International Journal of Computer Theory and Engineering, Vol.2, No.6, December, 2010
1793-8201

Realization of Threats and Countermeasure in Semantic Web Services

Mr. Suresh Kumar *, Mr. Rakesh Kumar Prajapati *, Dr. Manjeet Singh ** and Asok De*

***Abstract*—Semantic Web services (SWS) are worn by large number of companies as they rendering products and services to customers and business partners throughout the Internet and corporate extranets. The security necessities for these service providers are of supreme importance. Platform-based security services are provided by operating system. Internet Information Services can be used to offer point-to-point business solutions. However message based architecture of SWS is increasingly being used, so new challenges automatically inherited from previous one. All these scenarios require security to be deal with at the message level, transmission level and mutual authentication to carry out cross-platform interoperability and routing through multiple middle level agent nodes. Semantic Web service environment (SWSE) need to implement different level security solutions for including authentication, identification, privacy and data integrity. The threats of network eavesdropping or information disclosure at intermediate application nodes must be addressed if our SWS requests or responses messages convey sensitive application data as credit card numbers, SSN number, employee details, and so on. We will discuss various categories of vulnerabilities and find out causes of threats. Countermeasures will be part of our discussion like Symmetric encryption using shared keys, asymmetric encryption using X.509 certificates, and Digital signature.**

***Index Terms*—SWS, threats, security, SOAP, XML, X-KRSS, digital certificates, encryption**

I. Introduction

User passes sensitive application data in SWS by requests or response messages then users can not be ensure that they remain private and unaltered original data while in transit. SWS provides integrity checking through digital signatures, and it also supports XML encryption to encrypt-sensitive elements of the entire message. The advantage of SWSE is that it is based on the emerging SWS Security standard developed by W3C. That provides a solution for messages and user privacy during passing through multiple intermediate nodes [14]. Figure 1 shows, end user sends all request and response in form of HTTP request to web portal. Web portal responds in similar format to user. Web portal and service provider communicate with each other in SOAP messaging format. SOAP has two main parts envelop element and body element [9]. In the Semantic Web Service, XML based SOAP messages are used when the clients requests to Semantic Web Server or when Semantic web server sends response (web service) messages to the clients [4]. End user communicates to client's SWS by HTTP protocol.

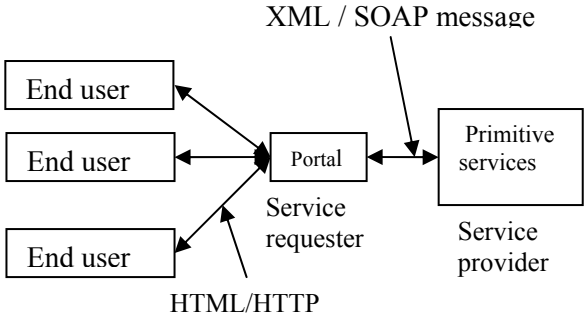

Figure 1: Example of Web Service

Uses of transport level encryption through SSL or IPSec channels are only appropriate where you are in control of both endpoints. When two parties agree to use SWS-Security, the precise format of the authentication token must also be agreed [15]. We can use declarative principal permission demands to control access to individual Web methods based on the identity, role or membership and group management of the caller. You can use SSL or IPSec to provide only transport layer encryption. In other environments and where messages are routed through intermediate application nodes, there is a need of message level solution [13]. The SWS-Security standard defines a confidentiality service based XML Encryption standard on the World Wide Web Consortium. We will discuss encryption that may be full message or partial a SOAP message before it is being transmitted [13]. To overcome from these highly defeating threats in distributed environment asymmetric encryption by using X.509 certificates, digital signature and Symmetric encryption by using shared secret keys will be used and it will be part of our discussion [14].

II. Review Stage

SWS use to process sensitive or restrictive information in distributed environment. It needs to authenticate clients. It also needs to know who are publishing SWS in web directory, it need a semantic searching techniques to get accessibility for every one from everywhere [3]. Authentication system should be able to confirm about requestor's Id's first, As in Windows environments user can be authenticated first if user request for Windows services, but you can not use Windows

sureshpoonia@yahoo.com, rkp.its@gmail.com, mstomar2000@yhaoo.com, asok.de@mail.com Ambedkar Institute of Technology, New Delhi (India) *,YMCA University of Science & Technology Faridabad (India) **.

authentication where are you not in control of both endpoints. W3C provides authentication solutions that are emerging WS-Security standard like XKMS, XKRRS. W3C provides a standard framework to secure system by using SOAP headers to pass authentication details in form of user names, passwords, Kerberos tickets, X.509 certificates and Digital [8].

III. Main Semantic Web Service threats

To build secure SWS we should know the associated threats. The pinnacle of threats directed at SWS are rooted here if we want to beat them some basic security principle should be enforced with secure web service at their design time [7]. A system can not be foolproof and secure from attacks but if you try to find out all known Vulnerabilities to at least develop required countermeasures for those attacks, which are very common. Message level attacks are easy to disrupt it, destroy it or drop all messages passing through intermediary devices [10]. Any malicious node can perform all above attack to irritate legitimate user. To overcome from this message format should be pre defined as what message contents are organized in body element, what content should be organized in header elements. SOAP envelop is outline structure of massage block that contains all related information about block size, sender & recipient address , protocol version using to encapsulate data.

SOAP body contains data about certificate, digital signature, certificates expiry date, use of certificates and issuer's name / Ids etc. Structure of SOAP message is depicted in figure 2. Data block header contains data size, format of data, and representation coding, and its version number [10].

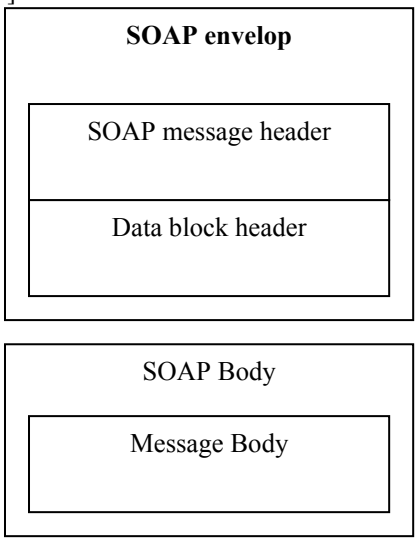

Figure 2: SOAP Message Structure

All threats and vulnerabilities should be realized before developing its countermeasures. First we discuss various types of threats/vulnerabilities in this section after that we will proceed to find its countermeasures consequently. Various types of threats are given:-

A) Unauthorized access
B) Parameter manipulation
C) Network eavesdropping
D) Disclosure of configuration data
E) Message replay

Figure 3 is showing attack scenario, it explains attacking stages of consumer Semantic Web Service. Consumer may be a system / machine/ or software agents. Consumer comes under attack when attackers compromise its working agents. Some time smart attacker access data by randomly generated passwords, dictionary attack and guessing passwords also lead to unauthorized access. Network eavesdropping is different type of cryptanalysis techniques like channel analysis, CPU overloading, and memory consumed etc. These all are source that attackers get about transmitted data. Attacker manipulates input parameter to get access sensitive data [10]. Massage replay is also a different type of attack that capture data during transit and after some time resent same message to show that it was from original sender. These three types of attacks occur during transmission and other type of attack occurs when dynamic WSDL describe its configurations [1]. Dynamic scripts are not considered to be very secure because some of them can be easily bypassed. F/W is firewall that provide data filtering for all incoming and outgoing communication.

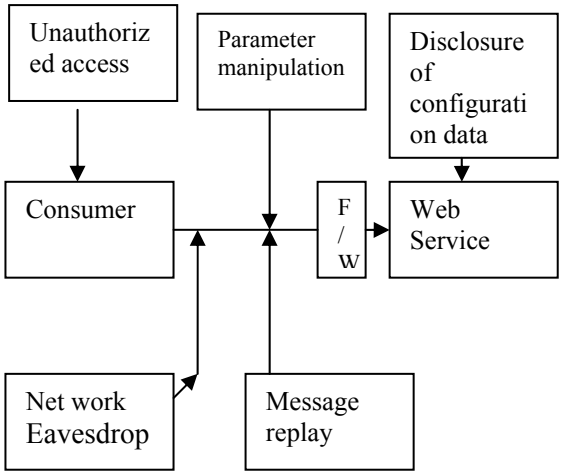

Figure 3: Attack scenario

A. A. Unauthorized Access

Semantic Web Service provides sensitive or restricted information authentication and authorize their callers. Weak authentication and authorization can be exploited to gain unauthorized access to sensitive information and operations [3].

A.1 Vulnerabilities

Vulnerabilities are all possible breach of security by means attackers can lead to unauthorized access of intended data through a Web service. Vulnerability may be a threat or may not. If vulnerability breached any how it leads to threat and threat leads to attack. It is not very tough to restrict vulnerability or threats but really unknown vulnerability makes quite tough. Some known vulnerabilities are discussed here.

1) No authentication used.
2) Passwords passed in plaintext in SOAP headers.
3) Basic authentication used over an unencrypted communication channel.
4) Smart hackers
5) Dictionary attack

6) Acceptance of default password

A.2 Countermeasures

We can use the following countermeasures to prevent unauthorized access

1) Use password digests in SOAP headers for authentication.
2) Use Kerberos tickets in SOAP headers for authentication.
3) Use X.509 certificates in SOAP headers for authentication.
4) Use Windows authentication.
5) Use role-based authorization to restrict access to Semantic Web Service. This can be done by using URL authorization to control access to the Web service file (.asmx) or at the Web method level by using principal-permission demands.

B. Parameter Manipulation

Parameter manipulation refers to the unauthorized modification of data sent between the Web service consumer and the Web service publishers. For example, an attacker can intercept a Web service message as it passes through an intermediate node in route to its destination, and can then modify it before sending it on to its intended endpoint.

B.1 Vulnerabilities

Vulnerabilities that can make parameter manipulation possible include messages that are not digitally signed to provide tamper proofing and messages that are not encrypted to provide privacy and tamper proofing.

B.2 Countermeasures

We can use the following countermeasures to prevent parameter manipulation.

1) Digitally sign the message, the digital signature is used at the recipient end to verify that the message has not been tampered with while it was in transit.
2) Encrypt the message payload to provide privacy and data integrity.

C. Network Eavesdropping

With network eavesdropping, an attacker is able to view Web service messages as they flow across the network. Let see, an attacker can use network monitoring software to retrieve sensitive data contained in a SOAP message. This might include sensitive application level data or credential information. Like credit card number, SSN number etc.

C.1 Vulnerabilities

Vulnerabilities that can enable successful network eavesdropping include:

1) Credentials information passed in plaintext in SOAP headers
2) No message level encryption used
3) No transport level encryption used
4) Key generated smaller size

C.2 Countermeasures

We can use the following countermeasures to protect sensitive SOAP messages as they flow across the network.

1) Use transport level encryption such as SSL or IPSec. This is applicable only if you have control to both endpoints.
2) Encrypt the message payload to provide privacy. This approach works in scenarios where we message travels through intermediary nodes route to the final destination.

D. Disclosure of Configuration Data

There are two main ways in which a Web service can disclose configuration data. First, the Web service may support the dynamic generation of Web Service Description Language (WSDL) or it may provide WSDL information in downloadable files that are available on the Web server [1].

WSDL describes the characteristics of a Web service, for example, its method signatures and supported protocols. Second, with inadequate exception handling the Web service may disclose sensitive internal implementation details useful to an attacker.

D.1 Vulnerabilities

Vulnerabilities that can lead to the disclosure of configuration data include.

1) Unrestricted WSDL files available for download from the Web server.
2) A restricted Web service supports the dynamic generation of WSDL and allows unauthorized consumers to obtain Web service characteristics.
3) Weak exception handling.

D.2 Countermeasures

1) We can use the following countermeasures to prevent the unwanted disclosure of configuration data.
2) Authorize access to WSDL files using NTFS permissions.
3) Remove WSDL files from Web server.
4) Disable the documentation protocols to prevent the dynamic generation of WSDL.
5) Capture exceptions and throw a "Soap Exception" or "Soap Header Exception" that returns only minimal and harmless information back to the client or user.

E. Message Replay

Web service messages can potentially travel through multiple intermediate servers. With a message replay attack, an attacker captures and copies a message and replays it to the Web service impersonating the client. The message may or may not be modified [10].

The most common types of message replay attacks are following:-

E.1 BASIC replay attack.

The attacker captures and copies a message, and then replays the same message and impersonates the client. This replay attack does not require the malicious user to know the contents of the message.

E.2 Man in the middle attack.

The attacker captures the message and then changes some of its contents, for example, a delivery address, and then replays it to the Web service.

E.3 Vulnerabilities

Vulnerabilities that can enable message replay include.

1) Messages are not encrypted.
2) Messages are not digitally signed to prevent tampering.
3) Duplicate messages are not detected because no unique message ID or digest is used.

E.4 Countermeasures

We can use the following countermeasures to address the

threat of message replay:

1) Use an encrypted communication channel like SSL/TLS security.
2) Encrypt the message payload to prevent man in the middle attacks where the message contents are modified before being replayed.
3) Use a unique message ID or nonce with each request to detect duplicates, and digitally sign the message to provide tamper proofing. Nonce is a cryptographically unique value used for each request and response.

IV. Web Service Enhancement & Design Considerations

Before start to develop Semantic Web Service, there are a number of issues to be considered at design time [7], [8]. We will discuss the key security considerations as given below:-

1) Authentication Requirements
2) Privacy and integrity of message.
3) Resource and access identities.
4) Code access security
5) SWS input output validation

A. Authentication Requirements

Web service provides sensitive or restrictive information. It needs to authenticate user to support Semantic Web Service security. Users must have a digital signature to show its possession same as X.509 certificates are used o check authenticity. X-KRSS (XML based Key registration request specifications) Key management server registers users for new certification. Figure 4 shows technique by which client registration are done means each public key binding with a private key. There was various techniques evolved to secure communication but X-KRSS is getting higher performance and feedback rather than digital signature, security tokens, or trusted third party system. Working of X-KRSS very simple - User application sends request to be register itself to XKMS server creates a request to check users' credential status. XKMS distribute and manage digitally signed certificate by trusted third party [8], [9].

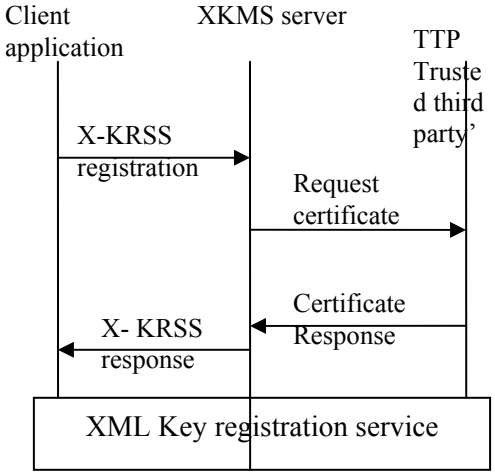

Figure 4: X-KRSS registration

Security standard of W3C provides a standard technique for using SOAP headers as authentication details in the form of user names, passwords, Kerberos tickets, or digital certificates [10].

B. Privacy and Integrity Requirements

Let see, we passed sensitive application data in web service request or response message, consider how we can ensure that they are remain private and unaltered while in transit. WSE provides integrity checking through digital signatures, and it also supports XML encryption [13] to encrypt sensitive elements of the entire message payload. The alternative is to use transport level encryption through SSL or IPSec channels. These solutions are only appropriate where we are in control of both endpoints.

C. Resource Access Identities

ASP.NET Semantic Web Services do not impersonate, and the least privileged ASP.NET process account is used for local and remote resource access. You can use this ASP.NET process account to access remote network resources such as SQL Servers that require Windows authentication, by creating a local account on the database server. On Windows Server 2003, the Network Service account is used by default to run Semantic Web Services [7].

D. Code Access Security

Consider the trust level defined by security policy in your deploying environment by Semantic Web technologies like XML, RDF. Your Semantic Web Service's trust level, defined by its "trust" element configuration, affects the types of resources that it can access and the other privileged operations it can perform [7].

E. SWS Input / Output Validation

SWS accept some input data to process request. SWS accept data as input to process and get some out put. What data are getting input must validated when passed validation to enforce their business rules, organization policy and to prevent potential security issues. Web methods attribute of Web service entry points. Web methods can accept strongly typed input parameters or loosely typed parameters that are often passed as string data. This is usually determined by the range and type of consumers for which the Web service is described. If we expose our Web service to a large number of Internet consumers and require secure operation. It require client to server, server to client and server to server level authentication. But when they are running in distributed systems, it stays not easy. Since the consumers will not have proper domain accounts in that environments to map their credential authentication can be achieved [3]. It becomes sticky large numbers of users to handle. There is needed a TTP to manage the certificates and binds how those certificates will be delivered to clients [3], [7].

V. XML Encryption / Decryption

XML is a technology for managing data exchange. The main objective of the XML Semantic Web Services is to provide interoperability to applications distributed on networks including the Internet. Simple Object Access Protocol (SOAP) is built on XML and HTTP protocols so that distributed software applications can communicate [13].

All type of data can be encrypted and decrypted easily by standard cryptographic techniques like RSA, DES, AES and Hash of a message calculated for integrity checker. XML is simple description about full certificate's contents [1], [11]. XML encryption and decryption can be done in following two manners:-

A. ASYMMETRIC Encryption USING X.509 CERTIFICATES
B. SYMMETRIC Encryption USING X.509 CERTIFICATES

A. Asymmetric Encryption Using X.509 Certificates

Sender SWS1 wants to send encrypted data to SWS2 that is running on receiver site. SWS1 uses the public key of SWS2. SWS2 decrypt message by owned private key. It is simple and easy but when number of user increased surprisingly this method not helpful. Why? Because required pair of encryption and decryption keys are very high. So to distribute digital key pairs need TTP (trusted third party) that can distribute and manage those pairs [12]. X.509 certificate is also a good solution of this problem, when you want to incorporate various other features not only encryption/ decryption. This certificate binds a private key with a public key. Public key are distributed by publicly but private key is known by only intended user. A simple configuration is described here to understand basic format of certificate used to maintain trust and privacy throughout communication between SWS [12].

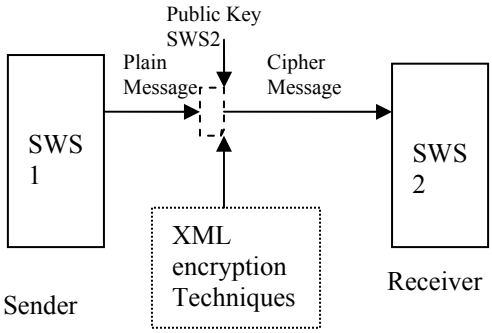

Figure: 5 XML Encryption

Simple format of Certificate X.509:-

```
<configuration>
<microsoft.web.services>
 <security>
 <x509 storeLocation="CurrentUser" />
< user name = " uname" / uid=" user">
< pwd="user ">
< public key = " q5mm….">
< validity >
< from date = "12/06/2009">
< expire date=" 28/04/2011">
</ validity>
< purpose = " money transfer">
< / x509 >
 </security>
 </microsoft.web.services>
</configuration>
```

B. Symmetric Encryption Using Shared Secret Keys

In this encryption technique large data can be encrypted and decrypted very fast manner. If all transiting data will be encrypted by asymmetric methods then it will take large time than usual. So a data less amount but higher security requirement will be encrypted by asymmetric method. Data can be like user Id, password, login phrase, and security tokens etc [6], [9]. while less secure data encrypted by symmetric key. To get same level of security asymmetric encryption requires double key size of symmetric encryption. Sender and receiver SWS encrypt and decrypt data with same key that is shared between communicating parties.

Security is really big deal in distributed environment but every body want security with moderate performance. Security services are creating burden on performance of system. It consumes processors, memory and time. There are some mobile applications where these all three thing matter more to be living for system. Performance and quality of services are needed to address with security enforcement strategy.

VI. CONCLUSION

Semantic Web Service (SWS) is the emerging standard in web based application. The SWS define new trends to develop secure communication between machines to machine. Machine can do bit of better but not as human. So SWS made machine to exercise on behalf of running services. Machine to machine level communication need similar techniques to identify, and verify a machine before sharing any information on network. Data transmit from different layer like application layer to transport layer. There are specified protocols to provide security in different layer. XML encryption and decryption was part of application layer protocol where data representation can be changed before going into next layer. To provide application layer security SOAP is used to specify architecture of message. Message contains three parts- envelop addressee, header details about sender and receiver, body of message.

SWS authenticate to each other by passing security tokens in a standard SOAP message [9]. Tokens can include user name and password credentials, Kerberos tickets, and X.509 certificates etc. SWS-Security addresses privacy and integrity issues by being dynamic in nature of SWS. We can encrypt whole or partial messages to provide privacy [6].

Using Digital certificates from a security point of view often it becomes easy for TTP (trusted third party) to control large numbers of producers and consumer simultaneously. We must carefully manage the certificates and consider how they should be delivered to clients. Certificate renewal and revocation can be done secure infrastructure. Other potential issues in Internet situation are the scalability of services, processing in the exhaust, encryption/decryption, certificate registration, validation, or fresh certification request. TTP is required to generate trust between Semantic Web Components. In distributed environment security leads to system's bad performance and lower quality of services. Without losing performance and quality of services maximum security foundation is an open research area in SWS domain.

ACKNOWLEDGEMENT

We want to extend thanks to our Principal "Prof. Asok De" and Registrar and Chairman Department of Computer Science and Engg. Prof. A. K. Sharma of YMCA University of Science and Technology for providing us all infrastructure and facilities to work in need and to do this research work.

REFERENCE

[1] Bhavani Thuraisingham, "Security standards for the semantic web", ELSEVIER, Computer Standards & Interfaces, Vol. 27, 2005, pp.257–268.
[2] Bhavani Thuraisingham , Confidentiality, Privacy and trust Policy Enforcement for the Semantic Web", 8th IEEE International Workshop on- Policy for Distributred System and Network, 2007.
[3] Dhafer Thabe," Toward Situational Secure Web Services Design Methods", IEEE International Conference on Web Services, 2007.
[4] David Geer, Taking Steps to Secure Web Services, IEEE Technology News, Oct 2003.
[5] Bhavani Thuraisingham and Pranav Parikh ,"Trustworthy Semantic Web Technologies for Secure Knowledge Management ",2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, IEEE/IFIP-2008, pp186-193.
[6] Genong Yu, Liping Di, "Secure Service Composition in Sensor Web", IGARR, IEEE, 2009.
[7] J. D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan Microsoft Corporation ," Chapter 12: Building Secure Web Services ", http://msdn.microsoft.com/en-us/netframework/aa663324.aspx, Published: June 2003.
[8] Liu Zhuang, Huang YuanFei , Qin RuLie , "A Web Service Secure Model" ,Proceedings of the First International Conference on Semantics, Knowledge, and Grid (SKG 2005), IEEE , 2006.
[9] Llanos Tobarra, "Analysis of Web Services Secure Conversation with Formal Methods", Second International Conference, Internet and Web Applictaion and service, ICWS-2007.
[10] Mohammad Ashiqur Rahaman, "SOAP-based Secure Conversation and Collaboration, IEEE international Conference on Web Services, 2007.
[11] Nils Agne Nordbotten, XML and Web Services Security Standards, IEEE Communication survey and Tutorials, Vol.11, no.3, 2009, pp. 4-18.
[12] Somchart Fugkeaw et al, "Adding SAML to Two-Factor Authentication and Single Sign-On Model for Dynamic Access Control", ICICS, IEEE, 2007.
[13] Xiaohong Li ,Ke He ,A Unified Threat Model for Assessing Threat in Web Applications , International Conference of Information Security and Assurance, IEEE 2008 , pp 142-146.
[14] Ying Liu , " Securing XML Web Services with Elliptic Curve Cryptography", IEEE, pp 974- 976.
[15] Youxiang Duan, Yongtang Bao ,"A Secure Web Services Model Based on the Combination of SOAP Registration Info and Token Proxy ", ISCSCT , IEEE , 2008, Vol.99 pp 15 -21.