УДК 004.05

ЗАЩИТА ИНФОРМАЦИИ, ПЕРЕДАВАЕМОЙ ПО ПРОТОКОЛУ SOAP. БЕЗОПАСНОСТЬ WEB-СЕРВИСОВ (WSSECURITY)

Ю.С. КУВЕЦКИЙ
(Представлено: канд. техн. наук И.Б. БУРАЧЕНОК)

Рассмотрены особенности протокола SOAP, проблемы с которыми можно столкнуться при его использовании и методы их решения. Проведен анализ основных современных способов обеспечения безопасности процесса общения с web-сервисами по сети Internetu, их реализации в стандарте WSSecurity. Сделаны выводы о необходимых мерах для обеспечения полной безопасности при использовании web-сервисов.

Во все времена необходимым условием развития человечества являлся процесс автоматизации различного вида его деятельности. Сегодня развитие передовых информационных технологий (ИТ) открывает новые возможности для создания современных автоматизированных систем (АС) во всех сферах народного хозяйства. Безусловно, возникают и проблемы, связанные с необходимостью повышения уровня безопасности таких систем. Причем выбор методов и средств защиты определяется не только важностью обрабатываемой информации, но и составом АС, ее структурой, способами обработки информации, а также количественным и качественным составом пользователей и обслуживающего персонала. Рассмотрим систему, основными задачами которой являются удаленная компиляция и выполнение программного кода (ПК) на различных web-сервисах. Среди разработчиков современных программных приложений наиболее популярна сервисная архитектура – такие приложения гораздо проще поддерживать, легче решается задача масштабирования, но появляется проблема безопасности, так как происходит передача данных по сети. В данной статье мы рассмотрим вариант построения системы, основываясь на протоколе передачи данных SOAP и стандарте обеспечения безопасности WSSecurity.

Во-первых, рассмотрим, что такое SOAP. Протокол SOAP (SimpleObjectAccessProtocol) используется для обмена произвольными структурированными сообщениями в формате XML по распределенной вычислительной среде. Он может использоваться с любым протоколом прикладного уровня, однако чаще всего SOAP используется поверх HTTP (HyperTextTransferProtocol). Его сообщение в формате XML (eXtensibleMarkupLanguage) представлено на рисунке 1 и представляет из себя контейнер (envelope) содержащий заголовок (header) и тело (body), тело в свою очередь может содержать элемент с ошибками и статусом запроса (fault). Microsoft позволяет использовать протокол SOAP при помощи WCF (WindowsCommunicationFoundation) [1].

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header>
        ...
    </soap:Header>

    <soap:Body>
        ...
    </soap:Body>
</soap:Envelope>
```

Рисунок 1. – Структура XMLсообщения, передаваемого по протоколу SOAP

Далее рассмотрим три стандарта безопасности применимых к XML: аутентификация, целостность данных, конфиденциальность данных.

Аутентификация гарантирует, что отправитель и получатель являются теми, кем они себя объявляют, доказывает подлинность сторон. Это может реализовываться различными способами. Простой вариант – предоставить идентификатор пользователя и его пароль. Более сложный – это использование сертификата, который содержит все необходимые идентификационные учетные данные и ассоциированную с ними пару из закрытого и открытого ключей.

Для обеспечения целостности информации, которой обмениваются стороны и гарантии того, что содержимое сообщений не будет изменено или повреждено во время их передачи по сети, данные подписывают цифровой подписью (ЦП) с использованием ключей безопасности. Часто для подписания ЦП SOAP-тела запроса используется закрытый ключ сертификата X.509 отправителя. Также можно подписывать SOAP-заголовки запросов, чтобы гарантировать целостность дополнительно передаваемой в транзакции информации.

Третьим требованием для обеспечения безопасности системы является конфиденциальность. Чтобы сделать обмен информацией в запросах и ответах web-сервисов нечитаемым для посторонних, используется технология шифрования. Цель – гарантировать, что любая попытка обратиться к данным при передаче, в памяти или после сохранения потребует соответствующих алгоритмов и ключей безопасности для дешифрования данных.

На данный момент все эти меры безопасности достаточно просто реализуются, а варианты реализации могут зависеть от способа транспортировки сообщений или быть специфичными для протокола SOAP. Каждый из них в отдельности может обеспечить достаточный уровень безопасности, но в таких случаях система всегда подвергается риску. Без аутентификации все будут иметь доступ к сервису. Без ЦП существует вероятность подмены или повреждения информации. При игнорировании правила конфиденциальности, любой злоумышленник сможет прослеживать всю информацию, проходящую по сети. Если учесть все перечисленные требования, то любую систему можно считать защищенной, при этом не так важны способы их реализации. Конечно, существуют системы, в которых можно пренебречь этими правилами, например, если никак не нужно ограничивать доступ к сервису, то можно опустить аутентификацию. Если масштаб системы ограничивается отправкой запросов между пользователем и одной точкой (сервисом), то достаточным будет использования HTTPS (HyperTextTransferProtocolSecure) и необходимость реализации шифрования отпадает. То же самое и с ЦП данных, в ней нет необходимости, если пересылаемые сообщения не несут какой-либо ценности и не могут навредить системе.

Обеспечение целостности, конфиденциальности и аутентичности сообщения и его отправителя при одновременном сохранении открытости для расширений – основные задачи стандарта WSSecurity [2]. Он не определяет никаких новых технологий, а опирается на уже существующие стандарты, к примеру, XMLEncryption, XMLSignature, сертификаты X.509 или различные криптографические алгоритмы. Благодаря тому, что его базовая концепция основывается на механизмах сообщений, становится возможным обеспечить безопасность из конца в конец (End-to-EndSecurity), например, посредством протокола SSL (SecureSocketLayer), вместо защиты ориентированной на транспорт. Основными элементами стандарта являются следующие базовые механизмы: токены безопасности (ТБ), шифрование, ЦП и отметки о времени.

Токены безопасности применяются при аутентификации, их задача нести подтверждения идентификации (credentials), без которых сама аутентификация невозможна. Чаще всего в роли credentials выступают идентификационный номер (UserID) и соответствующий пароль.

На тему шифрования можно привести много информации, но принципиально различают два механизма шифрования: симметричное и асимметричное. В первом случае для шифрования и дешифровки используется общий ключ, всегда доступный обеим сторонам, этот способ быстрее. Во втором для шифрования и дешифровки применяются разные ключи: личный ключ остается у владельца, а общий ключ распространяется свободно, этот способ более надежный. Оба подхода часто объединяют следующим образом: клиент генерирует симметричный ключ, шифрует им данные любого размера, после чего сам ключ шифруется при помощи асимметричного алгоритма и вкладывается в сообщение.

Цифровые подписи применяются для подтверждения целостности сообщений, благодаря им возможно распознать неправомерные модификации, такие как добавление, изменение или удаление данных.

В WS Security этот подход опирается на стандарт XML DigitalSignature. Принцип ЦП основан на создании контрольных сумм с помощью специальных алгоритмов (дайджет), после чего результаты присоединяются к сообщению и передаются в частично зашифрованном виде.

Стоит также рассмотреть технологию отметок о времени. Дело в том, что в рамках SOA (SimpleObjectAccess) сервисы должны производить определенное действие и таким образом поддерживать взаимодействие без учета состояния (Stateless). Данная особенность позволяет злоумышленникам производить атаки сброса (Replay), когда атакующий повторно отправляет либо сообщения целиком, либо отдельные их части. Для защиты от таких атак необходимо, чтобы каждое сообщение имело свой уникальный идентификатор (MessageID), которые сервис хранит и учитывает при последующих сообщениях.

В результате проведенного исследования можно сделать **вывод:** протокол SOAP достаточно гибок и удобен в использовании; он позволяет абстрагировать общение с различными web-сервисами за счет использования в них общего интерфейса. Благодаря стандарту WSSecurity можно обеспечить достаточный уровень защиты системы от угроз. Для организации общения клиента напрямую с web-сервисами было принято решение ограничиться использованием протокола SSL с достаточно высоким уровнем безопасности для передачи данных по сети напрямую от клиента к сервису.

ЛИТЕРАТУРА

1. Microsoft. Общие сведения о безопасности [Электронный ресурс] / MDN. – Режим доступа: https: //msdn.microsoft.com/. – Дата доступа: 10.09.2017.
2. IBM Knowledge Center. WS-Security [Электронный ресурс] / IBM. – Режим доступа: https://www. ibm.com/support/knowledgecenter/. – Дата доступа: 10.09.2017.