DOSSIER DE SECURISATION DES WEB SERVICES

ABREVIATIONS

Abréviation	**Signification**
SOAP	Simple Object Access Protocol
HTTP	Hyper Text Transfer Protocol
SSL	Secure Socket Layer
TLS	Transport Layer Security
UDP	User Datagram Protocol
SMTP	Simple Mail Transfer Protocol

REFERENCES

Abréviation	**Signification**
[DOSSIERSECWS]	Dossier_Securisation_Web_Services_v1r0.pdf
[ANASEC]	Annexe1_Analyse_Sécurité_Préalable_v1r0.pdf
[FILTRAGE]	Annexe2_Etude_ Securistation_WS_FiltrageIP_v1r0.pdf
[SSL]	Annexe3_Etude_ Securistation_WS_TunnelSSL_v1r0.pdf
[SYNAPSE]	Annexe4_Etude_ Securistation_WS_Synapse_v1r0.pdf
[SPRING]	Annexe5_Etude_ Securistation_WS_SPRING-Security_v1r0.pdf
[PROTOTYPE]	Annexe6_Etude_ Securistation_WS_Prototype_v1r0.pdf

TABLE DES MATIERES

TABLE DES ILLUSTRATIONS

1.INTRODUCTION ET CONTEXTE

La diffusion, par l'Agence, des premiers Web Services autour d'Harpège et d'Apogée a mis en avant un besoin de montée en compétence dans les établissements sur le thème de la sécurité associée à ces technologies. Aussi, l'Agence a mené, au second trimestre 2008, une étude visant à proposer des solutions pragmatiques et opérantes pour accroître le niveau de sécurisation de ces composants.

Un groupe de travail (1) a donc été constitué et accompagné par une prestation d'expertise confiée à la société GFI Informatique. A l'occasion de cette publication, l'Agence tient à adresser ses remerciements aux établissements ayant contribué à ce chantier.

Le présent dossier propose le résultat de ces travaux collectifs pour que chaque établissement dispose d'une boite à outils dans laquelle il puisse choisir les solutions de sécurité des Web Services qui lui semblent optimales et adaptées à son contexte qu'il s'agisse ou non des Web Services Amue.

Les thèmes majeurs de la sécurité des Web Services sont traités au travers de cette analyse (Confidentialité et intégrité des échanges, identification et authentification des entités communicantes, contrôle d'accès et autorisation).

Sept stratégies unitaires et modulaires sont proposées dans ce dossier. Pour celles requérant une implémentation technique, une annexe détaille les différentes étapes de leur mise en œuvre et fournit des éléments techniques pragmatiques. Ces annexes techniques sont agrémentées d'exemples de code pour faciliter l'implémentation de ces technologies et ainsi guider les établissements dans leur utilisation.

Au final, quelques exemples de combinatoires de ces stratégies unitaires sont présentés pour que les établissements puissent construire leur propre politique de sécurité des Web Services au sein de leur Système d'Information.

(1) Participants au groupe de travail :

Nom	Etablissement
R. Bourges	Université Rennes 1 / ESUP
D. Collart	Université Pierre et Marie Curie
F. Gravelat	Grenoble Universités
F. Jammes	Université Paris 1
V. Joussot	Université Toulouse 1
J. Kerleau	S.I.I.G Strasbourg
C. Le Roy	Université de Perpignan
V. Senges	Université Toulouse 1
J.M. Thia	Université Pierre et Marie Curie
C. Vigneron	S.I.I.G Strasbourg
O. Ziller	Université Nancy 2
B. Fabregue	AMUE – Pôle Intégration DEI
D. Rongeat	AMUE – Pôle Intégration DEI
F. Soldevila	AMUE – Pôle Intégration DEI

2.OBJECTIFS ET MODALITES DE L'ETUDE

2.1. CADRAGE ET RAPPEL DES OBJECTIFS DE L'ETUDE

2.1.1. LES WEB SERVICES CONSIDERES

Les web services considérés dans le cadre de cette étude sont constitués de messages de type SOAP transportés par le protocole HTTP.

2.1.2. LE CONTEXTE D'UTILISATION DES WEB SERVICES

Les web services qui sont la cible principale de l'étude proviennent des applications HARPEGE et APOGEE. Ces web services sont principalement utilisés dans un contexte interne, c'est-à-dire qu'ils sont appelés par des applications appartenant aux établissements membres de l'AMUE. Des utilisations dans un contexte externe sont actuellement en cours de mise en œuvre.

En outre, les résultats de cette étude ne s'appliquent pas uniquement à ces web services, ils peuvent s'appliquer également à d'autres web services SOAP/HTTP de l'architecture SOA proposée par l'AMUE ou encore développés en interne par les établissements.

2.1.3. LES OBJECTIFS DE L'ETUDE

L'objectif de l'étude est d'apporter des solutions pragmatiques de sécurisation des Web Services de l'architecture SOA de l'AMUE et des établissements. Les cinq thèmes à aborder durant l'étude sont :

- La confidentialité des échanges lors de l'interrogation des web services,
- L'intégrité des échanges lors de l'interrogation des web services,
- L'identification des entités communicantes,
- L'authentification des entités communicantes,
- Le contrôle d'accès et l'autorisation.

Par conséquent, lorsque les termes génériques « sécuriser » et/ou « sécurisation » sont employés à propos des web services, cela sous entend que l'on parle d'**au moins un des cinq thèmes évoqués ci-dessus**.

Il est possible que certaines solutions présentent une richesse fonctionnelle plus importante et propose des fonctions de sécurité traitant d'autres thèmes, auquel cas, on y fera référence de manière spécifique.

Précision sur les termes utilisés :

- Identification : action permet de *connaitre* l'identité d'un appelant. Ex : le login
- Authentification : action qui consiste à *vérifier* cette identité. Ex : mot de passe
- Autorisation : action qui consiste à *vérifier* que les droits de l'identité sont d'un niveau supérieur ou égal à ceux de la ressource demandée.

2.2. ORGANISATION DE L'ETUDE

L'étude apporte un ensemble de solutions pour sécuriser les web services selon les différents thèmes à aborder et des approches préconisées.

L'étude a été découpée en plusieurs phases.

- Un état de l'art qui a permis de recenser les différentes solutions unitaires. Une présentation a été effectuée le 4 Juin 2008 à Paris devant les établissements présents.
- Une phase de 2 itérations qui a permis d'étudier dans le détail quelques solutions (Synapse, SPRING security) et qui a donné lieu à deux réunions téléphoniques puis à une présentation à Paris devant les établissements présents. Un prototype développé pendant cette phase y a été présenté.
- Une phase de rédaction des différents livrables qui clôture cette étude.

2.3. IDENTIFICATION DES CRITERES POUR LE CHOIX DES SOLUTIONS

Nous avons identifié un certain nombre de critères permettant de qualifier les solutions présentées dans cette étude.

Les critères d'appréciation des solutions étudiées sont les suivants :

- Couverture fonctionnelle,
- Interopérabilité,
- Adéquation aux technologies en place dans les établissements
- Coût de mise en œuvre,
- Coût d'exploitation,
- Evolutivité,
- Référence à un ou plusieurs standards,
- Difficulté de mise en œuvre,
- Niveau de compétence requis (installation, administration, exploitation),
- Degré de maturité de la solution,
- Support disponible,
- Réactivité dans les évolutions et les corrections,
- Typologie de solution (open source ou éditeur),
- Niveau d'intégration de la solution dans le SI (côté serveur, côté client,…),
- Adaptabilité à différentes échelles de volumétrie,
- Conformité avec les préconisations des documents nationaux.

3. LES APPROCHES CONSIDEREES

3.1. PROPOSITION DE DEMARCHE

Nous proposons deux approches distinctes de sécurisation des web services :

- La sécurisation au niveau flux,
- La sécurisation au niveau du message SOAP.

La sécurisation par les couches basses s'appuie sur les protocoles de transport des web services au niveau des couches 3 et 4 de la pile TCP/IP pour mettre en place une session sécurisée au dessus de la couche transport.

La sécurisation au niveau du message SOAP est beaucoup plus fine car l'implémentation des mécanismes de sécurité s'effectue au cœur de la logique applicative.

Quelle que soit l'approche, il est souhaitable de s'appuyer sur des standards le plus possible. Ce sera donc le cas dans cette étude dans la mesure du possible.

L'étude a fait ressortir 7 stratégies unitaires qui vont être développées dans la suite de ce document.

- Stratégie 1 : Ne rien faire
- Stratégie 2 : Filtrage IP
- Stratégie 3 : Utilisation d'un tunnel SSL/TLS
- Stratégie 4 : Utilisation de Spring-Security
- Stratégie 5 : Utilisation de WSS4J
- Stratégie 6 : Utilisation de JBOSSws
- Stratégie 7 : Utilisation d'une passerelle XML

3.2. STRATEGIE 1 : NE RIEN FAIRE

Couches concernées : Aucune

Thèmes adressés : Aucun

Ce scénario consiste à réaliser une analyse de risque du SI afin de déterminer si le coût de la sécurisation est inférieur au coût du risque. En effet afin de ne pas réaliser de la sur-sécurité il est indispensable que pour chaque projet informatique, un dossier sécurité soit réalisé. Ce dernier synthétise dans un document unique les choix et justifications des besoins de sécurité.

Le dossier de sécurité a notamment pour objectif de faire prendre aux maîtrises d'œuvre les dispositions nécessaires pour respecter les exigences de sécurité, et de permettre aux maîtrises d'ouvrage de contrôler la prise en compte effective de leurs exigences. Il marque l'engagement des différents acteurs à respecter les orientations et options de sécurité retenues, et doit servir de cadre pour les phases de réalisation,

d'exploitation et de maintenance. Habituellement constitué au fur et à mesure du déroulement des phases de conception, le dossier de sécurité est généralement structuré autour des chapitres suivants :

- Présentation générale du projet
- Objectifs et besoins de sécurité
- Menaces sur l'information
- Fonctions de sécurité
- Synthèse des architectures de sécurité
- Sécurité et dégradation de service
- Prise en compte de la sécurité au cours du projet
- Vulnérabilités résiduelles
- Coûts

La conclusion de ce dossier vis-à-vis de la problématique de sécurité abordée pourra être de ne rien faire si le niveau de sécurité existant est suffisant vis-à-vis du risque ou si le coût de la sécurisation à mettre en œuvre est supérieur au coût estimé du risque.

3.3. STRATEGIE 2 : FILTRAGE IP

Couches concernées : Réseau

Thèmes adressés : Identification, Authentification

L'une des premières mesures à mettre en place pour restreindre l'utilisation des web services est le filtrage selon les adresses IP tel que cela est actuellement préconisé dans les dossiers d'exploitation des Web Services AMUE.

Cette mesure de sécurité ne peut pas être considérée comme couvrant le besoin d'authentification (IP Spoofing : changer son adresse IP pour se faire passer pour quelqu'un d'autre.) mais permet un premier niveau de contrôle en identifiant de façon simple, mais détournable, la provenance des flux.

Le principe est de bloquer les IP non autorisées. Le blocage IP est une technique de sécurité courante, disponible sur tous les principaux serveurs web, comme Apache et Internet Information Server (IIS) de Microsoft.

Il s'agit en fait du processus d'identification des adresses IP à partir desquelles les requêtes web seront acceptées. Il est généralement mis en œuvre en spécifiant une liste d'adresses IP autorisées. Chaque fois qu'une requête web est reçue par le serveur, celui-ci compare l'adresse IP qui émet la requête à la liste des adresses IP autorisées. Si elle figure sur la liste, la requête est traitée normalement, sinon le serveur renvoie une erreur HTTP 403.6 : "Accès interdit : adresse IP rejetée". À noter que la plupart des serveurs web offrent également la possibilité de spécifier les adresses IP bloquées, plutôt que les adresses autorisées.

Comme les services web sont généralement utilisés via une simple requête HTTP, le blocage IP fonctionne à l'identique pour les services web et pour des requêtes standard sur site web. Les clients figurant sur la liste autorisée auront la possibilité de faire appel à des services web, ainsi que de consulter les fichiers WSDL sur le site pour en savoir plus sur les services web proposés.

Ce filtrage IP peut aussi être réalisé au niveau d'un pare-feu, dont la fonctionnalité principale est de réaliser un tel blocage.

Points à prendre en compte

Avec le blocage IP, comme toutes les requêtes sont bloquées par le serveur web lui-même, il faut savoir que les clients ne pourront pas accéder à une partie du site web tant que vous n'aurez pas ajouté leur adresse IP à la liste des adresses autorisées. Ce comportement, sauf configuration spécifique, empêche des clients potentiels de consulter les fichiers WSDL pour prendre connaissance des offres de services web. De plus, il est important de noter que les appelants ayant des adresses IP non valides verront leur accès aux pages web de votre site bloqué. Un aspect qui peut avoir son importance, puisque les développeurs placent souvent les pages et les services web dans le même site web pour maximiser la réutilisabilité. Ainsi, sauf configuration spécifique, si vous avez recours au blocage IP pour vos services web, vous devez accepter que la même sécurité soit appliquée à vos pages web, ou bien créer des répertoires virtuels distincts sur votre serveur pour vos sites et services.

L'implémentation du blocage IP est relativement simple, mais le processus varie d'un serveur web à l'autre. Avec IIS, version 5, un appelant peut simplement choisir l'onglet Sécurité du répertoire, situé dans la fenêtre Propriétés du site web, et saisir les adresses IP acceptées. Avec Apache, vous pouvez modifier un fichier .htaccess qui spécifie les adresses pour lesquelles l'accès est autorisé.

Cependant cette technique peut rapidement être lourde à gérer si le nombre d'adresse IP augmente considérablement, ou si le Web Service devient multipoint (nombre d'appelant non déterminé)

Le détail de cette mise en œuvre se trouve dans le document [Filtrage]

3.3.1. LES AVANTAGES ET LES INCONVENIENTS DU FILTRAGE

Le filtrage IP présente les avantages suivants :

- solution techniquement facile à mettre en œuvre.
- coût faible.

Le filtrage IP présente les inconvénients suivants :

- La dimension organisationnelle peut s'avérer être très contraignante sur la gestion d'un grand nombre d'adresses IP.

3.4. STRATEGIE 3 : SECURISATION AU NIVEAU FLUX AVEC SSL

Couches concernées : Transport

Thèmes adressés : Confidentialité, Intégrité, Identification, Authentification

Les web services proposés par l'AMUE utilisent le protocole http pour transporter les messages SOAP. Il est tout à fait possible d'intervenir à ce niveau pour mettre en œuvre la sécurité. L'utilisation du protocole TLS/SSL (Transport Socket Layer/Secure Socket Layer) peut constituer une solution.

L'utilisation de TLS est recommandée car c'est un protocole normé mais nous utiliserons indifféremment les termes SSL ou TLS. Il y a très peu de différences entre SSL version 3 et TLS version 1 (qui correspond à la version 3.1 du protocole SSL) rendant les deux protocoles non inter-opérables, mais TLS a mis en place un mécanisme de compatibilité ascendante avec SSL. En outre, TLS diffère de SSL pour la génération des

clés symétriques. Cette génération est plus sécurisée dans TLS que dans SSL v3 dans la mesure où aucune étape de l'algorithme ne repose uniquement sur MD5 pour lequel sont apparues quelques faiblesses en cryptanalyse.

3.4.1. GENERALITES

SSL est un protocole client/serveur permettant d'établir une session sécurisée entre deux entités communicantes assurant :

- Le chiffrement des données échangées,
- L'authentification du serveur et/ou du client,
- Le contrôle d'intégrité des données échangées.

L'utilisation de SSL dans le cadre des web services s'effectue concrètement en sécurisant le protocole http.

3.4.2. EXPLICATIONS DETAILLEES DU PROTOCOLE

Ce protocole procède principalement selon trois phases :

- La phase de négociation des algorithmes supportés entre les extrémités communicantes,
- L'échange de clés et la phase d'authentification,
- L'établissement d'un chiffrement symétrique et d'un contrôle sur l'authenticité des messages échangés.

3.4.2.1. Les algorithmes cryptographiques

Chaque phase possède des algorithmes cryptographiques robustes bien connus. Voici un tableau résumant les algorithmes :

Phase SSL	Algorithmes cryptographiques
Négociation	Aucun
Echange de clés	Diffie -Hellman
Chiffrement/Contrôle	AES, 3DES, RC4,…

3.4.2.2. Authentification par certificats X509

SSL permet l'authentification du serveur et du client. Dans le contexte d'un échange classique en SSL sur les sites de commerces en ligne notamment, seul le serveur Web s'authentifie auprès du client.

Il s'agit d'une pratique commode car il serait difficile d'exiger la mise en œuvre d'une authentification de la part des appelants finaux.

En revanche dans le contexte de la sécurisation des web services, il est tout à fait possible de mettre en œuvre une authentification mutuelle (le client s'authentifie vis-à-vis du serveur et réciproquement) si le niveau de sécurité l'exige.

Lors d'une session SSL, l'authentification est effectuée grâce à un certificat X509. Un certificat est l'équivalent d'une carte d'identité numérique.

Un certificat X509 contient :

- l'ensemble des informations d'identité de l'entité (client ou serveur),
- la clé publique associée à cette entité,
- la signature de l'ensemble de ces données par une autorité de confiance.

3.4.3. SECURISATION DES WEB SERVICES AVEC SSL

3.4.3.1. Pré-requis

Deux pré-requis majeurs sont à identifier avant de mettre en œuvre une session SSL

- Implémentations de SSL côté client et côté serveur,
- Disposer d'une infrastructure permettant la génération des clés et des certificats X509.

3.4.3.2. Mise en œuvre

Les détails de la mise en œuvre d'un tunnel SSL/TLS se trouve dans le document [SSL]

3.4.4. LES AVANTAGES ET LES INCONVENIENTS DE SSL

SSL présente les avantages suivants :

- SSL est une technologie mâture, éprouvée et maîtrisée,
- SSL est implémenté dans beaucoup de logiciels propriétaires ou open source,
- SSL est assez répandu au sein des systèmes d'informations actuels.

SSL présente les inconvénients suivants :

- SSL ne permet qu'une communication selon un mode point à point,
- SSL ne permet pas de réaliser un contrôle d'accès fin sur les données et les services adressés,
- La gestion des certificats induit une complexité supplémentaire,
- La sécurité de SSL repose essentiellement sur la confiance dans les certificats et dans les clés publiques. Par conséquent, le processus et les outils associés doivent être suffisamment maîtrisés pour limiter le risque d'utiliser des clés publiques contrefaites.

3.5. STRATEGIE 4 : UTILISATION DE SPRING-SECURITY

Couches concernées : Message SOAP.

Thèmes adressés : Identification, Authentification, Autorisation.

ACEGI est un framework permettant d'introduire une couche de sécurité pour les web services au niveau du framework SPRING. ACEGI Security est devenu SPRING Security, projet officiel de sécurité de SPRING. Ce framework fournit un niveau de sécurité pour les applications J2EE. Le principe mis en place par SPRING Security pour sécuriser une ressource consiste en une série de « filtres » interposés entre l'appelant et la ressource elle-même. Ces différents filtres ont chacun un rôle précis dans la chaîne de sécurisation.

SPRING Security permet essentiellement de gérer deux choses :

- L'authentification, qui consiste à garantir que l'entité connectée est bien celle qu'elle prétend être,
- Les autorisations, qui consistent à vérifier que l'entité connectée a bien les permissions d'effectuer une action donnée.

La notion centrale de base utilise la notion de **filtre** autour des servlets pour l'authentification et d'un mécanisme d'interception pour gérer les autorisations.

SPRING Security se place en coupure des Web services. Cela ne requiert donc pas de modification du code existant. SPRING Security fonctionne sur un principe d'application successif de filtres afin de déterminer, dans notre cas, si l'appelant est identifié, puis authentifié, et enfin autorisé. Il y a un donc une chaine de filtres, comme présenté ci-contre.

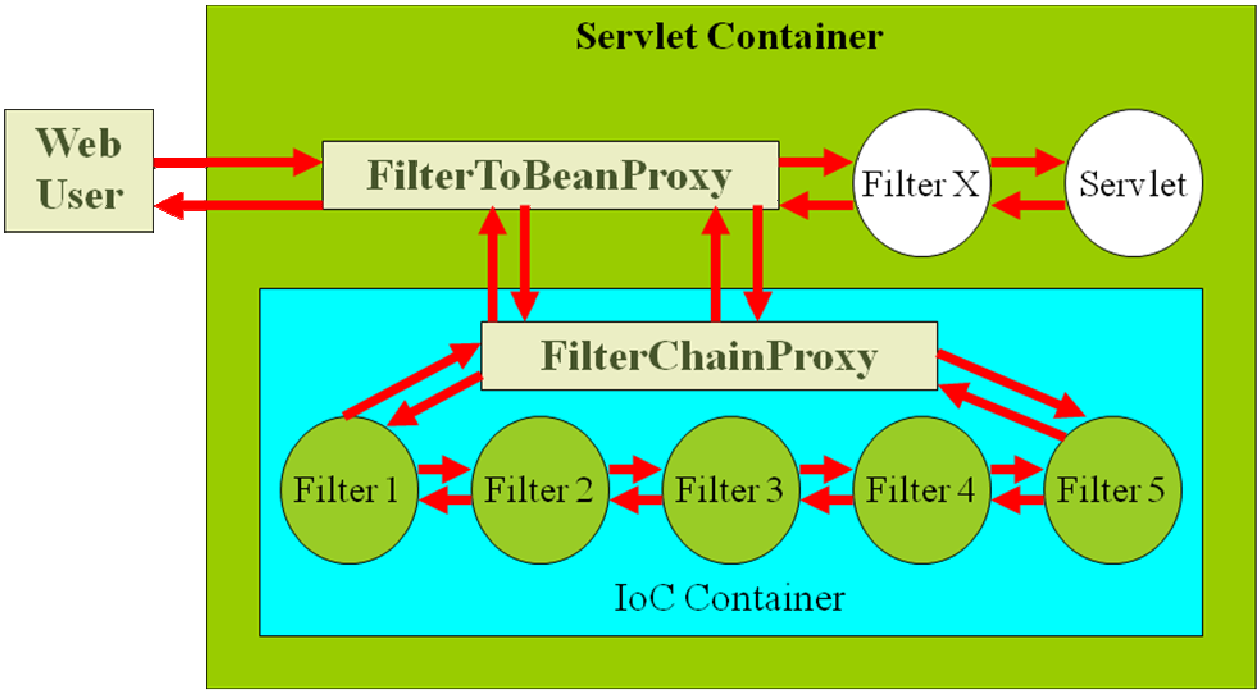

Figure 1 : Fonctionnement des filtres de SPRING-Security

3.5.1. L'AUTHENTIFICATION

La première étape de la sécurisation d'une application Web est l'authentification des entités (appelants) qui la manipulent. Pour cela, il est nécessaire de mettre en place un mécanisme d'authentification permettant ensuite de manipuler les informations propres à l'entité, ses droits, éventuellement ses

groupes, etc. Cette authentification se fait à partir d'informations stockées dans une base de données, un fichier XML, un annuaire LDAP, etc.

3.5.2. LES AUTORISATIONS

Les éléments d'une application pouvant être sécurisés sont :

- Les URLs, c'est à dire les pages et les servlets,
- Les méthodes des beans,
- Les objets eux mêmes.

Les filtres d'autorisation permettent d'interdire ou non à un appel authentifié l'accès à ces ressources

3.5.3. MISE EN ŒUVRE

La mise en œuvre de SPRING-Security est décrite dans le document [SPRING]

3.5.4. LES AVANTAGES ET LES INCONVENIENTS DE SPRING-SECURITY

SPRING-Security présente les avantages suivants :

- La sécurité se traite au niveau message SOAP donc elle s'effectue indépendamment du contexte réseau,
- Cette solution s'appuie sur les technologies utilisées par les établissements et par l'AMUE (SPRING),
- Cette technologie possède une certaine flexibilité quant à son paramétrage (les modifications à apporter restent au niveau applicatif),
- SPRING-Security dispose d'une richesse fonctionnelle importante.

SPRING-Security présente les inconvénients suivants :

- La richesse dans les modèles d'authentifications supportés et la finesse proposée dans les autorisations peuvent induire une complexité de mise en œuvre relativement importante.
- SPRING-Security seul n'implémente pas le standard de sécurisation des Web Services WSS (Web Services Security), il faut installer le Framework WSS4J (Web Services Security 4 Java) pour cela. Il existe le framework SPRING Web Services qui intègre SPRING-Security ainsi que le support de WSS. Ce standard WSS est décrit dans le chapitre suivant.

3.6. STRATEGIE 5 ET 6 : UTILISATION DE WS-SECURITY

Couches concernées : Message SOAP

Thèmes adressés : Confidentialité, Intégrité, Identification, Authentification

3.6.1. UN STANDARD : WS SECURITY

Au cours des dernières années, les différentes initiatives permettant de doter SOAP de mécanismes pour sécuriser les messages ont convergé en 2004 vers un standard : **SOAP Security Message ou Web Service Security.**

WS Security provient d'un travail commun entre IBM, Microsoft et Verisign. L'ensemble de leur travail a été soumis au consortium indépendant OASIS pour initier une démarche de standardisation.

Bien entendu, WS Security n'a pas été la seule initiative de sécurisation des web services. Il existe d'autres spécifications intéressantes qui ont abouti. Mais WS Security est devenu incontournable dans ce domaine.

3.6.1.1. Les fonctionnalités de WS Security

Le standard WS Security adresse les différents thèmes de la sécurité suivant :

- Confidentialité,
- Intégrité,
- Authentification.

3.6.2. LES IMPLEMENTATIONS DE WS SECURITY

Du fait de l'importance croissante des web services et de la facilité pour les implémenter en environnement hétérogène, de nombreux éditeurs proposent les implémentations de WS Security au sein de leur produit. Les communautés open source ne sont pas en reste sur ce sujet.

Pour la partie serveur, il existe des implémentations de WS Security sur :

- les serveurs d'applications,
- Framework 3 tiers,
- Serveur Web

Pour la partie cliente, il existe des implémentations de WS Security sur :

- Framework .NET,
- Environnement Java,
- Librairie PHP

3.6.3. STRATEGIE 5 : SOLUTION AUTOUR DU MOTEUR SOAP AXIS : WSS4J

3.6.3.1. Généralités

WSS4J est une implémentation de WS Security qui se présente sous deux formes :

- Une API indépendante,
- Un module pour AXIS 1, le moteur SOAP.

Les principales fonctionnalités proposées par WSS4J sont les suivantes :

- Instanciation du module permettant de traiter l'en-tête SOAP,
- Signature d'une enveloppe SOAP,
- Ajout d'un jeton d'horodatage,
- Chiffrement du corps d'un message SOAP,
- Ajout d'un UserNameToken pour l'authentification,
- Ajout d'une assertion SAML pour le transport d'informations liées à la sécurité.

3.6.3.2. Les avantages et les inconvénients de WSS4J

WSS4J-Axis présente les avantages suivants :

- La sécurité se traite au niveau message SOAP donc elle s'effectue indépendamment du contexte réseau.
- Cette solution s'appuie sur les technologies utilisées par l'AMUE (AXIS).
- Elle couvre l'essentiel des besoins de sécurité identifiés par l'AMUE.

WSS4J-Axis présente l'inconvénient suivant :

- WSS4J est une implémentation de WS Security qui gère AXIS dans les versions 1.x. Si une montée de version survient sur le moteur SOAP AXIS, il sera indispensable de revoir la stratégie (utilisation de RAMPART…)

3.6.4. STRATEGIE 6 : SOLUTION AUTOUR DES SERVEURS D'APPLICATIONS

La plupart des serveurs d'application propose une implémentation de WSS. Par exemple JBOSS propose un module JBOSSws qui met en œuvre cette norme. Les avantages et les inconvénients de cette solution sont identiques à celle de WSS4J. L'utilisation d'un serveur d'application par rapport à Axis dépend du contexte de l'établissement.

3.6.5. COMPARATIF ENTRE SSL ET WSS

Le protocle SSL et la norme WSS peuvent tous les deux remplir la fonction de confidentialité et d'intégrité. Mais il existe des différences dans la facilité d'utilisation, dans la maturité, etc. Ce comparatif ne se veut pas exhaustif.

- Le protocole SSL est maitrisé, omniprésent et est facilement paramétrable alors que WS-Security est actuellement moins utilisé (compétences moins répandues) et requiert une mise en œuvre plus complexe.
- SSL et WS-Security s'insèrent aisément dans une architecture de web services, même si le protocole SSL s'avère plus facile à mettre en œuvre en raison de l'observation précédente.
- SSL fournit la sécurité pour la totalité d'une connexion. Il assure l'intégralité du message, qu'il soit sensible ou non. WS-Security sécurise chaque message un à un. Il est également possible avec WS-Security de sécuriser certaines parties du message en utilisant différentes clés ou même différents algorithmes. Ceci permet de rendre accessible certaines parties du message à différentes entités sans en exposer la totalité.
- SSL est étroitement lié à certains protocoles comme http, ftp, pop ou imap. SSL ne peut être utilisé si le protocole de transport des requêtes est autre que ceux sus cités. A l'heure actuelle, ce n'est pas le cas pour la grande majorité des requêtes. Cependant, il existe actuellement des exemples de SOA utilisant UDP et SMTP comme protocole de transport. WS-Security quant à lui est indépendant du protocole sous-jacent et donc globalement plus interopérable.
- Enfin, SSL offre une solution point à point (Le message est alors décrypté et ré-encrypté à chaque point intermédiaire) alors que WS-Security permet une sécurisation de bout en bout (le message passe tous les points intermédiaires sans y être décrypté).

3.6.5.1. Synthèse

SSL permet de chiffrer le flux entier, c'est tout ou rien. Le contenu doit être à chaque fois déchiffré pour effectuer un routage seulement dans le cas où il existe plusieurs points intermédiaires entre les deux entités en présence. Le chiffrage avec SSL survit seulement pour la durée de vie d'une connexion, on peut avoir besoin de chiffrer à nouveau pour la connexion suivante.

WS-Security pour sa part, peut chiffrer le contenu entier, certains éléments sélectionnés, le contenu d'éléments, ou une donnée arbitraire. Il peut utiliser des clés diverses pour que diverses parties puissent lire différentes partie du message indépendamment. Une partie du contenu peut être laissée lisible pour permettre le routage, sans effet sur le contenu chiffré.

A la vue de ce comparatif et au regard des usages actuels et prévus des établissements, nous avons concentré notre étude sur SSL plutôt que WS-Security. Ce choix a été fait car SSL suffit largement à couvrir notre besoin en terme d'intégrité et de confidentialité des données échangées. Malgré tout, l'étude n'exclut pas l'utilisation de WS-Security dans des contextes très particuliers.

3.7. STRATEGIE 7 : UTILISATION D'UNE PASSERELLE XML

Couches concernées : Message SOAP

Thèmes adressés : Confidentialité, Intégrité, Identification, Authentification, Autorisation

Certains éditeurs proposent des solutions de sécurisation des web services via des appliances matérielles ou logicielles. Généralement, ces dispositifs proposent une richesse fonctionnelle importante qui va au-delà des thèmes de sécurité étudiés dans ce document. (Répartiteur de charge, Filtrage de contenu, vérification de la structure de l'XML, reprise sur incident, SLA…) De la même façon, SOAP n'est pas le seul type de protocole de messages pris en compte car ces appliances sont souvent orientées XML au sens large. Cette richesse fonctionnelle a un prix qui est souvent élevé.

Nous présentons quelques éditeurs incontournables du marché. Cette analyse n'est en aucun cas exhaustive et ne revêt pas un caractère d'analyse comparative des différentes solutions.

3.7.1. LES EDITEURS DE SOLUTIONS MATERIELLES DU MARCHE

3.7.1.1. IBM

IBM propose une solution d'appliance sécurité Datapower XS40. Il s'agit d'un boîtier placé en coupure qui permet les fonctionnalités suivantes :

- Firewall SOAP/XML,
- Chiffrement et signature des flux XML (implémente WS Security),
- Contrôle d'accès au web services,
- Analyse de contenu,
- Validation des données (détection d'attaques applicatives,…),
- Routage en fonction du contenu.

Cette passerelle de sécurité XML est à la fois robuste et évolutive au regard des différents standards dans le monde XML : il est en effet possible de modifier les schémas spécifiés par défaut. Elle bénéficie naturellement du niveau de performance requis pour une appliance.

XS40 fait partie des produits phare dans le domaine de la sécurisation des flux XML. Ce dernier a par ailleurs reçu la certification de la part du département Américain de la défense pour une utilisation de XS40 dans les administrations.

Coupe-feu : validation de schémas, inspection de contenu et surveillance des attaques les plus courantes. Gestion des identités, moteur de règles, chiffrement et signature. Standards gérés : SAML, XACML, WSSecurity, LDAP, serveurs SSO, HTTPS, SSL, TLS – XML Encryption, XML Digital Signature.

3.7.1.2. VORDEL

VORDEL propose différentes solutions autour du XML dont une passerelle XML permettant de traiter notamment la sécurité des flux SOAP. Ce produit se nomme VORDEL SECURE.

Il offre les fonctionnalités suivantes :

- Inspection des messages Soap,
- Détection des attaques les plus courantes (DoS, Overflow, injection SQL),
- Validation de schémas,
- Chiffrement et signature électronique,
- Gestion des accès et des identités,
- Pont vers les outils les plus fréquents.

Standards gérés : LDAP, SAML, WS-Security, WSTrust, XML-Signature.

3.7.1.3. Computer Associates

Le produit eTrust Transaction Minder propose :

- Gestion centralisée des accès et des droits via un moteur de règles.
- Gestion du SSO. Fédération d'identités. Standards gérés : XML Digital
- Signature, XML Encryption, LDAP, SAML, WS-Security.

3.7.1.4. REACTIVITY

Le produit REACTIVITY Security Gateway est une appliance de sécurité permettant :

- Le chiffrement,
- La signature électronique,
- La validation de schémas,
- Le filtrage actif du contenu et reconnaissance des principales attaques : DoS, spoofing, etc. Moteur de règles,
- La gestion des principaux annuaires et PKI. Standards gérés : LDAP, HTTPS, SSL, TLS – XML Encryption, XML Digital Signature, WS-Security, SAML.

3.7.2. LES EDITEURS DE SOLUTIONS LOGICIELLES DU MARCHE

3.7.2.1. OWSM

Le produit Oracle Web-Service Manager fait partie de la suite Oracle SOA et permet un support de :

• Message digests: MD5, SHA-1

Algorithme de chiffrement : AES-128, AES-256, 3-DES

• Message structure: XML / SOAP / WS-Security1.0

• Mode d'authentification: Login/mdp, X.509, SAML

• Intégrité du message: XML Signature,

• Confidentialité du message: XML Encryption

3.7.2.2. Synapse

Il existe aussi des solutions open-source qui peuvent être utilisées pour tenir un rôle de passerelle XML. L'une de ces solutions est synapse, projet Apache et sa surcouche WSO2 apportant essentiellement une Interface Homme Machine à la solution. Dans le détail ce produit est un Entreprise Service Bus.

D'un point de vue technique, le rôle d'un ESB se résume à la connexion et à la médiation entre les services et applications du Système d'Informations. C'est un intermédiaire qui permet à des applications hétérogènes de communiquer au travers de protocoles standard. A ce titre, ses principales responsabilités sont les suivantes :

- La réconciliation des mondes hétérogènes, à l'aide de standards d'interopérabilité ou de connecteurs spécialisés – c'est le rôle classique d'un middleware d'intégration.

- Découpler consommateurs et fournisseurs de services : le consommateur ne connaît que l'ESB, mais ne connaît ni les formats, ni les protocoles d'échanges spécifiques utilisés par le fournisseur du service.
- Agréger des services de niveau N afin de construire des services de niveau N+1. Si l'agrégation est complexe, ou nécessite des structures de contrôle du flux d'exécution, un moteur d'orchestration reposant par exemple sur le langage BPEL(Business Process Execution Language), est mis à contribution.
- Tracer les messages qui transitent. Devenant une zone de passage incontournable, l'ESB joue un rôle fondamental dans la traçabilité et le monitoring des traitements.

  La mise en œuvre de cette solution est présentée dans le document [SYNAPSE]

4.SYNTHESE DE L'ETUDE

4.1. POSITIONNEMENT DES STRATEGIES

Les 7 stratégies peuvent être positionnées en fonction de la couche qu'elles sécurisent ainsi que des fonctionnalités de sécurité qu'elles couvrent.

Figure 2 : Tableau de synthèse des 7 stratégies

4.2. STRATEGIES SELECTIONNEES

L'étude a fait ressortir que les stratégies suivantes vont être, pour l'instant, mises en avant :

- Filtrage IP,
- Tunnel SSL
- SPRING-Security
- Passerelle synapse

Les autres solutions ne sont pas à proscrire. Au contraire, dans un cadre donné elles peuvent être la meilleure solution.

4.3. SCENARII D'UTILISATION

Afin de couvrir un ou plusieurs thèmes, il est possible de combiner les stratégies unitaires. Nous avons identifié, à titre d'exemples, plusieurs scénarii pouvant être envisagés à partir des stratégies unitaires sélectionnées.

4.3.1. SCENARIO 1 : NE RIEN FAIRE

Figure 3 : Schéma du Scénario 1

- Confidentialité : Aucun
- Intégrité : Aucun
- Identification : Aucun
- Authentification : Aucun
- Autorisation : Aucun

4.3.2. SCENARIO 2 : CHIFFREMENT SSL SIMPLE

Figure 4 : Schéma du Scénario 2

- Confidentialité : Chiffrement SSL
- Intégrité : Chiffrement SSL
- Identification : Aucun
- Authentification : aucun
- Autorisation : Aucun

4.3.3. SCENARIO 3 : FILTRAGE IP + SPRING-SECURITY

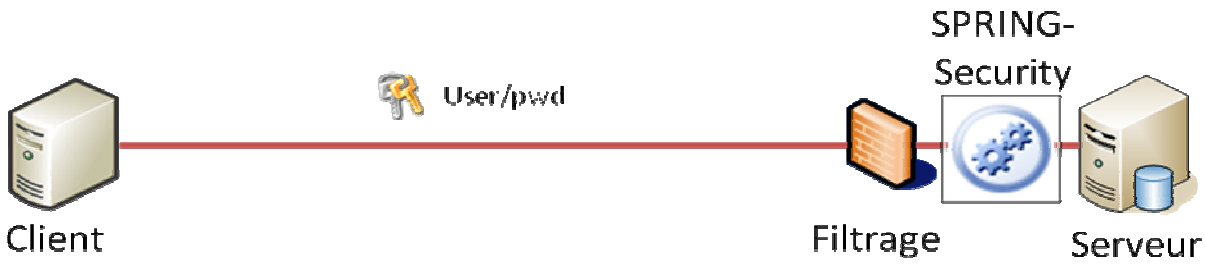

Figure 5 : Schéma du Scénario 3

- Confidentialité : Aucun
- Intégrité : Aucun
- Identification : Filtrage IP + Spring-Security
- Authentification : Filtrage IP + Spring-Security
- Autorisation : Spring-Security

4.3.4. SCÉNARIO 4 : CHIFFREMENT SSL SIMPLE + SPRING-SECURITY

Figure 6 : Schéma du Scénario 4

- Confidentialité : Chiffrement SSL
- Intégrité : Chiffrement SSL
- Identification : Spring-Security
- Authentification : Spring-Security
- Autorisation : Spring-Security

4.3.5. SCÉNARIO 5 : CHIFFREMENT SSL SIMPLE + SPRING-SECURITY + FILTRAGE IP

Figure 7 : Schéma du Scénario 5

- Confidentialité : Chiffrement SSL
- Intégrité : Chiffrement SSL
- Identification : Filtrage IP + Spring-Security
- Authentification : Filtrage IP + Spring-Security
- Autorisation : Spring-Security

4.3.6. SCENARIO 6 : CHIFFREMENT SSL SIMPLE + ESB + FILTRAGE IP

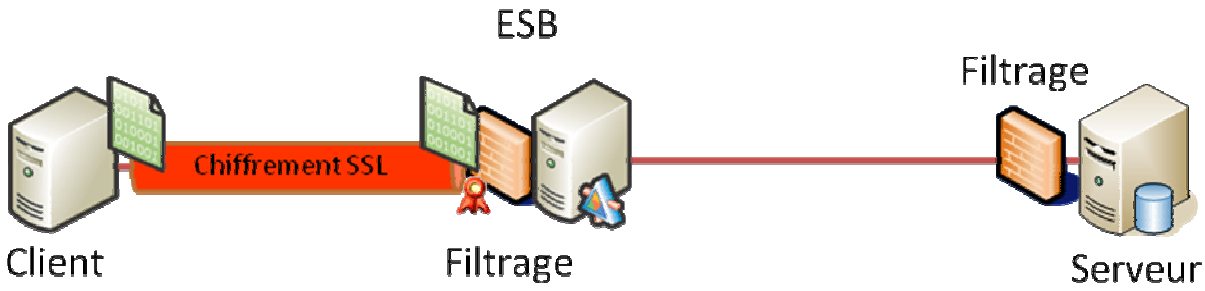

Figure 8 : Schéma du Scénario 6

- Confidentialité : Chiffrement SSL (uniquement entre le client et l'ESB)
- Intégrité : Chiffrement SSL (uniquement entre le client et l'ESB)
- Identification : Filtrage IP
- Authentification : Filtrage IP
- Autorisation : Aucun

4.3.7. SCÉNARIO 7 : CHIFFREMENT SSL SIMPLE + ESB + FILTRAGE IP + SPRING-SECURITY

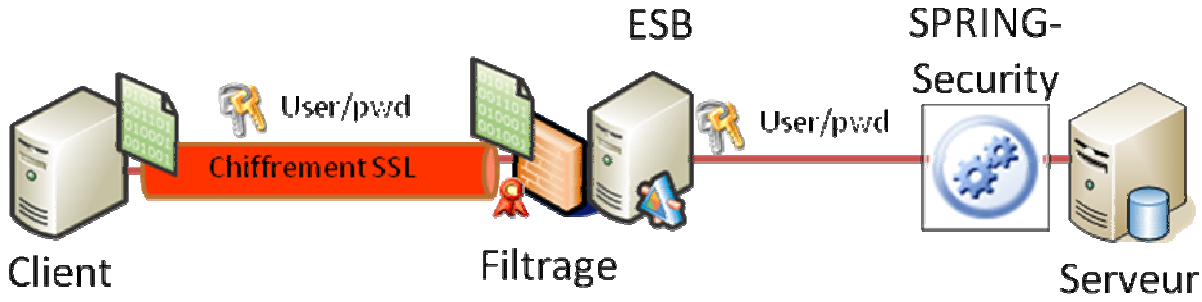

Figure 9 : Schéma du Scénario 7

- Confidentialité : Chiffrement SSL (uniquement entre le client et l'ESB)
- Intégrité : Chiffrement SSL (uniquement entre le client et l'ESB)
- Identification : Filtrage IP (sur l'ESB) + Spring-Security
- Authentification : Filtrage IP (sur l'ESB) + Spring-Security

- Autorisation : Spring-Security

4.3.8. SCÉNARIO 8 : CHIFFREMENT SSL SIMPLE + ESB + FILTRAGE IP + SPRING-SECURITY

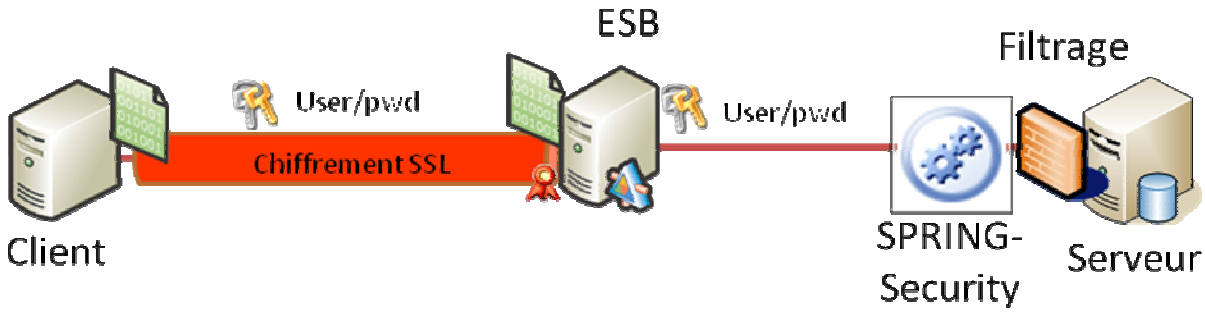

Figure 10 : Schéma du Scénario 8

- Confidentialité : Chiffrement SSL (uniquement entre le client et l'ESB)
- Intégrité : Chiffrement SSL (uniquement entre le client et l'ESB)
- Identification : Filtrage IP (sur le serveur de WS) + Spring-Security
- Authentification : Filtrage IP (sur le serveur de WS) + Spring-Security
- Autorisation : Spring-Security

4.3.9. SCENARIO 9 : DOUBLE CHIFFREMENT SSL SIMPLE + ESB + FILTRAGE IP + SPRING-SECURITY

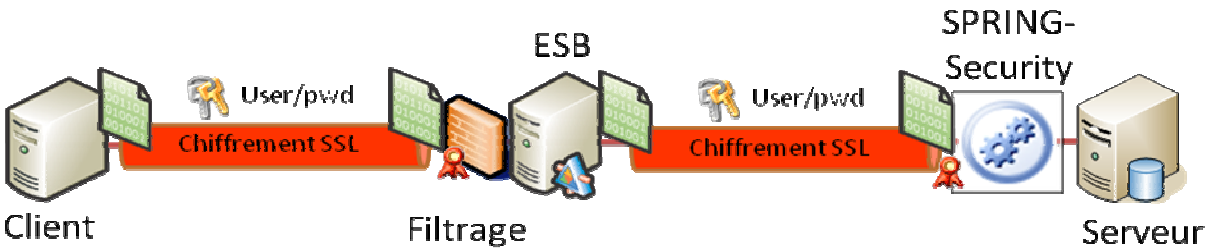

Figure 11 : Schéma du Scénario 9

- Confidentialité : Chiffrement SSL entre tous les composants
- Intégrité : Chiffrement SSL entre tous les composants
- Identification : Filtrage IP + Spring-Security
- Authentification : Filtrage IP + Spring-Security
- Autorisation : Spring-Security

4.4. CONCLUSION

Cette étude a permis de mettre en avant différentes solutions de sécurisation des Web Services. L'approche retenue a été de choisir des solutions unitaires permettant de sécuriser des aspects différents des Web Services. Une utilisation modulaire de plusieurs de ces solutions permet d'avoir un niveau de sécurité convenable de bout en bout.
Un prototype a été réalisé et présenté au groupe de travail afin de valider la faisabilité de ces recommandations. [PROTOTYPE]. Il a permis aussi de vérifier la possibilité de paramétrer l'utilisation ou non des mesures de sécurité de façon simple.
Ces recommandations ne se veulent pas contraignantes pour les établissements mais elles doivent être prises comme un catalogue de solutions dans lequel il est possible de puiser afin de réduire les risques détectés.

Afin de permettre aux établissements d'implémenter l'ensemble des stratégies unitaires autour des Web Services Apogée et Harpège, l'Agence programme les modifications à apporter à ces Web Services pour qu'ils puissent implémenter, de façon optionnelle, les stratégies de chiffrement SSL et de Spring-Security.