SERVICES AND MECHANISMS FOR WEB SECURITY

Santosh Kumar [1, 2], Karuna Shankar Awasthi[2] and Laxmi Shankar Awasthi[3]

[1] Ph.D. Research Scholar, Department of Computer Science and Engineering, School of Engineering and Technology, Shridhar University, Pilani-Chirawa Road, Pilani, Rajasthan- 333031 (India).

[2] Assistant Professor, Department of Computer Science, Lucknow Public College of Professional Studies, Vinamra Khand, Gomti Nagar, Lucknow - 226010 (India).

[3] Associate Professor, Department of Computer Science, Lucknow Public College of Professional Studies, Vinamra Khand, Gomti Nagar, Lucknow - 226010 (India).

ABSTRACT: Consumer attitudes toward the website's information quality, trust, privacy issues, reputation, security concerns, and the company's reputation all have a significant impact on Internet users' trust in the site. Privacy and security are two major issues that both e-commerce customers and websites face. Security is the attempted access to data by unauthorized users, whereas privacy is the control over one's personal data. As a result, information security is a critical managerial and technical need for any efficient and effective internet payment transaction activity. Integrity, Privacy, Non-repudiation, Authenticity, Confidentiality, and Availability are the aspects to be investigated in e-commerce security because they safeguard e-commerce assets against illegal access, destruction, alteration, or usage. This paper will discuss E-commerce privacy, security, and its purpose, as well as numerous security options.

KEYWORDS: RISK, INTERNET CONSUMER BEHAVIOUR, PRIVACY AND SECURITY, E-COMMERCE SECURITY ISSUES.

INTRODUCTION: A web service is a set of open protocols and standards that allow data to be exchanged between different applications or systems. They are self-contained, modular applications that may be specified, published, located, and invoked across a network, most commonly the Internet [IBM]. Local, distributed, or web-based applications are all possibilities. TCP/IP, HTTP, Java, HTML, and XML are all open standards that web services are built on. They use SOAP, XML, and WSDL to enable communication between applications. The three roles of the Service Oriented Architecture are essentially involved in Web Services: service provider, service requester, and service broker. A service provider can be an industry, a business, or a corporation that provides services. A requester can also be a firm or a business that requires the service, whereas a broker is a location, entity, or system that facilitates the discovery of both the service provider and the service requester. Web services are platform and language agnostic service components that can be exposed using a common Web Services Description Language (WSDL) and gistered with UDDI registries.The Universal Description, Discovery, and Integration (UDDI) specification defines a series of services that aid in the discovery and inquiry of Web service availability. "An XML format for describing network services as a group of endpoints working on messages carrying either document oriented or procedure-oriented information," according to the Web Services Description Language (WSDL). This WSDL file can be provided to potential users directly or published in UDDI registries. SOAP stands for Simple Object Access Protocol. The SOAP acronym stands for Simple Object Access Protocol, which is used to communicate between Web Services. It's a simple protocol for sharing structured data in the context of web services deployment. SOAP is defined in a way that is agnostic of the messaging transport mechanism in use. It enables for the usage of a variety of different message transports. Despite this, because HTTP is one of the key bindings provided in the SOAP definition, most first-generation Web services interact over it. The focus of this paper is on the security challenges that arise when using web services. The key security requirements of any web-based application are explored in depth, as well as the threats that web services face and major attacks. The available tools for putting the thoughts into practise are also discussed.

RELATED WORK: Web services entail the integration of multiple systems and resources. In such a dynamic context, providing end-to-end security necessitates the use of various technologies. A survey on the key concerns about online service security was published in 2012 [1]. Denial of service attacks, malicious code injection, and session hijacking are all examples of web service assaults. In a dynamic composition situation, Sindhu et al. [2] proposed solutions for WS-Address spoofing and SOAP action spoofing. However, such solutions should be transformed to APIs so that the functionality may be extended to XML as well. Azzam Mourad [3] et al. has suggested a new method for securing web services in a dynamic context during composition. They employed Aspect Oriented Programming (AOP) and BPEL (Business Process Execution Language), which separates security and business processes while allowing composition to be changed in real time. When it comes to web service interoperability, there are several security precautions to consider. A GE Energy study article [4] addressed these issues. Ladan et al. highlight the key security assaults that can occur in a web-based application, notably web services, as well as the existing solutions to ensure security [5]. Wu, X.et al. established a security model for service-oriented multi-application architecture, which provides a comprehensive layer of security when constructing integrated systems [6]. WSDL files are files that define the basic structure of a web application. WSDL files, which specify the web service's core structure, are vulnerable to modification, scanning, and other attacks. Mirtalebi et al. [7] presented a model for encrypting WSDL files to ensure their security. Pan Et.al [10] analyses the main security challenges with Restful web services and proposes a secure XAuth solution for Restful WCF Services.

SECURITY IN WEB SERVICES: Web services, like any distributed programme, require significant security protocols to enable secure data transport. It is commonly recognised that we do not rely on programming languages, architectures, or systems when exchanging existing information or functionality via web services. This interoperability and cross-platform accessibility of web services necessitates a larger focus on security considerations. Authentication, Authorization, Confidentiality, Integrity, Availability, and Non-Repudiation are the essential security criteria of any web-based application.

- ✓ Authentication is the process of determining a user's identity. We endeavour to ensure the user's identification and validate the identity that the user claims to be when we employ this notion.

- ✓ Authorization is the process of granting a user permission to do something. When a user is given access, authorization is generally considered as both the initial setting up of permissions by a system administrator and the validation of the permission values that have already been put up.

- ✓ Confidentiality: In the field of information security, confidentiality refers to the necessity that data in transit between two communicating parties not be accessible to a third party in order to prevent spying. A Virtual Private Network (VPN) or encryption is the general strategy.

- ✓ Integrity refers to the ability to identify tampering with information. The procedure frequently employs hashing methods, which are mathematical techniques.

- ✓ Availability: It necessitates that the resources and services be accessible to authorised individuals at all times. The Denial of Service attack is a typical attack on data availability. Its goal is to exhaust all of the service's resources, making them unavailable to legitimate users.

- ✓ Non-repudiation: The sender of a communication cannot claim that he or she did not send the message.

The following methods are used to implement the above security standards in web services:

AUTHENTICATION METHODS: In practically all applications, basic authentication is employed. A username and password are required before the user can access the application's features. Both have been verified. The fundamental flaw in the implementation is that the credentials are sent from the client to the server in an unencrypted format. Any sniffer on the network might read the sent packages. Strong authentication and authorisation tokens are provided using the Security Assertion Markup Language (SAML). SAML is an open framework designed by OASIS for sharing security information over the Internet via XML documents. The user asks the service provider to offer a service. The service provider asks the identity provider for and receives an identity assertion. The service provider can make an access control decision based on this assertion, which means it can determine whether or not to perform some service for the connected user. Before sending the identity assertion to the Service, be sure you have everything you need. The identity provider may ask the user for some information, such as a user name and password. Before transferring the password to the server, the digest authentication encrypts the password given by the user. A hash function, such as SHA or MD5, is usually utilised.

The client submits an unauthenticated request to the Web service, and the server replies with a digest authentication challenge, showing that digest authentication is supported.The client produces a nonce and sends it together with a date, digest, and username to the service. The digest is a hash of the password, nonce, and timestamp in cryptographic form. The hash is generated by the server using the password (retrieved from the service store), nonce, and timestamp (from the message), and the request is authorised if the generated hash matches the hash in the request. The benefit of digest authentication is that it is fast. Digest authentication has the advantage of being resistant to replay attacks.

AUTHORIZATION TECHNIQUES: The two technologies used to determine authorization information are XACML (eXtensible Access Control Markup) and SAML (Secure Assertion Markup Language). Given an authentication assertion and an attribute assertion, an authorization decision assertion in SAML includes deciding whether or not a principal can access a specific resource. The entities Policy Decision Point (PDP) and Policy Enforcement Point (PEP) determine and enforce authorization decisions, respectively, in accordance with a policy. SAML is a standard way to describe a security token that may be passed across numerous steps of a business process or transaction, from a browser to a portal to a network of web services, and it's a feature that OWSM supports as well. Role-based Access Control (RBAC) and Context-based Access Control (CBAC) are the two main types of authorization (CBAC). Security management is mapped to an organization's structure via RBAC. Each user can be allocated privileges based on their role in the organisation with RBAC.

TECHNIQUES FOR CONFIDENTIALITY: This element of security management is implemented via XML encryption. It's used when a SOAP message needs to be kept private while being conveyed over a multihop SOAP transaction. XML encryption is also beneficial if content in a SOAP message must be maintained encrypted after the SOAP message has been processed by a web service. The W3C recommends XML Encryption. XML can be used to express encrypted data, or elements of an XML document can be encrypted selectively. Triple-DES and AES are two popular algorithms.

INTEGRITY TECHNIQUES: The WSDL file explains the web service's capabilities. The WSDL is used to pick the web service when a service requester talks with the UDDI registry, especially during composition. If it is tampered with, the online service's integrity is compromised. XML Signature can also be used to ensure the integrity and non-repudiation of WSDL files, allowing a web service specification to be published and thereafter trusted to remain unaltered. It was created by the World Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF). Data integrity may be present at lower tiers of the OSI stack in web service communication. However, we cannot term integrity persistent if it is only implemented for SOAP communication. In a workflow situation, persistent integrity is important when a document is made up of parts written by various entities. The XML signature ensures the integrity of specific parts of the document while allowing participants to alter other parts that aren't signed. The OASIS standard Web Services (WS)-Security defines a collection of SOAP header extensions for end-to-end SOAP messaging security. By letting communication partners to exchange signed encrypted messages in a Web Services context, it ensures message integrity and confidentiality.

TECHNIQUES OF NON-REPUDIATION: To ensure non-repudiation, WS-Security employs digital signatures. The specification supports a wide range of signature formats, encryption techniques, and numerous trust domains, as well as X.509 certificates, Kerberos tickets, SAML Assertions, and custom-defined tokens. It works in the application layer and adds security features to the SOAP header. The X.509 specification is used to format the information in the digital certificate and is fully expandable. A digital certificate often contains identity information about the entity that holds the private key, as well as a serial number, expiration date, and the public key.

THREATS AND CHALLENGES: Web services may encounter risks at the message or service level. Threats to UDDI, WSDL, and XML can all be found at the service level. The following are some of the most serious risks to service levels. The service level information is available in the WSDL files and the UDDI registry in the WSDL and UDDI attacks. Any publicly available WSDL file can be tampered with by an attacker. WSDL Scanning or WSDL Tampering is two types of attacks. The first scans the WSDL file for operational details, ports, and so on. The latter tampers with the data and has access to sensitive information. Traditional means of protection, such as authentication and authorization, are ineffective against these attacks. Identity spoofing and malicious code injection: These attacks mostly target XML files. An attacker can insert malicious code into the service, causing it to malfunction. When a hacker assumes the identity of either the service requester or the service provider, this is known as an identity spoofing attack. In the first scenario, the attacker can create a well-formed XML request message and send it to the service provider, leading the service provider to believe the response is being provided to a legitimate service requester. In the second scenario, the attacker will deceive legitimate service requesters into sending messages to the phoney service provider. Such attacks are difficult to detect. Tampering with the XML Schema: An attacker can change the XML schema and make it incorrect. Such risks result in service provider failures at the end points. Session Hijacking: An attacker can take the user's session token and obtain unauthorised access to the resources available. As a result, bogus requests or responses are sent, and the session between the service requester and the provider is said to have been hijacked. Threats to web services at the message level include the following:

MESSAGE INJECTION OR MODIFICATION: Messages sent between the client and the server can be changed or new malicious messages injected. This can happen with XML files and give the hacker a lot of power.

COMMUNICATION REPLAY: An attacker captures a legal message and replays it later to gather sensitive information by gaining unauthorised access to the services. This is usually the initial step in hacking a web service, where the hacker takes control of the session and manipulates the services. Even if the same or similar payload is carried across many media such as HTTP, HTTPS, and SMTP, or across different interfaces, patterns can be detected more rapidly with the right tools.

EAVESDROPPING AND MESSAGE CONFIDENTIALITY: Interception of messages is always a hazard to web services. Traditional security techniques such as VPN and SSL are unable to protect online services from such assaults.

EXISTING SECURITY STANDARDS: XML Signature, XML Encryption, and SAML have all been addressed as part of the basic implementation of security features in online services. This section covers the rest of the key standards.

WS-SECURITY: The protocol is now officially known as WSS and is being developed in Oasis-Open by a committee. The standard specifies how security tokens should be included in SOAP communications, as well as the XML security specifications for encrypting and signing the tokens. The communications have tokens linked to them, as well as timestamps. The methods for encrypting other elements of the SOAP message are specified in the standard. The standard's key features are WS-Policy, WS-Trust, and WS-Privacy. Organizations can use the WS-Policy to set the security criteria for their web services. The policy information is attached to the web services using WSDL binding. Ws-Trust is a concept that describes how trust connections are formed. Trust can be earned directly or through intermediaries. A trust is broken red in the situation of brokered trust. A trust proxy is used in the event of brokered trust to read the policy information and request the necessary security token for inclusion in the SOAP message. The WS-Trust is used to evaluate the privacy claims enclosed within the SOAP messages against the user and organization's preferences, and the WS-Privacy defines how privacy requirements can be included in the policy. Client security and OAuth are the two most important security measures in a RESTful web service. Basic authentication and authorisation functions can be provided through client security. Basin HTTP Authentication and HTTP Digest Authentication are the most common options. OAuth is a protocol that defines a safe authentication model that allows one user to authenticate on behalf of another. In general, OAuth is commonly utilised in popular social Web sites to provide third-party consumers access to a user account and associated resources (application). The user data is usually accessed via RESTful Web Services by the consumer. Message signatures are calculated using specified signature procedures in the OAuth1 protocol. Signatures are rather complicated, thus they're implemented in their own module. Jersey has packages for both OAuth1 and OAuth2. In contrast to OAuth 1, OAuth 2 is a framework rather than a formal protocol. Many extension points are defined in the OAuth 2 standard, and it is up to service providers to implement these details and describe them for service users. Furthermore, OAuth 2 defines many authorization flows. For Restful WCF Services, XAuth and an authentication manager have been implemented. To combat Denial of Service (DoS) assaults, server-side streaming solutions are currently deployed. To address security problems, steps can be made to ensure that the security framework is properly installed. The deployment analysis is divided into three sections. Shell for in-transit, vendor-controlled, and deployment. The security for the SOAP messages that are exchanged is provided by the In-Transit security. For secure communication, we can use Secure Socket Layer

(SSL), and for services, we can utilise WS-Security. Effective IP security and access restrictions should be given. We must ensure that the deployed services are secure enough to prevent information leaking for deployment shell security. Errors and exceptions should be handled carefully so that the attacker does not exploit loopholes in the system, making it more vulnerable. Auditing and logging assist in identifying and resolving issues. Security breaches can be identified and prevented with the use of auditing and logging. To protect the service at the code level, there are a few principles to follow.

The following are the details:-

- Never put your total reliance in a user's input.
- Do not repeat the information provided by the user.
- Validation should not be done using client-side scripting languages. [23]

We can follow the additional guidelines within the web services. Incorporating digital signatures into SOAP communications is possible. Firewalls based on XML can be set up. With header information, all requests and answers should be encrypted. Cross-Site Scripting can be avoided by using validations. To prevent input parameter tampering, exceptions should be handled properly, and passwords and other critical fields should be encrypted. HTTP queries are vulnerable to assaults, thus precautions should be taken.

CURRENT CHALLENGES AND THREATS: Security based on the end user, preserving security when routing between numerous web services, and abstracting security from the underlying network are all ongoing concerns. The online service may not have access to the end user's information or details. They will be accessible through the website or any client that accesses the web service on the user's behalf. This problem can be solved by including the information directly in the SOAP message. This could save the user from having to re-authenticate each time a SOAP request is sent. This category includes single sign-on and federated trust. The traversal of SOAP messages over numerous hops before reaching the end point is referred to as web service routing. Maintaining the secrecy of data from SOAP intermediaries is required when routing between online services. Traditional security mechanisms (such as SSL) operate at the transport layer, whereas SOAP operates at the application layer. As a result, improved mechanisms must be developed to assure security in SOAP gaps generated by intermediates. Acunetix is software that has recently gained popularity in order to meet the obstacles. Acunetix Web Vulnerability Scanner is a feature-rich solution for safeguarding web applications and finding vulnerabilities. With the Web Services Security scanning tool, you'll be able to do an automated vulnerability evaluation against a Web Service using a more accurate and updated version of the same scanning engine that previously examined web applications. Another new feature is the Web Services Security Editor, which enhances the Web Services scanner's functionality by allowing for more in-depth examination of XML answers, WSDL structure, WSDL XML analysis, syntax highlighting for all coding languages, and regular expression searches. Databases in SOA setups are also vulnerable to threats. When a database application is exposed as a service or is merely accessed by a Web service, the database's security is totally dependent on the security of the apps that contact it directly. Confirming the requester's authority to access the database should be accounted for by the exposing application. In addition, before sending requests to the database, the programme must filter them. SQL injection attacks or other efforts to undermine the database or retrieve sensitive information may be included in incoming requests. To avoid unintended leaking, the application must filter the data before sending it to the requester.

CONCLUSION: The security of the resources being given is a crucial concern when launching a web service. The distributed and autonomous nature of web services, as explained in the paper, has left them exposed to a wide range of threats and attacks. Existing security procedures will not be sufficient to meet the security needs. New attacks on WSDL and XML files, which specify the core structure of web services, are on the rise. Maintaining security regulations and checks against violations and message manipulation is a potential concern due to the interoperable nature of web services. As a result, it's critical to integrate security procedures from the start and to allow cooperating services to adopt a similar strategy. We can only hope that new solutions and technology will bridge the gap between services and take security feature operations to the next level.

REFERENCES:

[1]. Balasubramanian, N., & Ruba, A. (2012, August). Security: a major threat for web services. In *Advanced CommunicationControl 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.

[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 REST ful 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.

[25]. Aruna. S (2016), Security in Web Services- Issues and Challenges (IJERT)