SOA Security Design and Implementation

11-25-2008

Enterprise IT Applications & Security [1]

- Security Requirements & Technologies
  - Making and verifying claimed identity
    - Username/password
    - Username and password digest
    - Kerberos
    - Digital signature
    - Authentication using JAAS against a variety of repositories
  - Protecting data confidentiality
    - Point-to-point secure transport with SSL (Secure Socket Layer)
    - Selective encryption with shared secrets, PKI, or Kerberos
  - Verifying data integrity and non-repudiation
    - Point-to-point secure transport with SSL
    - Selective signing with PKI or Kerberos

- Existing security infrastructure & technologies
  - Preconfigured users, groups, and roles
  - LDAP (Lightweight Directory Access Protocol) directories
    - Serving data on individuals, system users, network devices and systems over the network for e-mail client, applications, applications requiring authentication or information
  - Firewalls
  - PKI (Public Key Infrastructure)

- Different security solutions for different applications
  - Enterprise firewall
  - Trust boundaries

- Service invocation chain
  - Support both internal & external services
  - Accesses services that wrap different systems and applications
  - This model breaks not only the application boundaries but also the application centric security model
  - The service implementation bridges multiple disparate applications into one service environment
  - This shift requires a new security approach, above and beyond the existing application security

Building Blocks of SOA Security [4]

- WS-Security
- Authentication (Username Token/Password)
- Kerberos (The network authentication protocol)
- XML Encryption
- XML Signature
- WS-SecureConversation
- JAX-RPC Handlers
- JAAS (Java Authentication and Authorization Service) framework
- JGSS (Java bindings of the General Security Services) API
- Apache XML-Security

**Standards for Implementing Security as a Service** [4]
- Standards/Technology  
  - WS-Addressing  
  - Application-Oriented Network (AON)  
  - Security Assertion Markup Language (SAML)  
  - WS-Trust  
  - SAML protocol

**Service Layer Security** - add a security layer spanning existing security infrastructures with the following responsibilities: [1]
- Accommodation of multiple application platforms
- Provide security management and identity propagation/management across multiple security domains (internal, external, business unit silos)
- Support multiple security credentials (Kerberos, SAML, various Token Profiles, PassTickets)
- Support multiple transport protocols (HTTP/S JMS, MQ)
- Maintain the thread of identify across the service Boundaries

**Architecting Security for Service-Based Solutions**
- Using a Security Gateway
- Using an Interceptor in Security Implementations
- Security Service (security as a service)

**Security Gateway**
- Also called XML firewall or XML proxy
- Can be a software package or hardware appliance that filters service traffic upstream of a service and blocks unauthorized traffic before it can reach a protected service
- Untrusted traffic => Security Gateway => Trusted Traffic
- Access control rule  
  - Security token  
  - Authentication with username /password  
  - Password digest-based authentication (not secure)  
    - Hide the password without encrypting the entirety of a message  
    - Do not send password p, instead send f(p) which is a function that transform password p.  
    - The receiver can recomputed f(p), but the  
  - Security Hash Algorithm-1 (SHA-1) – not completely secure
Use f(p,n) function where n is called nonce, a number that is never repeated, use only once.

SAML (Security Assertion Markup Language)

- Implementation
  - Reuse the existing security infrastructure (preconfigured users, groups, and roles)
  - Adapters to existing security technologies such as LDAP directories, traditional firewalls, and PKI infrastructures

- Drawbacks
  - Leaves the actual service endpoint address unprotected

- Approaches to deals with drawbacks
  - Restricting access to the service endpoint to a limited number of physical nodes
  - Mutual authentication between the gateway and the service endpoint

Using an Inceptor in Security Implementations

- The inceptor, or agent, brings the security implementation directly to the service endpoint through the use of platform-specific hooks such as
  - Internet Server Application Programming Interface (ISAPI) filters
  - Java API for XML-based RPC (JAX-RPC) handlers
  - Japa API for XML-based Web Services (JAX-WS) handlers
  - MQ exits

- Policy-aware environments
  - Microsoft’s Web Services Extension (WSE)
  - Window’s Communications Foundation (WCF)
  - IBM’s WebSphere Application server (WAS)

- Untrusted Traffic => Interceptor ++ Service Implementation ⇔ Security Service ⇔ Interceptor ++ Service Implementation => Trusted Traffic

Example: Architecting Security for Policy Issuance Solutions

Service participating in a policy issuance solution exchange a significant amount of sensitive information. As a result a proper security implementation is paramount for this solution.

Using an interceptor is the most appropriate approach for the security implementation of the policy issuance solution. Every participants service should contain an interceptor supporting (and enforcing) the service’s security policy.

For ACME’s services implemented in Java, this interceptor can be implemented as a JAX-WS handler, invoked as part of the service invocation.

For services implemented using WCF, programmatic WCF security can be used as a security interceptor.

**Security Related References**


M. Al-kofahi, S. Chang, and T.E. Daniels, “SCWIN An Integrity Model for SOA Networks,” 2008 IEEE Int. Conf. on Web Services, pp. 675-682.


Secure Socket Layer (SSL), http://en.wikipedia.org/wiki/SSL


