



**Livre Blanc de sécurité SOA.
Sécurité niveau Transport. v.1.0**

Project Documentation

Table of Contents

1 Introduction	
1.1 Avant Propos	1
2 La Mise en Oeuvre	
2.1 Sécurité Niveau Transport	3
2.2 Les Solutions Open Source	5
3 Annexes	
3.1 Où trouver les spécifications?	6

1.1 Avant Propos

«Les services Web XML vont rouvrir 70% des chemins d'attaques fermés par les parefeu lors de la dernière décennie. Ils peuvent transporter virtuellement toutes les données utiles sur le port 80 et le pare-feu ne peut les arrêter.»

- Gartner Group , 2003

Les services Web apportent des bénéfices significatifs pour des applications basées sur l'Architecture Orientée Services, mais exposent des risques importants en terme de sécurité. Créer et gérer un environnement sécurisé pour les services web nécessite la manipulation et la maîtrise de spécifications et standards divers et variés ainsi que des technologies et logicielles et matérielles conséquentes.

Il convient d'étudier la mise en œuvre de la sécurité pour les Architectures Orientées Services (SOA) via les quatre volets suivants :

- **La sécurité niveau Transport** : pare-feu (firewall), VPN (Virtual Private Networks), authentification basique, non-répudiation et cryptage
- **La sécurité niveau Message** : Utilisation des jetons de sécurité afin de valider l'identité du consommateur du service ou du processus, utilisation des assertions d'autorisation pour valider l'accès au services.
- **Sécurité niveau application** : Sécuriser les composants appelés par les Web Services, EJBs, Servlets appelés via les services Web.
- **Sécurité niveau Données** : Cryptage et signature des messages afin de protéger les données stockées ou transmises.
- **Sécurité niveau Environnement** : Monitoring, logging et audit afin d'identifier les problèmes qui doivent être fixés et résolus et établir des communications sûres et fiables.

L'émergence des services web pose plusieurs problématiques dont celle de la sécurité des échanges de messages entre partenaires. Dans une architecture Orientée Services, les services web peuvent exposer des processus métier sensibles qui nécessitent un traitement particulier en terme de sécurité aux deux bouts du canal de communication. En plus, les services web sont des technologies récentes, ceci implique de nouvelles vulnérabilités et attaques ou menaces.

Les services web sont utilisés dans les cas suivants (la liste n'est certainement pas exhaustive) :

- Intégration de systèmes point-à-point
- Intégration d'applications entreprise
- Collaboration et partenariat Business
- E-Business
- Composition des processus métier
- Protection et ouverture des systèmes d'information
- Réduction des coûts du cycle de vie du développement et la maintenance des systèmes d'information

Les solutions de sécurité des services web doivent prendre en charge les concepts suivants :

- L'authentification
- L'autorisation
- La confidentialité
- L'intégrité
- La non-répudiation

En plus de ces concepts, le système de sécurité doit prendre en charge l'audit des actions et messages envoyés afin de tracer l'activité de la sécurité des web services.

Les utilisateurs des services web doivent être identifiés soit via un nom d'utilisateur combiné à un mot de passe soit via un certificat digital. Une fois l'utilisateur identifié, il doit posséder l'autorisation ou l'habilitation nécessaire afin d'effectuer le traitement qu'il a demandé. Toutes informations ou messages sensibles mis en jeu par le traitement doivent être confidentiels et ne doivent pas subir d'altération qui touche à son intégrité d'origine. Une fois le traitement exécuté, des mesures de non-répudiation doivent être mises en place afin d'éviter tout dénis des deux parts (consommateurs/fournisseur).

Les services web reposent sur des topologies applicatives diverses et variées telles que : l'Internet mobile, des passerelles, zones démilitarisées (DMZ), systèmes distribués La communication entre ces technologies s'effectue via des intermédiaires.

La sécurité n'était pas la priorité des organisations travaillant sur les spécifications de la stack WS. La sécurité au niveau de la couche de Transport n'étant pas suffisante, il demeurait un vide qui freinait l'adoption des services web. Heureusement, plusieurs propositions majeures ont été diffusées afin de combler ce vide dont **WS-Security** qui apporte un support globale de l'intégrité, de la confidentialité et l'authentification des messages et **SAML** qui définit un langage commun d'interopérabilité permettant de partager les informations liées à l'authentification et l'autorisation afin de faciliter la mise en place de fonctionnalités SSO et de permettre la délégation des droits. Il est intéressant à noter que d'autres propositions et spécifications tendent à émerger comme **Kerberos** , **XKMS** , **XACML** afin d'apporter un support complémentaire à la stack de sécurité.

La sécurité des messages est d'autant plus nécessaire si des intermédiaires sont présents dans une communication point-à-point. L'expéditeur d'origine et le destinataire final doivent établir des relations de confiance avec ces intermédiaires afin d'assurer la sécurité de bout en bout.

2.1 Sécurité Niveau Transport

La sécurité niveau transport fournit la confidentialité et l'intégrité des données transportées entre deux applications, mais aussi l'authentification de ces dernières. Le modèle de sécurité le plus utilisé dans le monde de l'Internet est la combinaison entre SSL et HTTP Basic Authentication. HTTP Basic Authentication nécessite une user Id et un mot de passe et SSL étant un protocole sécurisé de transmission de données via des méthodes de cryptage.

Le protocole SSL (Secure Socket Layer) est une solution de sécurité transparente aux applications qui sont basées sur le protocole TCP. Il reste un protocole générique, mais ne couvre pas tous les besoins d'applications telles que des applications de paiement par Internet. Il est important de signaler que le protocole SSL avait pour ambition première de sécuriser le protocole http.

La sécurité niveau transport peut être gérée via les standards suivants :

- SSL (Secure Socket Layer) : repose sur un procédé de cryptographie par clef publique afin de garantir la sécurité de la transmission de données sur Internet Le système SSL est indépendant du protocole utilisé.
- Au milieu de l'année 2001, le brevet de SSL appartenant jusqu'alors à Netscape a été racheté par l'IETF (Internet Engineering Task Force) et a été rebaptisé pour l'occasion TLS (Transport Layer Security).

SSL/TLS offrent des dispositifs de sécurité dont l'authentification, l'intégrité et la confidentialité des données. SSL/TLS assurent la sécurité de session point-à-point.

- IPSec est un protocole destiné à fournir différents services de sécurité. Il propose ainsi plusieurs choix et options qui lui permettent de répondre de façon adaptée aux besoins des entreprises, nomades, extranets, particuliers, etc... Néanmoins, son intérêt principal reste sans conteste son mode dit de tunneling, c'est-à-dire d'encapsulation d'IP qui lui permet entre autres choses de créer des réseaux privés virtuels ou VPN. Cette technologie a pour but d'établir une communication sécurisée (le tunnel) entre des entités éloignées, séparées par un réseau non sécurisé voir public comme Internet, et ce de manière quasi-transparente si on le désire.

Malgré le fait que les protocoles décrits ci-dessous sont adaptés aux applications basées sur le protocole TCP ou HTTP. Ils restent limités et insuffisants dans un contexte de communication basé sur les services web. En effet :

- les messages SOAP peuvent inclure plusieurs intermédiaires, ce qui nécessite un contexte de sécurité partagé et interopérable entre les différentes parties mises en jeu, ce qui n'est pas le cas de SSL vu qu'il suppose que seules deux parties sont concernées par une transmission de données.
- SSL s'occupe de crypter le message en totalité. Dans le cas d'une transaction mettant en jeu plusieurs participants, il se pourrait que plusieurs parties des données transmises soient cryptés séparément et différemment pour que chaque participant décrypte la partie le concernant.

Les Avantages de ma sécurité niveau transport

Parmi les avantages de la sécurité niveau transport, on pourrait dénoter les points suivants :

- TTPS fournit un support performant et rapide à mettre en place.
- L'authentification peut reposer sur une authentification HTTP basic ou via des certificats Clients.
- La sécurité niveau transport fournit l'intégrité des données entre le client et le serveur HTTP en utilisant des clefs cryptographiques asymétriques.
- Elle fournit aussi un niveau de confidentialité.
- Il existe dans le marché du Hardware plusieurs types d'accélérateurs.
- SSL/TLS sont des technologies matures et dont l'implémentation présente rarement des problèmes d'interopérabilité.
- SSL/TLS crypte le message y compris l'entête et le corps ainsi que toutes les données attachées.

La sécurité niveau transport peut être considérée comme optionnelle et peut être remplacée par la sécurité niveau messages. Mais il est intéressant de prendre en compte les considérations ci-dessous :

- Il est préférable d'utiliser la sécurité niveau transport s'il n'est pas nécessaire de partager un contexte de sécurité entre le service et son client.
- La sécurité niveau transport serait à prendre en compte s'il n'existe pas des intermédiaires ne sont pas mis en jeu entre un service et son client.

2.2 Les Solutions Open Source

Il existe plusieurs projets travaillant sur l'implémentation de spécifications relatives à la sécurité des services Web. La majorité de ces implémentations utilisent les langages Java ou C++. Ci-dessous un descriptif des projets les plus conséquents :

- Apache XML Security : Ce projet a pour but de fournir une implémentation des standards de sécurité relatifs à XML. Ces standards sont XML-Signature et XML Processing. Le projet vise de travailler sur les spécifications XML Key Management (XKMS)
- OpenSAML consiste en un ensemble de bibliothèques Open Source Java et C++ qui sont en conformité avec les spécifications SAML 1.0 et 1.1.
- Apache Directory Project : Ce projet livre un module de sécurité comportant :
 - Un framework d'authentification : AuthX
 - Une librairie compatible RFC Kerberos
 - ChangePw pour changer les mots de passe de manière sécurisée.
- VeriSign WS-Security toolkit : Librairie Open Source pour faciliter l'utilisation et l'intégration de WS-Security dans le développement des applications sécurisées basées sur les services Web.
- La Crypto API de Bouncy Castle inclut les points suivants:
 - Une API légère de cryptographie en Java.
 - Un fournisseur pour la JCE et JCA.
 - Une implémentation convenable de la JCE 1.2.1.
 - Une bibliothèque pour lire et écrire des objets encodés en ASN.1.
 - Des Générateurs pour les versions 1 et 3 des certificats X.509 et des fichiers PKCS12.
 - Des Générateurs pour la version 2 des certificats de l'attribut X.509.
 - Des Générateurs/Processeurs pour S/MIME et CMS (PKCS7).
 - Des Générateurs/Processeurs pour OCSP (RFC 2560).
 - Des Générateurs/Processeurs pour TSP (RFC 3161).
 - Des Générateurs/Processeurs pour OpenPGP (RFC 2440).
 - Une version jar signée, appropriée pour les JDK 1.4/1.5 et la JCE de Sun.

3.1 Où trouver les spécifications?

SAML	http://www.oasis-open.org/committees/security/
Security Services TC	http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
WS-Federation	http://www-106.ibm.com/developerworks/webservices/library/ws-fedworld/
WS-Security	http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
WS-SecureConversation	http://www-106.ibm.com/developerworks/webservices/library/ws-secon/
WS-SecurityPolicy	http://www-106.ibm.com/developerworks/webservices/library/ws-secpol/
WS-Trust	http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-trust.asp
XML-Encryption	http://www.w3c.org/Encryption/2001/
XML-Signature	http://www.w3c.org/Signature/