Nilambari Joshi, Paras Patel, Dr. B.B. Meshram / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 3, May-Jun 2013, pp.1211-1217

Survey of Security in Service Oriented Architecture

Nilambari Joshi*, Paras Patel, Dr. B.B. Meshram*****

*(Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Matunga, Mumbai - 400019)

**(Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Matunga, Mumbai - 400019)

*** (Head of Department, Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Matunga, Mumbai – 400019)

ABSTRACT

Service Oriented Architecture (SOA) is a driving force behind all present and evolving techniques of data exchange and resource sharing over the network. This helps in adapting to the changing market needs efficiently and effectively. Inherent characteristics of SOA framework have nurtured agility and flexibility in distributed computing environment but also have posed high security challenges. This is especially because of the anonymity of the end user of a service and data exchange over unsecured network. This paper discusses risks posed by SOA related to important aspects of security Authentication, Authorization, Confidentiality, Data Integrity and Non-repudiation. It also presents mechanisms which are being used by service providers to deal with these security concerns.

Keywords – *Kerberos, Public Key Cryptography, Public Key Infrastructure (PKI), Security, Service Oriented Architecture (SOA),*

I. INTRODUCTION

1.1 Service Oriented Architecture

Service-oriented architecture (SOA) is now a days well established framework that addresses the requirements of distributed computing by loosely coupled, standards-based, and protocol independent communication among involved software resources. In SOA, software resources are packaged as "services", which are well defined, self-contained modules that provide standard business functionality and are independent of the state or context of other services. Services are described in a standard definition language, have a published interface, and communicate with each other requesting execution of their operations in order to collectively support a common business task or process [2].

Service Oriented Architecture is a methodology for achieving application interoperability and reuse of IT assets that feature a strong architectural focus on ideal level of abstraction, a deployment infrastructure and reusable library of services (W3C definition) [9]. It also incorporates support for organizing and utilizing resources that are under control of different administrations.

It is need of time for enterprises to quickly respond to business changes with efficiency and leverage existing investments in applications and application infrastructure to address newer business requirements. The solution proposed and actively being used to cater these requirements is Service Oriented Architecture, which allows enterprises to plug in new services or upgrade existing services in a granular fashion to address the new business requirements. It provides the option to make the services consumable across different channels, and exposes the existing enterprise and legacy applications as services, which is basic building block of Service Oriented Architecture.

1.2 SOA Characteristics

Service Oriented Architecture emphasize on reusability of existing resources by means of following characteristics

1. Discoverable - A service consumer that needs a service discovers what service to use based on a set of criteria at runtime. The service consumer asks a registry for a service that fulfils its need.
2. Loosely coupled – SOA binding minimizes dependencies between services and thus achieves loose coupling through discovery and contract.
3. Autonomous – The service controls the business logic they encapsulate. The service only exposes interface of underlying functionality and can change implementation without any change required at consumer's side.
4. Stateless - Statelessness refers to services that do not keep track of transaction or session information.
5. Composable - Service composition is assembling service capabilities that consist of smaller units of logic to solve larger problems. It facilitates the assembly of composite services.
6. Interoperable - The ability of systems using different platforms and languages to communicate with each other. Each service

provides an interface that can be invoked from a client which can run on any operating system and can be implemented in any language. The only requirement is it should abide to the data format and protocol as suggested in the service interface.

II. SOA SECURITY CHALLENGES

Characteristics and design principles of SOA make systems more susceptible to security threats because of the following reasons.

1. Service interface is exposed publicly to whole world. The owner has least control over who can consume the service.
2. Data is exposed to wide range of users. Data protection during transit and in storage is important to ensure data integrity and privacy.
3. Data travels through heterogeneous environments, having different policies, technologies, network protocols etc. therefore it is difficult to integrate and synergize different security measures deployed.
4. Connectivity in SOA is not point to point. It is hop by hop. This limits use of SSL for data protection while on network.
5. This system is still vulnerable to a replay attack which simply replays a valid signed message, and gains unauthorized access.

2.1 Framework Induced Security Concerns

Following generic security concerns are applicable to SOA, but with a bigger impact due to SOA characteristics.

1. Authentication - Since services are exposed publicly, it is difficult to know beforehand who the users of the service are. The services invoked might be across different organizational domain. It is required to have common trusted authentication mechanism across services to ensure identity of the ultimate user using them.
2. Authorization - It is important to verify capability and rights of user to take an action or get some information. Traditional approach of role based authorization might not be sufficient in SOA, since same user can invoke the service in different context with different capabilities. Also there should be way to communicate user capability information across services, which are integrated as a part composite service.
3. Confidentiality - Since data is shared across different services and across different domains, there are high chances of data being exposed to unintended recipients unless strict measures to protect data are imposed.
4. Integrity - There is high possibility of data being tampered during transit over the network. There might be different mechanisms of data protection deployed for different services, which are part of same transaction. There should be mechanism to communicate and agree upon security measures to be incorporated across services to ensure data integrity.
5. Non-repudiation – Whenever data is shared across services, there should be way to ensure that it is from authenticated source.

2.2 Technology related Security Concerns

SOA security is also affected by certain technological aspects .XML being the most widely used mechanism for service invocation and message transfer, XML related security issues need to be handled. Following security concerns are frequently observed in SOA paradigm.

1. SQL Injection - SQL Injection attacks involve the insertion of SQL fragments into XML data to return inappropriate data, or to produce an error which reveals database access information.
2. XML External Entry – Document Type Definition (DTD) functionality that is available in XML is used to define syntax of document elements. It also allows outside data to be embedded into an XML document. By specifying a local file, some XML engines could be made to access unauthorized information from the local file system.
3. XML Denial of Service – This attack takes advantage of, the ability to pull in entities which are defined in a DTD. Pulling the entities recursively causes memory to exhaust and thus deny service to further requests.
4. Capture Relay Attacks – A service in SOA is protected by a policy which ensures that service requests are digitally signed. This system is still vulnerable to a replay attack which simply replays a valid signed message, thus gaining unauthorized access.

III. SOA SECURITY MECHANISMS

Security measures are designed and applied to address different security aspects as mentioned in section 2.1. Most widely used and comprehensive mechanisms to deal with SOA security can be considered as Public Key Infrastructure (PKI) and Kerberos.

3.1 Public Key Infrastructure (PKI) provides the framework of services, technology, protocols, and standards that enable you to deploy and manage a strong and scalable information security system [6]. It is based on public key cryptography for encryption. The PKI creates digital certificates which map public keys to entities, securely stores these certificates in a central repository, and revokes them if needed.

The most popular uses X.509 identity certificates. In this PKI, a highly trusted CA issues X.509-based certificates where a unique identity

name and the public key of an entity are bound through the digital signature of that CA [1].

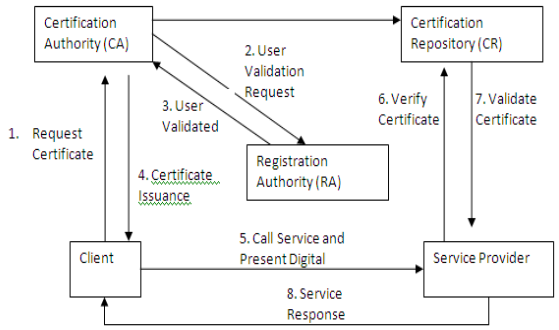

Fig.1 PKI Workflow

3.1.1 PKI Workflow [1] [6] [9]

As shown in Fig.1 the main steps involved in PKI workflow are as below -

Step 1 – The user provides credentials to the Certification Authority (CA) and Requests for the certificate.

Step 2 – CA contacts Registration Authority (RA) to validate the user credentials. If user is authentic, RA gives go ahead to CA to issue certificate.

Step 3 – CA issues certificate which includes user's public key and certificate expiration date.

Step 4 – User Presents the Certificate to service provider while requesting a service

Step 5 – Service Provider verifies the certificate and if certificate is valid, the communication starts

3.1.2 Security Considerations addressed by PKI.

PKI addresses most of the security challenges posed by SOA framework.

1. Authentication – Certification Authority (CA) acts as a trusted third party to ensure authenticity of service requester. Both the entities involved in data exchange trust CA as intermediary. Thus without exchanging credentials directly with the service provider, service requester can be authenticated.
2. Confidentiality – Each entity involved in Data exchange maintains a pair of Public and private key pair. Whenever data is sent over the network, it is encrypted using public key of the receiver. At the receiver end, it is decrypted using receiver's private key. Any unintended person cannot decrypt the message without knowing actual receiver's private key.
3. Integrity – Data integrity can be achieved by two ways. Either by using Digital signature or by using Message Authentication Code (MAC). Any data tampering results in non-verification of the digital signature.
4. Non-Repudiation – The entity's signing private key is used to bind the entity to a particular piece of data this can be used as non-repudiable evidence to prove to a third party that this entity did originate this data.

3.1.3 PKI Limitations –

1. PKI doesn't deal with Authorization of user to perform a particular action or invoke a service.
2. PKI is a resource consuming technique in terms of CPU and Memory so it cannot be easily integrated with low power web enabled devices mobile phones.
3. It can be tedious process to obtain X.509 certificates from a trusted CA, especially if a local RA does not exist.
4. Each site involved trusts its users, CAs, and other sites. If the trust between any of these is broken, then the impact can potentially be severe.

3.2 Kerberos - Kerberos is an authentication protocol which works on the basis of "tickets" or session keys to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner. Kerberos uses symmetric key cryptography. Cross-realm authentication is a useful and component of Kerberos aimed at enabling secure access to services across organizational boundaries.

3.2.1 Kerberos Protocol Workflow [1] [4] [5] –

As shown in Fig.2 the main steps involved in Kerberos workflow are as below

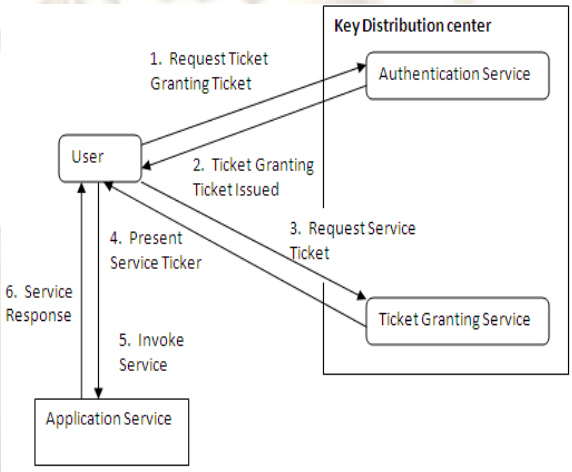

Fig.2 Kerberos Workflow

1. User sends credentials to Authentication Service, and requests for Ticket Granting Ticket (Session key to be used with ticket granting service encrypted with TGS secrete key).
2. AS verifies user with reference to the data maintained at its end. After verification, it generates encryption key, and a timestamp (same as that in the user session) and expiration time usually 8 hrs. AS sends session key to be used between user and TGS encrypted with user's secrete key and TGT encrypted with TGS secrete key and sends it to the user.

{{Ttgs,Ksession}Ktgs,Ksession}Kuser

3. User sends this TGT and service ID to Ticket Granting Service.

). {TGT,{request,client-IP,time}Ksession} (where TGT = {Ttgs,Ksession}Ktgs)

4. TGS decrypts TGT and recovers session key to be used with the client for future communication.

5. TGS generates session key (Client-Service Key) to be used by client for further communication with the service. It encrypts it with Service' secrete key and sends back to the client. {{Tservice,Kservice-session}Kservice,Kservice-session}Ksession
6. Client sends the encrypted Service ticket (obtained from TGS) to the service and requests an operation.

({Tservice,Kservice-session}Kservice

7. Service decrypts the ticket and obtains session key. This key is further used for communication with the client till its timestamp expires

3.2.3 Security Considerations addressed by Kerberos

1. Authentication – In Kerberos client authentication is done initially by authentication service. During first communication of client with TGS and application service, both these entities decrypt the ticket (TGT and service ticket respectively) with their secrete key with AS. Successful decryption ensure authentication of the client.
2. Confidentiality – Data can be exchanged confidentially by using secrete key encryption. The secrete keys are generated for every client –server session and are time bound.
3. Integrity – Kerberos enforce session based keys for client-server communication which are time bound. So if any user intercepts the message while in transit, and try to decrypt it to get session key, it is almost impossible to use that key and send a forged message to the server within session key expiration duration.

3.2.4 Kerberos Limitations –

1. Existing services need modifications to handle Kerberos protocol.
2. Key Distribution center can be a single point of failure. If it is affected, the entire Kerberos system is at risk.
3. Kerberos cannot be used when interacting with a non-kerberosed system.

3.3 Authorization Mechanisms

Service Oriented architecture facilitates services being shared across different administrative domains. Services sharing necessitates authorization mechanisms which determine who is authorized to access these resources and in which ways, and who is not authorized. Authorization is usually under the control of the service provider. A generic authorization framework, as shown in Fig.3 defined by the ISO Access Control Framework X.812 standard for cross domain service invocation.

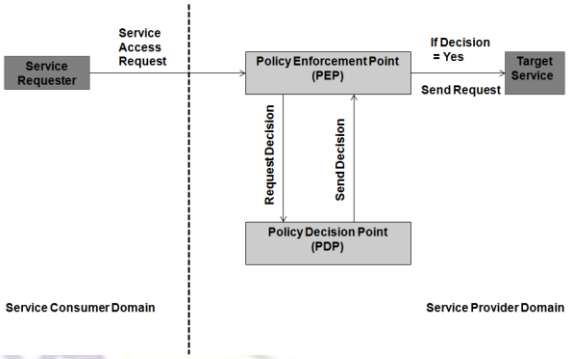

Fig. 3 Generic Authorization Framework

Two key components to support authorized access to the target are -

- Policy Enforcement Point (**PEP**) - The PEP ensures that all requests to access the target service go through the PDP.
- Policy Decision Point (**PDP**) - PDP makes the authorization decision based on a set of rules or policies. In SOA paradigm policy languages are xml based.

By default unless a PDP explicitly determines a request to be valid and access should be granted, it is set to deny access.

Some of the commonly used authorization techniques are mentioned below -

1. Community Authorization Service (CAS) - The main idea of CAS is that a resource owner delegates the allocation of authorization rights to a community administrator and lets the community administrator determine who can use this allocation. The main component used in CAS server, which decides whether a user has sufficient privileges and give the user the rights to perform the requested actions depending on their role in the community, which is established through Role Based Access Control (RBAC).
2. PrivilEge and Role Management Infrastructure Standard (PERMIS) - It is an advanced authorization infrastructure based on the X.509 Privilege Management Infrastructure (PMI). In PMI, an authority issues X.509 attribute certificates (ACs) to users and an AC is used as a credential to store a binding between a user's distinguished name and the user's privileges. In PERMIS access control decisions are made based upon users' attributes, not just upon their organizational roles as in conventional Role Based Access Systems.

3.4 Trust Management Systems

In a large distributed environment communication over internet, creating a single local database of all

potential requesters to a service is not a wise solution. Furthermore a potential user can not always be predictable and domain administrator might not have proper information about the user. In addition to that authorization cannot be purely based on, since security in modern distributed systems utilizes more sophisticated features like delegation, separation of duties etc. These problems can be addressed by the use of trust management systems.

Trust management automates the process of determining whether access should be allowed, on the basis of policy, rights, and authorization semantics.

3.6 Data Confidentiality and Integrity Measures

Data protection is an important aspect of SOA security paradigm.
SOA message transfer is from service to service (rather than source to destination) it is significant to provide data protection to incremental message content.
XML being a language of message exchange in SOA, traditional data protection mechanisms like encryption and digital signature are extended to work with XML data.
Protection is at individual data item level rather than at message level or document level

3.6.1 XML Encryption

With XML encryption one cam encrypt part of document or complete. We can encrypt one or all of the following portions of an XML document:

1. The entire XML document
2. An element and all its sub-elements
3. The content portion of an XML document
4. A reference to a resource outside of an XML document

The steps involved in XML encryption are as follows:

1. Select the XML to be encrypted (all or part of an XML document).
2. Convert the data to be encrypted in a canonical form (optional).
3. Encrypt the result using public key encryption.
4. Send the encrypted XML document to the intended recipient.

3.6.2 XML Digital Signature

Usually digital signature is calculated over the complete message. It cannot be calculated on part of a message. This is because message digest which is used in digital signature is calculated on whole message. But in practice users may want to sign only specific portions of a message. For example, in a purchase order, the purchase manager may want to authorize only the quantity portion, whereas the accounting officer may want to authorize only the rate portion, this is mainly because they are responsible to share, update or decide upon different information.
In such cases XML digital signatures can be used. This technology treats a message or a document as consisting of many elements, and facilitates for the signing of one or more such elements.

3.7 Security design principles

While designing an application which is SOA based, more attention should be given to make it secured since it can be consumed by entities not in same administrative domain as that of the service provider. To ensure information security in SOA based application certain key aspects need to be considered as below. [12]

- There should be generic service contract to expose service interface so that it can be consumed and followed by different service consumers irrespective of administrative domain.
- Security mechanisms should be policy based and platform independent so that they can be easily implemented.
- There is always tradeoff between loosely coupling and security of a service. The extent of loose coupling should be optimized so that it will allow only necessary and sufficient metadata in the service contract and at the same time provide enough security.
- Define and clarify information security requirements which can be reused for different contexts.
- Build a trust component.

IV. PROPOSED SOLUTION

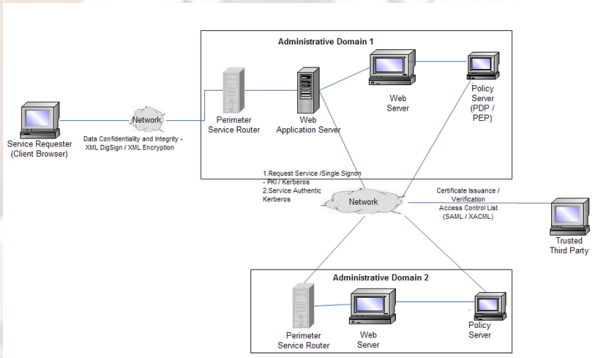

Fig. 4 Proposed System

Taking into consideration different aspects of security with reference to SOA as discussed in the earlier sections, the system proposed as shown in Fig.4 is a comprehensive approach for secured service oriented framework**.** To ensure complete security in the system, it is required to deploy all elements of security viz. Authentication, Authorisation, Confidentiality, Integrity and Non-Repudiation. Also it is required to ensure seamless talk between different administrative domains.

Main Components of the system are as below –

1. Service Requester-It is a client who wants to call a service. In most cases it is client side web browser.
2. Perimeter Service Router – The perimeter service router provides an external interface on the perimeter network for internal Web services. It accepts messages from external applications and routes them to the appropriate Web service on the private network. Thus acting as an intermediary for the internal web services it can monitor traffic and allows only legitimate traffic to enter into the private network and use its resources.
3. Web Application Server – Web Application Server host services under one organisation domain. The service can be a simple service or it can be collection of different services interacting with each other through service orchestration. Web Application server takes care of clubbing, interaction and integration of different web services within and across the organisations.
4. Web Server – Web server hosts simple services within one organisation boundary.
5. Policy Server – Policy Server takes care of policy enforcement, decision making, authorisation decisions etc. It consists of Policy Enforcement Point, Policy Decision Point. It interacts with trusted third party for certificate verification and validation
6. Trusted Third Party – It is third Party entity trusted by different administrative domains who are participating in some service invocation. It verifies, validates service providers and issues certificates.

Security Mechanisms

1. Different Security mechanisms can be implemented at different levels of communication.
2. Data Confidentiality and Integrity needs to be ensured over the network during service invocation and response. XML DigSign, XMLEncryption techniques can be used to achieve this. This will ensure message level security.
3. While invoking services authenticity and authorisation of client is important. This is to ensure information access to legitimate user only. PKI and Kerberos can be deployed to ensure the same. This involves trusted third party to issue and then to verify certificates.
4. It also ensures seamless integration of services across different organisational domains.
5. Authorisation policies, parameters should be communicated correctly and securely also appropriate decision needs to be communicated. XML based communication with (Security Assertion Markup Language) SAML and (eXtensible Access Control Markup Language) XACML can be used to achieve this.

V. CONCLUSION AND FUTURE SCOPE

Following points need to be considered while designing security framework for service oriented architecture

1. Increasing demand of resource sharing across organizations.
2. Data exposure to wide range of known, unknown users.
3. Least control over services due to cross domain interactions.

Thumb rules to be observed are

1. Create Security awareness among all stakeholders at all levels.
2. Prepare, Monitor and Enforce Security Policies.
3. Security should be considered at different levels of application development, right from requirement analysis till deployment, with a vision of prospective threats.

Security can be enhanced with by taking proper measures at operating system level, network level, Application Level and Data storage level.

With the extension of SOA towards the cloud environment, systems are becoming more susceptible to security threats. Current security measures are addressing the security part to a greater extent. But new emerging business models and exponential enhancement on technology side to cope up with this changing paradigm are posing more challenges. In addition to the security aspects considered in this paper, challenges related to multi-tenancy, accounting, billing, policy integration have to be addressed meticulously.

REFERRENCES

[1] "A Review of Grid Authentication and Authorization Technologies and Support for Federated Access Control, WEI JIE, Thames Valley University et al.- ACM Computing Surveys, Vol. 43, No. 2, Article 12, January 2011.

[2] "Service oriented architectures: approaches, technologies and research issues", Mike P. Papazoglou · Willem-Jan van den Heuvel - The VLDB Journal (2007) 16:389–415 DOI 10.1007/s00778-007-0044-3.

[3] "Computationally Efficient PKI-Based SingleSign-On Protocol PKASSO for Mobile Devices", Ki-Woong Park, et al. - IEEE TRANSACTIONS ON COMPUTERS, VOL. 57, NO. 6, JUNE 2008.

[4] "How Kerberos Authentication works",http://learn-

networking.com/network-security/how-kerberos-authentication-works.

[5] ”Kerberos Explained”, http://technet.microsoft.com/en-us/library/bb742516.aspx

[6] “Basic Components of a Public Key Infrastructure”, http://technet.microsoft.com/en-us/library/cc962020.aspx

[7] “Kerberos: An Authentication Service for Computer Networks” , http://gost.isi.edu/publications/kerberos-neuman-tso.html

[8] ”Core PKI Services: Authentication, Integrity, and Confidentiality ” http://technet.microsoft.com/en-us/library/cc700808.aspx

[9] ”Introduction to Digital Certificates”, http://www.verisign.com.au/repository/tutorial/digital/intro1.shtml#step1

[10] ”Formalizing Service Oriented Architectures”, Khalil A. Abuosba and Asim A. El-Sheikh, - Published by the IEEE Computer Society July / August 2008

[12] “Towards An Information Security Framework For Service-oriented Architecture”, Jacqui Chetty, Marijke Coetzee – published in Information Security for South Africa (ISSA), 2010.

[13] A Secure Information Flow Architecture for Web Service Platforms, Jinpeng Wei, Lenin Singaravelu, and Calton Pu,- IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 1, NO. 2, APRIL-JUNE 2008