УДК 007.681

Ряба О.І., Черниш Л.Г., Гончар Г.Ю.

СТВОРЕННЯ КОМПЛЕКСНОГО ЗАХИСТУ WEB-СЕРВІСІВ SOAP НА БАЗІ СПЕЦИФІКАЦІЇ WS-SECURITY

В статті запропоновано та розглянуто механізми та технології, які в поєднанні дозволяють створити комплексний захист для web-сервісів, що використовують протокол SOAP. Розглянуто основні загрози web-сервісів та механізми їх нейтралізації. Розглянуто рівні, де можливо реалізувати захист, та створено схему на базі специфікації WS-Security, що дозволяє сворити комплексний захист.

Ключові слова: web-сервіс, SOAP, WS-Secutity, SSL, HTTP-аутентифікація.

Вступ. Безпека - одна з основних вимог багатьох видів корпоративних сервісів. Це також область ризику при спробах створення своїх власних сервісів, оскільки, навіть незначна і непомітна помилка може привести до серйозних вразливостей. Ці характеристики роблять обов'язковою стандартизацію управління безпекою, яка дозволяє багатьом експертам внести свій вклад в стандарт і уникнути будь-яких індивідуальних упущень.

На сьогоднішній день SOAP (Simple Object Access Protocol) є найпоширенішим протоколом обміну структурованими повідомленнями в розподілених обчислювальних системах, що базується на форматі XML. SOAP має проблеми безпеки, які включають в себе наступне. SOAP не виконує аутентифікацію між кінцевими точками SOAP або між посередниками, тому немає можливості перевірити походження SOAP повідомлення. SOAP не передбачає механізму для забезпечення цілісності даних і конфіденційності, як під час зберігання, так і під час транспортування. SOAP не передбачає механізму для виявлення перенаправлення повідомлення SOAP.

Аналіз останніх досліджень та постановка проблеми. На даний момент існує велика кількість технологій, стандартів, алгоритмів для захисту веб-сервісів. Серед них WS-Security, SSL/TLS, WS-Security Tokens, SSL/TLS X.509 Certificates, XACML, XrML, RBAC, ABAC, EPAL, XACML, UDDI, ebXML, SWSA, OWL-S, ebXML, WS-Trust, XKMS, X.509, SAML, WS-Trust, WS-Federation, IDFF, WS-Policy, WS-SecurityPolicy, WS-ReliableMessaging, WS-Reliability [1]. То ж перед розробником та компаніями, що мають на меті захистити свої веб-сервіси, постає питання, які саме технології обрати щоб створити комплексний захист і при цьому уникнути дублювання функціональності та зайвих витрат на розробку. То ж створення комплексного захисту веб-сервісів SOAP є доцільним та актуальним.

Основна частина. Web-сервіси на базі SOAP можуть використовувати широко підтримуваний стандарт WS-Security і пов'язані з ним стандарти для задоволення потреб в галузі безпеки, дозволяючи налаштовувати захист для кожного сервісу. Рішення безпеки повинні бути забезпечені завжди з розумінням загроз, що стоять перед системою. Відповідно до WS-I [2] існують такі основні загрози веб-сервісам:

Зміна повідомлення. Зловмисник вставляє, видаляє або змінює інформацію в повідомленні.

Втрата конфіденційності. Інформація в повідомленні несанкціоновано розкривається.

Підробка повідомлень. Створюються фіктивні повідомлення, що відправляються зловмисником від імені допустимого відправника.

«Людина по середині». Третя сторона знаходиться між відправником та постачальником і передає повідомлення іншим учасникам. Два учасники не знають, що зловмисник переглядає та може змінювати всі повідомлення.

Спуфінг. Зловмисник створює і відправляє повідомлення з обліковими даними уповноваженого відправника повідомлення.

Відтворення повідомлень. Зловмисник намагається відтворити раніше відправлене повідомлення.

Відтворення частин повідомлень. Зловмисник включає частину або декілька раніше надісланих повідомлень в нове повідомлення.

Відмова в обслуговуванні. Атакуючий змушує систему витрачати ресурси непропорційно, так, що допустимі запити не можуть бути задоволені.

Важливість цих загроз може змінюватись в залежності від потреб організації та цілей. У деяких випадках повідомлення не повинні бути конфіденційними, так, що втрата конфіденційності не буде мати значення. Крім того, організації можуть надавати веб-сервіси для публічного доступу. Наприклад, веб-сервіс, який надає інформацію про поточний прогноз погоди, може не турбуватися, якщо прийде запит від фальсифікованого відправника. Незважаючи на це, важливо зрозуміти ці загрози і які технології доступні для їх усунення. Захист Web-сервісів може бути реалізовано на двох рівнях: на транспортному рівні і рівні повідомлень.

Захист на транспортному рівні. Захист на транспортному рівні представляє собою механізм захисту «точка-точка», який можна використовувати для ідентифікації і аутентифікації суб'єктів, забезпечення цілісності повідомлень і конфіденційності. HTTP, самий використовуваний в інтернеті комунікаційний протокол, в даний час також є найпопулярнішим протоколом для Web-сервісів. HTTP за своєю суттю є незахищеним протоколом, оскільки, вся інформація передається звичайним текстом між не-аутентифікованим суб'єктами по незахищеній мережі. При використанні системи захисту на транспортному рівні з'єднання між клієнтом і сервером додатків, зазвичай, захищено за допомогою протоколу Secure Sockets Layer (SSL), а клієнт і сервер аутентифікують справжність один одного і взаємодіють за допомогою зашифрованих повідомлень. У системі захисту на транспортному рівні взаємодія шифрується повністю.

*Захист на рівні повідомлень***.** Захист на рівні повідомлень являє собою підхід, коли вся інформація, що відноситься до системи захисту, інкапсулюється в SOAP-повідомлення. Захист на рівні повідомлень забезпечує захист за допомогою маркера імені користувача (UserName Token), XML-шифрування і цифрових підписів. Захист на рівні повідомлень грунтується на специфікації WS-Security. Вона часто застосовується в поєднанні із захистом на транспортному рівні. Схема, що пропонується в даній статті для захисту на рівні повідомлень зображена на рис. 1.

Таким чином, в результаті проведеного дослідження обраний набір засобів для комплексного захисту веб-служб включає:

- W3C XML Шифрування;
- W3C XML Цифровий підпис;
- Маркери WS-Security. Підтримувані типи маркерів, включають ім'я користувача / пароль, OASIS SAML Assertion, IETF сертифікат X.509, ISO Rights Expression Language, Маркери IETF Kerberos [3];
- W3C WS-Адресація;
- SSL / TLS з аутентифікацією клієнта;
- HTTP Аутентифікація.

*XML Шифрування та цифровий підпис***.** XML Encryption - специфікація, що визначається W3C як рекомендація, яка визначає, як зашифровується вміст елемента XML . XML Signature – специфікація W3C для створення цифрового підпису документів XML. Обидва: XML Signature і XML Encryption використовують елемент KeyInfo, який відноситься, як нащадок до елементів SignedInfo, EncryptedData, або EncryptedKey, і надає інформацію для одержувача про ключі, які використовуються для перевірки підпису або розшифровки зашифрованих даних. Елемент KeyInfo є необов'язковим - він може бути приєднаний до повідомлення, або доставлений через безпечний канал. Приклад використання XML Encryption:

```
<?xml version='1.0'?>
<ExampleInfo xmlns='http://example.com/info'>
   <Name>Ivan Ivanov</Name>
   <EncryptedData Type='http://www.w3.org/2001/04/xmlenc#Element'
    xmlns='http://www.w3.org/2001/04/xmlenc#'>
     <CipherData>
```

```
      <CipherValue>A37F8H9</CipherValue>
     </CipherData>
    </EncryptedData>
</ ExampleInfo >
```

*Маркер UsernameToken WS-Security***.** Завдання UsernameToken полягає в тому, щоб передавати інформацію про ім'я користувача та пароль в заголовках WS-Security. Найпростіша форма UsernameToken передає ім'я користувача та пароль як звичайний текст. Це не оптимальне рішення з точки зору безпеки (хоча немає нічого поганого у використанні цього підходу при безпечних з'єднаннях), затее, легко побачити, що саме передається, а це робить його корисною відправною точкою. Конфігурація WS-SecurityPolicy для UsernameToken, переданого як текст, може бути простою. Ця політика складається з стандартної оболонки WS-Policy (елементи з префіксом wsp) навколо затвердження WS-SecurityPolicy UsernameToken.

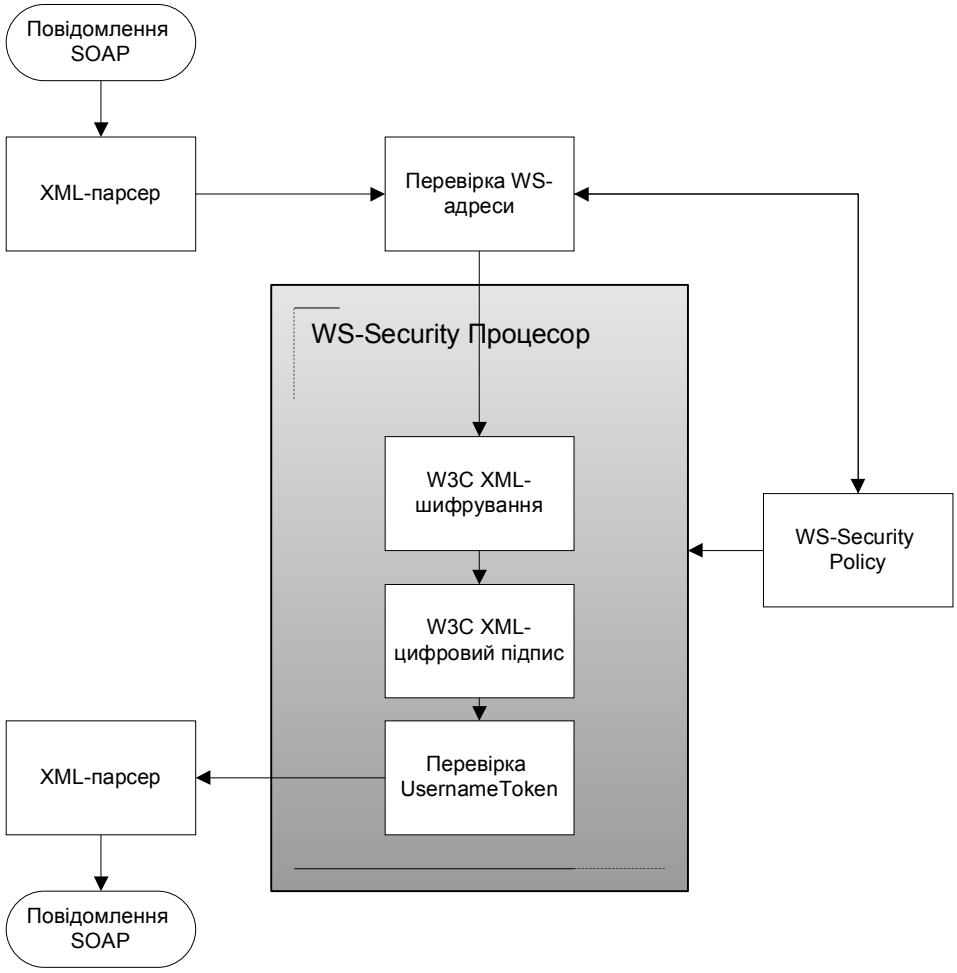

Рис.1. Схема захисту на рівні повідомлень

WS-Адресація (Web Services Addressing) - це специфікація транспортного механізму, що дозволяє WEB службам обмінюватися інформацією про адреси. По суті, вона складається з двох частин: структура, яка містить кінцеві точки служб, і набору властивостей повідомлення, що містять інформацію про адресу з конкретним повідомленням. WS-Адресація - це стандартний спосіб включення інформації про маршрутизації в SOAP заголовки. Замість того, щоб повертати інформацію про маршрутизацію з мережевого рівня повідомлення, що використовує WS-Адресацію, може містити власні метадані в стандартному SOAP заголовку. В даному випадку мережевий рівень відповідає тільки за доставку повідомлення диспетчеру, здатному читати метадані такого повідомлення. Коли це повідомлення приходить до диспетчера, той визначає URI, робота мережевого рівня на цьому закінчується.

Властивості повідомлення містять адресну інформацію, що відноситься до доставки повідомлення на Web сервіс: пункт призначення повідомлення (Message destination), кінцева точка ресурсу (Source endpoint) - кінцева точка служби, яка відправляє повідомлення, Reply endpoint - кінцева точка на яку буде відправлено відповідь, Fault endpoint - кінцева точка на яку буде надіслано повідомлення про помилку, дія (Action) - параметр, що відображає зміст повідомлення, ID повідомлення URI, взаємозв'язок з попереднім повідомленням (A Pair Of URIs).

SSL дозволяє виконувати аутентифікацію через сертифікати на стороні клієнта і на стороні сервера. Він гарантує передачу і незмінність цієї інформації, а також те, що вона прийде тільки на той сервер, для якого призначена. HTTPS Web-сервіси можна застосовувати з усіма типами клієнтів, включаючи Java EE-клієнти та автономні Java-клієнти. В даний час, HTTPS із сертифікатами сервера є найбільш широко використовуваною конфігурацією в Web. У такій конфігурації сервер повинен надати свій сертифікат клієнта, для того щоб він визначив ідентичність сервера. Клієнту не потрібно надавати свій власний сертифікат серверу, для того щоб сервер визначив ідентичність клієнта. Іншими словами, клієнт може автентифікувати сервер, а сервер не може автентифікувати клієнта. Однак можна використовувати HTTPS спільно з базовою аутентифікацією, яка дозволяє серверу теж автентифікувати клієнта.

*Базова HTTP-аутентифікація***.** Простим способом надати аутентифікаційні дані для клієнта Web-сервісу є базова HTTP-автентифікація на захищених кінцевих точках сервісу. Інформація для базової аутентифікації розташована в HTTP-заголовку, який містить SOAP-запит. Для запобігання можливості безпосереднього читання імені користувача і пароля ким-небудь ще, перед передачею вони кодуються в послідовність символів base-64.

Базова HTTP-аутентифікація відрізняється від підтримки базової аутентифікації, що надається WS-Security. Інформація для базової аутентифікації, що надається WS-Security, є *SOAP-заголовок,* тоді, як інформація для базової HTTP-аутентифікації є *HTTP-заголовок.*

Базова аутентифікація має недоліки в системі захисту: інформація для аутентифікації передається в кодуванні base-64, декодувати яку не складає труднощів [4]. Передача інформації в кодуванні base-64, майже, так само незахищена, як і відкритий текст. Рекомендується передавати дані по протоколу HTTPS.

В даний час широко застосовується комбінація базової HTTP-аутентифікації з HTTPS, оскільки, вона вирішує такі завдання системи захисту: аутентифікація, авторизація, забезпечення конфіденційності, забезпечення цілісності [5].

Таким чином обраний набір засобів захисту дозволяє створити комплексний захист для веб-сервісів. У таблиці 1 наведено перелік загроз з можливістю їх нейтралізації за допомогою обраних засобів захисту.

Нейтралізація основних загроз Таблиця 1

	Зміна повідомлення	**Втрата конфіденційності**	**Підробка повідомлень**	**«Людина по середині**	**Спуфінг**	**Відтворення повідомлень**	**Відтворення частин повідомлень**	**Відмова в обслуговуванні**
XML шифрування		+		+	+	+	+	
XML цифровий підпис	+		+		+	+	+	
Маркери WS-Security			+		+			
WS-Адресація						+	+	
SSL/TLS	+	+	+	+	+	+		
HTTP-аутентифікація			+		+			

Висновки. Таким чином, обраний набір засобів захисту дозволяє нейтралізувати основні загрози WS-I для веб-сервісів. Також, обрані засоби захисту функціонують як на транспортному рівні, так і на рівні повідомлень протоколу веб-сервісу SOAP. XML-шифрування дозволяє засекретити вміст повідомлень при роботі через будь-яке з'єднання, навіть, коли в обробці беруть участь ненадійні посередники. Використання підписів XML гарантує, що повідомлення дійсно відправлені тим, хто значиться відправником, і їх вміст не було змінено при передачі. Ці потужні засоби безпеки мають важливе значення для багатьох видів обміну бізнес-даними, але вони дорого коштують з точки зору накладних витрат по додатковій обробці. WS-Security робить істотний вплив на продуктивність, і в деяких випадках кращим рішенням буває просте захищене з'єднання "точка-точка" SSL. Але для багатьох типів програм SSL недостатньо. У цих випадках потрібно WS-Security (або його продовження WS-SecureConversation), і втрата продуктивності стає просто необхідною жертвою. WS-Addressing усуває всяку залежність від транспортних заголовків або передачі специфічних параметрів при отриманні доступу до Web-сервісів. Це особливо важливо при використанні архітектури SOA, де не потрібно, щоб сервіси передавалися по HTTP, хоча більшість розробок, на сьогоднішній момент, засновані на цьому повсюдно, використовуються в протоколі.

ЛІТЕРАТУРА

1. OASIS Standard, working Draft 17. Web Services Security: SOAP Message Security. - http://xml.coverpages.org/WSS-SOAPMessageSecurity-17-082703-merged.pdf
2. OASIS Standard Specification. Web Services Security: SOAP Message Security 1.1 (WS-Security 2004). - http://docs.oasis-open.org/wss/v1.1/
3. OASIS Standard 200401. Web Services Security UsernameToken Profile 1.0. - http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.
4. Chomsiri T. HTTPS Hacking Protection / Chomsiri T. // Advanced Information Networking and Applications Workshops. - 2007. - №21 - – С. 77–80.
5. Белозеров Е.В. Особенности проектировани и построения защищенных web-систем / Белозеров Е.В., Ладик Д.А.// Вісник Східноукраїнського національного університету ім. В. Даля. - 2007. - №1 - с 85-88.

Надійшла: 15.12.2011

Рецензент: д.т.н., проф. Юдін О.К.