to provide security to web services in a dynamic environment, during composition. They have used Aspect Oriented Programming (AOP) and BPEL (Business Process Execution Language) where in the security and the businesses are separated and composition is allowed to be modified during run time. Several security measures are to be considered while dealing with interoperability of web services. These have been discussed by a research paper of GE Energy [4]. The major security attacks that can happen to a web based application, particularly web services has been discussed by Ladan et.al and the current technologies to provide security are also mentioned [5]. Wu, X.et.al have proposed a security model for service-oriented multi-application architecture and it provides a comprehensive layer of security when the integrated systems are constructed [6]. WSDL files, which define the basic structure of the web service, suffer from tampering, scanning and other attacks. Mirtalebi et.al [7] has proposed a model for ensuring the security of the WSDL files by encrypting them. The main security issues related to Restful web services are analyzed and a secure XAuth mechanism has been proposed by Pan Et.al [10] for Restful WCF Services.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 09, September-2016

3 SECURITY IN WEB SERVICES

Like any distributed application, web services also require major security mechanisms to ensure secure transfer of data. When sharing existing information or functionality through web services, it is well known that we do not depend on programming languages, architectures or systems. Due to this interoperable capability and cross platform access of web services, it demands a greater attention of security concepts. The basic security requirements of any web based application are Authentication, Authorization, Confidentiality, Integrity, Availability and Non-Repudiation.

- Authentication: It is the process to identify the user. When we use this concept, we are trying to ensure the identity of the user and we verify the identity that the user claims to be.
- Authorization: It is the process of giving permission to the user to do something. Authorization is often seen as both the introductory setting up of permissions by a system administrator and the checking of the permission values that have already been set up when a user is getting access.
- Confidentiality: In the world of information security, confidentiality is used to refer to the requirement for data in transit between two communicating parties not to be available to a third party, to avoid snooping. The broad approach is to use either a Virtual Private Network (VPN) or encryption.
- Integrity: It means that if the information is tampered with, it can be detected. The process often relies on mathematical algorithms known as hashing algorithms.
- Availability: It requires that the resources and the services should be available to the authorized parties at all times. The common attack to the availability of data is the Denial of Service attack. It aims to use up all the resources of the service so that they are unavailable to the legitimate users.
- Non-repudiation: The originator of a message cannot claim not to have sent the given message.

The above security requirements are implemented in web services in the following ways:

Authentication techniques: Basic authentication is used in almost all the applications. Before gaining functionality to the application, the user is requested a username and password. Both are validated. The main drawback of the implementation is that the credentials are propagated in a plain way from the client to the server. Any sniffer could read the sent packages over the network. Security Assertion Markup Language (SAML) is used to provide strong authentication and authorization tokens. SAML, developed by OASIS, is an open framework for sharing security information on the Internet through XML documents. The user requests a service from the service provider. The service provider requests and obtains an identity assertion from the identity provider. On the basis of this assertion, the service provider can make an access control decision – in other words it can decide whether to perform some service for the connected user. Before delivering the identity assertion to the Service

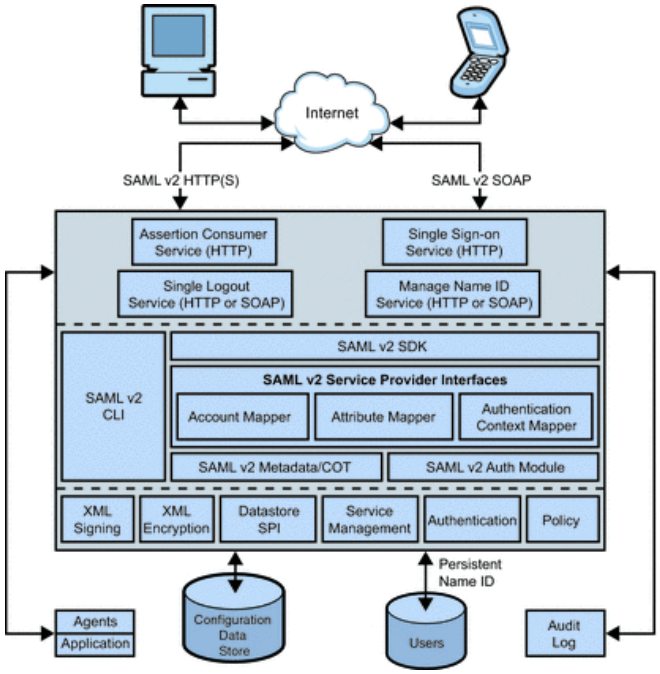

Fig. 1 SAML Architecture (Provided by Sun OPENSSO Enterprise)

Provider, the identity provider may request some information from the user– such as a user name and password.

The digest authentication encrypts the password entered by the user, before sending to the server. Typically a hash function is used such as SHA or MD5. The client makes an unauthenticated request to the Web service, and the server sends a response with a digest authentication challenge indicating that it supports digest authentication. The client generates a nonce and sends it to the service along with a timestamp, digest, and username. The digest is a cryptographic hash of the password, nonce, and timestamp. The server generates the hash itself from the password (retrieved from the service store), nonce and timestamp (from the message), and if the generated hash matches the hash in the request, the request is allowed. The advantage of digest authentication is it is resistant to replay attacks.

Authentication can also be provided using certificates. When the client attempts to access a protected web service resource, it provides a certificate to the server which contains the credentials as well as a unique private-public key pair. The server verifies the same. This mechanism must use HTTPS as there is no secure channel for communication.
Kerberos token is a cross-platform authentication and single sign-on system. The Kerberos protocol provides mutual authentication between two entities relying on a shared secret (symmetric keys).

Authorization techniques: XACML (eXtensible Access Control Markup and SAML Language) are the two technologies that are used to ascertain authorization information. Using SAML, an authorization decision assertion involves making a decision about whether or not a principal can access a specific resource, given an authentication assertion and attribute asser-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 09, September-2016

tion. According to a policy, the entities Policy Decision Point (PDP) and Policy Enforcement Point (PEP) make and enforce authorization decisions respectively. SAML provides a standard way to represent a security token that can be passed across the multiple steps of a business process or transaction, from browser to portal to networks of web services, also a feature supported by OWSM. The authorization basically involves Role based Access Control (RBAC) and Context Based Access Control (CBAC). RBAC maps security management to an organization's structure. With RBAC, each user can be assigned privileges based on the role of the user in the organization.

Confidentiality Techniques: XML encryption is the technique used for implementing this feature of security management. It is used if information in a SOAP message must be kept confidential while it is sent over a multihop SOAP transaction. Additionally, if information in a SOAP message must be kept encrypted after the SOAP message has been processed by a web service, XML encryption is also useful. XML Encryption is a W3C recommendation. The encrypted data can be expressed using XML or the portions of an XML document can be selectively encrypted. Algorithms like Triple-DES, AES are commonly used.

Integrity Techniques: The WSDL file describes the functionality of the web service. When a service requester communicates with the UDDI registry, the WSDL forms the basis for selecting the web service, particularly during composition. If it is tampered, then it destroys the integrity of the web service. XML Signature may also be used for integrity and non repudiation of WSDL files so that a definition of a web service can be published and later trusted to not have been tampered with. It was developed by the W3C and the IETF. In a web service communication, data integrity may also be present at lower layers of the OSI stack. However when integrity is implemented only for SOAP communication, we cannot call it as persistent. The persistent integrity is useful where a document maybe composed of portions authored by different entities, in a workflow scenario. XML signature guarantees the integrity of certain portions of the document but allows the participants to edit other unsigned portions of the document. Web Services (WS)-Security, produced by OASIS, defines a set of SOAP header extensions for end-to-end SOAP messaging security. It supports message integrity and confidentiality by allowing communicating partners to exchange signed encrypted messages in a Web Services environment.

Non-Repudiation Techniques: WS-Security uses digital signatures in order to guarantee non-repudiation. The specification allows a variety of signature formats, encryption algorithms and multiple trust domains, and is open to various security token models, such as X.509 certificates, Kerberos tickets, SAML Assertions, and custom-defined tokens. It adds security features in the SOAP header and works in the application layer. The X.509 specification is fully extensible and is used to format the information in the digital certificate. A digital certificate typically includes identity information about the entity holding the corresponding private key, a serial number, expiry date and the public key.

4 THREATS AND CHALLENGES

The threats faced by web services can be message level or service level. Service level threats can involve the threats faced by UDDI, WSDL and XML. The following are some of the major service level threats.

- WSDL and UDDI attack: The service level information are available in the WSDL files and UDDI registry. An attacker can access any publicly available WSDL file and tamper with it. The attacks can be in the form of WSDL Scanning or WSDL Tampering. The former scans the WSDL file and exposes the operational information, ports, etc. The latter tampers with the data and can even gain access to confidential information. Protection against these threats is not easy with the typical methods like authentication and authorization.
- Malicious Code Injection and Identity Spoofing: These attacks take place mostly in XML files. An attacker can inject malicious codes and spoil the functionality of the service. Identity spoofing attack occurs when a hacker takes off the identity of either the service requester or the service provider. In the first case, the attacker can compose a well-formed XML request message and post it to the service provider, causing the service provider to assume the response is sent to a valid service requester. In the second case the attacker will trick genuine service requesters to post messages to the fake service provider. Such attacks cannot be detected easily.
- XML Schema Tampering: An attacker can modify the XML schema and make it erroneous. Such threats end up with failures at the end points of the service providers.
- Session Hijacking: An attacker can steal the session token of the user and gain unauthorized access to the resources provided. This leads to sending of false requests or replies and thus the session between the service requester and provider is said to be hijacked.

The message level threats to the web services can be of the following types:

- Message Injection or Alteration: Messages between the client and the server can be altered or new malicious messages can be added. This can happen to XML files and can provide the hacker with many privileges.
- Replay of Messages: An attacker captures a valid message and replays it later to gain sensitive information via unauthorized access to the services. Usually this is the initial step in hacking a web service wherein the hacker can hijack the session and tamper with the services. With the right tools, patterns can be detected more easily even if the same or similar payload is being sent across multiple mediums like HTTP, HTTPS, and SMTP or across different interfaces.
- Message Confidentiality and Eavesdropping: Interception of messages is always a threat to web services. Traditional security mechanisms like VPN or SSL are not sufficient to secure the web services against such threats.

5 EXISTING SECURITY STANDARDS

Under the basic implementation of the security features in web services, XML Signature, XML Encryption, SAML have been discussed. The other major standards are covered in this section.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 09, September-2016

- WS-Security: The protocol is currently officially called WSS and developed via committee in Oasis-Open.The standard defines how security tokens are contained in SOAP messages, the XML security specifications to encrypt and sign the tokens. Tokens are attached to the messages and timestamps are also inserted. Other parts of the SOAP message can also be encrypted and the methods are mentioned in the standard. The main features of the standard include WS-Policy, WS-Trust and WS-Privacy. The WS-Policy allows the organizations to specify the security requirements for their web services. WSDL binding is used in order to attach the policy information to the web services. Ws-Trust defines how trust relationships are established. Trust can be either direct or brokered. In the case of brokered trust, a trust proxy is used to read the policy information and request the appropriate security token for enclosure in the SOAP message. WS-privacy explains how the privacy requirements can be included in the policy and the WS-Trust is used to evaluate the privacy claims encapsulated within the SOAP messages against the preferences of the user and the organization.
- Security in Restful Web Services: The major security mechanisms in a web service based on REST are Client security and OAuth. Client security can provide the basic authentication and authorization features. Normally Basin HTTP Authentication and HTTP Digest Authentication are used. OAuth is a specification that defines secure authentication model on behalf of another user. OAuth in general is widely used in popular social Web sites in order to grant access to a user account and associated resources for a third party consumer (application). The consumer then usually uses RESTful Web Services to access the user data. OAuth 1 protocol is based on message signatures that are calculated using specific signature methods. Signatures are quite complex and therefore are implemented in a separate module. Jersey provides the packages for both OAuth1 and 2. OAuth 2, in contrast with OAuth 1, is not a strictly defined protocol, rather a framework. OAuth 2 specification defines many extension points and it is up to service providers to implement these details and document these implementations for the service consumers. Additionally, OAuth 2 defines more than one authorization flow. XAuth along with authentication manager has been implemented for Restful WCF Services. Currently server side streaming technologies are used in order to counter Denial Of Service (DoS) attacks.

Fig. 2 The web service security stack [18]

TABLE 1
SECURITY STANDARDS AND APPLICATIONS

STANDARD	APPLICATION
SSL(Transport level security)	Authentication, confidentiality, and secure key exchange.
XML Encryption	Confidentiality
SAML	Authorization
XML Signature	Integrity, Authentication
Kerberos, HTTP Digest	Authentication
WS-Security	Confidentiality, Authentication

TABLE 2
CURRENT THREATS AND ATTACKS [23]

Web service layers	Current threats and attacks
Web services in transit	In transit sniffing or spoofing, WS-Routing security threats, Replay attacks
Web services engine	Buffer overflow, XML parsing attacks, Denial of services
Web services deployment	Fault code leaks, Permission and access issues, Poor policies, Authentication and Certification
Web services user code	WSDL probing, Bruteforce attacks, Content spoofing, Session tampering, Information leakage, Authorization issues

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 09, September-2016

In order to handle the security concerns measures can be taken to ensure that the security framework is deployed well. The deployment analysis involves three broad divisions. In-transit, vendor-controlled and deployment shell. The In-Transit security involves provodong the security for the SOAP messages which are exchanged. We can use Secure Socket Layer (SSL) for secure communication as well as WS-Security for the services. IP security and access controls should be provided effectively. Foe the deployment shell security, we need to ensure that the deployed services are secure enough in order to prevent the information leakage. Errors and exceptions should be handled well so that the attack will not find the loop holes which make the services more vulnerable. Auditing and logging helps to identify and take preventive measures against security breaches. A few guidelines can be followed in order to secure the service at a code level. They are as follows:-

- Never trust a user input completely.
- Do not echo the data supplied by the user.
- Do not use the client-side scripting languages for validation [23]

Within the web services, we can follow the additional guidelines. Digital signatures can be incorporated into SOAP messages. XML firewalls can be implemented. All the requests and responses should be encrypted with header information. Validations can be used to prevent Cross-Site Scriptong. Exceptions should be handled cleanly and passwords and other important fields should be encrypted in order to prevent input parameter tampering. Measure should be taken to prevent attacks that come through HTTP requests.

6 CURRENT CHALLENGES AND THREATS

The current challenges involve security based on the end user, maintaining security while routing between multiple web services and the challenge of abstracting security from the underlying network. The information or details of the end user might not be available to the web service. They will be available to the web site or any client that accesses the web service on behalf of the user. This challenge can be addressed by adding the information in the SOAP message itself. This might prevent the need for the user to re-authenticate every time when a SOAP request is sent. Single Sign-on and Federated trust come under this category. Web service routing indicates the traversal of SOAP messages over multiple hops before reaching the end point. While routing between the web services it is mandatory to maintain the confidentiality of the data from the SOAP intermediaries. Traditional security mechanisms work in the transport layer(SSL) whereas SOAP functions in the application layer. Hence, better techniques have to be developed in order to ensure the security in the SOAP gaps created by the intermediaries. In order to meet the challenges recently the software which is gaining popularity is Acunetix. Acunetix Web Vulnerability Scanner is a feature-packed solution for detecting vulnerabilities and securing web applications. The Web Services Security scanning tool will allow you to run an automated vulnerability assessment against a Web Service with a more accurate and improved version of the same scanning engine which till now assessed web applications. Another new addition is the Web Services Security Editor which extends the functionality of the Web Services scanner by allowing deeper analysis of XML responses, WSDL structure, WSDL XML analysis, syntax highlighting for all coding languages, and regular expression searching.

Threats are faced by databases as well in the SOA environments. When a database application is exposed as a service or is simply accessed by a Web service, the security of the database relies entirely on the security of the applications directly accessing it. The exposing application should account for confirming the requester's authorization to access the database. Additionally, the application must filter requests before forwarding them on to the database. Incoming requests may include SQL injection attacks or other attempts to subvert the database or retrieve sensitive information. The application must also filter the data before providing it to the requester to prevent unintentional leakage.

7 CONCLUSION

While deploying a web service the major concern is to ensure the security of the resources being offered. As discussed in the paper, the distributed and autonomous nature of the web services has made it vulnerable to a wide variety of threats and attacks. The existing security practices will not be completely sufficient to cater the security requirements. Current trends indicate new attacks on WSDL and XML files which define the basic structure of web services. Due to the interoperable nature of the web services, maintaining security policies and checks against violations and message tampering is a potential issue. Hence it is important to include the security mechanisms in the initial stages itself and allow the co-operating services to follow a similar policy. We can hope that the emerging solutions and technologies will merge the gap between the services and take the operations of the security features to the next level.

REFERENCES

[1] Balasubramanian, N., & Ruba, A. (2012, August). Security: a major threat for web services. In *Advanced Communication Control and Computing Technologies (ICACCCT), 2012 IEEE International Conference on* (pp. 104-109). IEEE.

[2] Sindhu, S. M., & Kanchana, R. (2014, May). Security solutions for Web Service attacks in a dynamic composition scenario. In *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on* (pp. 624-628). IEEE.

[3] Mourad, A., Ayoubi, S., Yahyaoui, H., & Otrok, H. (2010, August). New approach for the dynamic enforcement of Web services security. In *Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on* (pp. 189-196). IEEE.

[4] Gruschka, N., Jensen, M., Iacono, L. L., & Luttenberger, N. (2011). Server-side streaming processing of ws-security. *Services Computing, IEEE Transactions on*, *4*(4), 272-285

[5] Ladan, M. I. (2011, February). Web services: security challenges. In *Internet Security (WorldCIS), 2011 World Congress on* (pp. 160-163). IEEE.

[6] Wu, X., & Li, C. (2011, June). Research and design of one security model for service-oriented multi-application architecture. In *Computer Science and Service System (CSSS), 2011 International Conference on* (pp. 3990-3993). IEEE.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 09, September-2016

[7] Mirtalebi, A., & Khayyambashi, M. R. (2011, August). Enhancing security of Web service against WSDL threats. In *Emergency Management and Management Sciences (ICEMMS), 2011 2nd IEEE International Conference on* (pp. 920-923). IEEE.

[8] Mougouei, D., Rahman, W. N. W. A., & Almasi, M. M. (2012, June). Evaluating fault tolerance in security requirements of web services. In *Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on* (pp. 111-116). IEEE.

[9] Serme, G., De Oliveira, A. S., Massiera, J., & Roudier, Y. (2012, June). Enabling message security for RESTful services. In *Web Services (ICWS), 2012 IEEE 19th International Conference on* (pp. 114-121). IEEE.

[10] Pan, G., & Wang, Y. (2012, June). Securing RESTful WCF Services with XAuth and Service Authorization Manager-A Practical Way for User Authorization and Server Protection. In *Computational Sciences and Optimization (CSO), 2012 Fifth International Joint Conference on* (pp. 651-653). IEEE.

[11] Lee, S., Jo, J. Y., & Kim, Y. (2015, June). Method for secure RESTful web service. In *Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference on* (pp. 77-81). IEEE

[12] Masood, A. (2013, November). Cyber security for service oriented architectures in a Web 2.0 world: An overview of SOA vulnerabilities in financial services. In *Technologies for Homeland Security (HST), 2013 IEEE International Conference on* (pp. 1-6). IEEE.

[13] De Backere, F., Hanssens, B., Heynssens, R., Houthooft, R., Zuliani, A., Verstichel, S., ... & De Turck, F. (2014, May). Design of a security mechanism for RESTful web service communication through mobile clients. In *Network Operations and Management Symposium (NOMS), 2014 IEEE* (pp. 1-6). IEEE.

[14] Babu, B. C., & Kishore Kumar R, C. (2013, December). API based security solutions for communication among web services. In *Advanced Computing (ICoAC), 2013 Fifth International Conference on* (pp. 571-575). IEEE.

[15] Sharifi, M., Movahednejad, H., Tabatabei, S. G. H., & Ibrahim, S. (2009, December). An effective access control approach to support web service security. In *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services* (pp. 529-535). ACM.

[16] Dwivedi, A. K., & Rath, S. K. (2015). Incorporating Security Features in Service-Oriented Architecture using Security Patterns. *ACM SIGSOFT Software Engineering Notes*, *40*(1), 1-6.

[17] Lakshminarayanan, S. (2010). Interoperable security standards for web services. *IT professional*, (5), 42-47

[18] Singhal, A. (2007). Web Services Security: Challenges and Techniques. *POLICY*, *7*, 282-282.

[19] Web Services Security: Beauty and the Beast, Paul Korzeniowski E-Commerce Times, Aug. 2007.

[20] Jo, J., Kim, Y., & Lee, S. (2014, October). Mindmetrics: Identifying users without their login IDs. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on* (pp. 2121-2126). IEEE.

[21] Mark, O. N. (2003). *Web services security*.

[22] Saravanaguru, R. A., Abraham, G., Ventakasubramanian, K., & Borasia, K. (2013). Securing Web Services Using XML Signature and XML Encryption. *arXiv preprint arXiv:1303.0910*.

[23] Shah, S. (2006). *Hacking Web Services (Internet Series)*. Charles River Media, Inc..

[24] Mark, O. N. (2003). *Web services security*.