Segurança em Web Services

Jandson Almeida da Silva
Av. Gonçalo Rollemberg Leite, 1913
Edf. Jackson Figueiredo apto 204
(79) 9971-6928
falefacil@ig.com.br

ABSTRACT

This document explains the forms to guarantee security in Web Services since the transport layer until the application layer.

KEYWORDS

web service, security, XML, encryption, digital signature.

RESUMO

Este documento explica as formas de garantir segurança em Web Services desde a camada de transporte até a camada de aplicação.

1. INTRODUÇÃO

Os Web Services surgiram com o objetivo de implementar a integração entre sistemas heterogêneos de forma simples e padronizada utilizando a malha da Internet.

Uma das regras que nortearam a criação do padrão foi "não invente nada de novo". Basicamente um Web Service funciona como uma página Web, com a diferença que ao invés de HTML, utiliza-se XML. Desta forma os dados podem ser descritos e o pacote da mensagem pode ser manipulado com grande facilidade tanto por quem envia, quanto por quem recebe [2].

Um exemplo de uso de Web Service [1] é o envio de requisição a um serviço publicado em uma URL via http, usando o protocolo SOAP (*Simple Object Access Protocol*). Ele recebe a mensagem, processa-a e retorna uma resposta usando o protocolo SOAP como mostra a figura 1.

A definição da interface de um Web Service é dada através de seu WSDL (*Web Service Definition Language*).

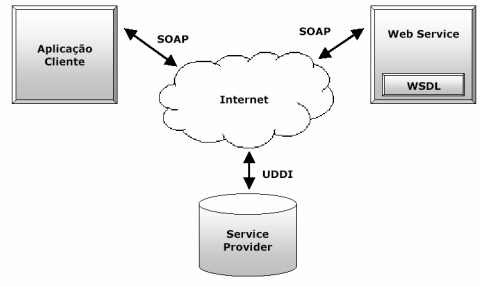

Figura 1 – Diagrama conceitual de um Web Service

Este artigo se propõe a mostrar questões relacionadas à segurança dos dados usando a tecnologia dos Web Services.

A preocupação com a segurança desperta uma atenção especial quando pensamos em ambientes no mundo real. Pela grande diversidade de sistemas que compõem atualmente os vários ambientes de TI, é necessário que haja uma abordagem modular à questão de segurança em Web Services [2]. Na medida em que cresce o uso de Web Services entre empresas que adotam diferentes modelos de segurança em sua colaboração, as especificações de segurança e confiabilidade propostas devem oferecer um modelo flexível, que permita a essas empresas se interconectarem de forma confiável. Como sabemos, Web Services utilizam o protocolo SOAP em sua comunicação, o qual é baseado em XML, ou seja, um arquivo texto perfeitamente legível. Para tornar um Web Service seguro é necessário encriptar a comunicação e existem duas maneiras de se fazer isso: garantir a segurança ao nível de transporte e ao nível de XML [2].

2. SEGURANÇA AO NÍVEL DE TRANSPORTE

2.1 SSL

Segurança ao nível de transporte significa proteger o protocolo de rede que o Web Service utiliza para se comunicar [2]. O *Secure Sockets Layer* (SSL) é um protocolo padrão para encriptar comunicações sobre TCP/IP. Neste modelo, um cliente abre um *socket* seguro para um Web Service e então o utiliza para trocar mensagens SOAP via HTTPS.

A implementação de SSL garante segurança encriptando todo o tráfego de rede sobre o socket. Utilizando-se SSL pode-se também autenticar um Web Service para um cliente utilizando-se um Certificado Digital através da Infra-estrutura de Chaves Públicas (PKI).

2.2 PKI

A Infra-estrutura de Chaves Públicas (PKI) garante a segurança de comércio eletrônico e de comunicações na Internet através dos seguintes elementos [2]:

Autenticação - quando uma organização abre suas portas para a Internet, é muito importante verificar a identidade dos usuários e das máquinas. Mecanismos de autenticação robustos, como o PKI, verificam a identidade sem permitir a transmissão ou o armazenamento de senhas reutilizáveis. Eles certificam-se de que as pessoas ou máquinas sejam as entidades que dizem ser. Esta tarefa é geralmente envolve um terceiro no processo, que pode ser uma empresa de autenticação confiável ou um serviço de certificação convencional. Uma vez usando o PKI de forma adequada, torna-se virtualmente impossível que uma pessoa não autorizada acesse o sistema.

Encriptação - Encriptação e algoritmos de integridade são utilizados para proteger comunicações e garantir a privacidade durante o envio de dados de um computador a outro.

Não repúdio - Não repúdio significa que os remetentes de e-mails ou transações assinadas digitalmente não podem alegar que não a transmitiram. Assinaturas digitais utilizando PKI podem garantir com segurança que a pessoa que assina a transmissão eletrônica é realmente quem deveria ser, uma vez que ninguém mais pode criar aquela assinatura digital. Uma assinatura digital PKI prova que um determinado usuário realizou certas operações no sistema. As credenciais seguras consistem em Chaves Públicas e Privadas e devem ser utilizadas quando as entidades desejem se comunicar de forma segura. Esta forma de criptografia usa uma chave privada secreta e uma chave pública relacionada matematicamente à chave privada. Somente a chave pública pode ser utilizada para encriptar a informação, e somente a chave privada correspondente pode ser usada para decriptá-la. Somente o proprietário do par de chaves conhece a chave privada, enquanto a chave pública pode ser amplamente distribuída e mantém-se associada ao proprietário. A mensagem encriptada com a chave pública pode ser somente decriptada pelo proprietário, o qual conhece a chave privada associada. Estas chaves também são utilizadas em assinaturas digitais para prevenir que um indivíduo se faça passar por outro ou prevenir repúdio de mensagens válidas.

Certificados Digitais - Certificados são identidades digitais atribuídas por uma terceira entidade que identifica os usuários e as máquinas.

Entidade Certificadora - a Entidade Certificadora é um terceiro agente envolvido que atua como um provedor independente e confiável de certificados digitais. O uso de um par de chaves criptografadas para criar um canal de comunicação seguro garante a privacidade da mensagem e proporciona a possibilidade de validação do remetente. A ampla distribuição da chave pública não afeta a sua segurança, pois a chave privada nunca é compartilhada. A chave pública de um indivíduo é publicada por uma Entidade Certificadora em um certificado atribuído ao usuário. As entidades que desejarem trocar informações seguras podem codificar a informação com a chave pública da entidade destinatária. Uma entidade que receba uma comunicação codificada por este método pode usar sua própria chave privada para decodificar a mensagem. Se a mensagem original não for decodificada usando a chave adequada o resultado da decodificação será ilegível.

2.3 IPSec

Descrevendo ainda a segurança ao nível de transporte, o projeto IPSec representa um esforço desenvolvido pelo *Working Group* IPSec da IETF (*Internet Engineering Task Force*) para desenvolver uma arquitetura de segurança para o protocolo IP, integrando mecanismos de autenticação, gestão e distribuição de chaves que podem ser usados com qualquer das versões do protocolo IP [2].

Através dos seus componentes, a IPSec usa este conceito para permitir a implementação de redes virtuais privadas e seguras (VPN) através de redes públicas tais como a Internet.

O IPSec integra gestão manual de chaves. A gestão é de responsabilidade de protocolos criados para este fim. Os componentes da IPSec são, o Cabeçalho de Autenticação (AH), o Cabeçalho de Encapsulamento de Dados de Segurança (ESP) e os Mecanismos de Gestão de Chaves.

Cabeçalho de Autenticação - O cabeçalho de autenticação representa um cabeçalho de extensão do protocolo IPv6 e foi criado para validar a identidade de entidades que estão a comunicar: identifica o emissor e destino correto, podendo ser usado para verificar se o emissor que afirma ter enviado os dados é exatamente quem afirma ser e foi desenhado de modo a providenciar mecanismos de autenticação aos datagramas IP.

É usado normalmente em conjunto com o cabeçalho de encapsulamento de dados, já que, durante a transmissão, este cabeçalho por si só não fornece proteção contra ataques de análise de tráfego ou confidencialidade.

Cabeçalho de Encapsulamento de Dados de Segurança - O cabeçalho de encapsulamento de dados de segurança (ESP) é um cabeçalho de extensão pertencente ao protocolo IPv6 que fornece integridade e confidencialidade aos datagramas IP através da cifra dos dados contidos no datagrama. A utilização do ESP pode ser efetuada de dois modos:

- **Modo de Transporte (*transport-mode*)** - Este modo encripta a informação do protocolo da camada de transporte, adicionando-lhe de seguida um novo cabeçalho IP não-encriptado.
- **Modo de Túnel (*tunnel-mode*)** - Providencia proteção ao pacote IP. Para tal, após a adição dos campos ESP ao pacote IP, todo o pacote é tratado como o módulo de dados de um novo pacote IP. Um exemplo é o envio de pacotes IP através de canais virtuais criados numa rede IP pública, como a Internet.

Mecanismos de Gestão de Chaves - Além dos mecanismos de autenticação e validação da informação a IPSec necessita de um mecanismo eficiente de gestão de chaves. A gestão de chaves diz respeito à criação, eliminação e alteração das chaves. Embora o IPSec não integre um mecanismo de gestão de chaves, a IETF definiu como norma de gestão o protocolo híbrido ISAKMP/Oakley também denominado *Internet Key Exchange* (IKE), que se encontra baseado nos documentos:

- ISAKMP - *Internet Security Association and Key Management Protocol*. Protocolo que descreve uma infra-estrutura para a gestão de associações de segurança;
- Oakley - protocolo que define material de chaves para cifra, *hashing* e autenticação e é compatível com a gestão de associações de segurança ISAKMP;
- *Internet Domain Of Interpretation* - define parâmetros ISAKMP para as associações de segurança IPSec no domínio Internet;
- Resolução ISAKMP/Oakley - define o perfil do protocolo híbrido ISAKMP/Oakley, escolhido como norma de gestão de chaves criptográficas pela Internet Engineering Task Force;
- IKE - *Internet Key Exchange*.

3. SEGURANÇA AO NÍVEL DE XML

Por sua vez, segurança ao nível de XML envolve a encriptação e decriptação de documentos XML [2]. O *World Wide Web Consortium* (W3C), o qual mantém o padrão XML, tem criado grupos de trabalhos para definir padrões para segurança em XML,

incluindo assinaturas digitais, encriptação e gerenciamento de chaves para XML.

3.1 XKMS (XML *Key Management Services*)

Esta especificação estabelece um formato de documento em XML que permite que a autenticação de chaves e gerenciamento de certificados sejam feitos em aplicações Web [2]. A XKMS permite um nível mais profundo de interoperabilidade entre sistemas de PKI, ao mesmo tempo em que permite uma integração mais fácil entre as aplicações da empresa e a infra-estrutura de chaves públicas corporativa.

3.2 WS-Security

Em abril de 2002, a Microsoft Corporation, a IBM Corporation e a VeriSign Inc. uniram-se e publicaram um conjunto de novas especificações de segurança denominadas WS-Security, com o objetivo de que as empresas pudessem criar e construir aplicações de Web Services com ampla interoperabilidade [2].

As especificações do WS-Security, propostas pela IBM e Microsoft, são fundamentais para a concretização de um planejamento amplo de recursos que possam atender à crescente necessidade de oferecer suporte mais seguro para construção de Web Services. O documento "A Segurança no mundo dos Web Services", de autoria da Microsoft e da IBM, explica as novas especificações de segurança para Web Services que essas empresas pretendem desenvolver em conjunto com alguns de seus principais clientes, parceiros de mercado e entidades responsáveis pela padronização.

O WS-Security suporta, integra e unifica vários modelos, mecanismos e tecnologias de segurança em uso no mercado, permitindo que vários sistemas possam interoperar em plataformas e linguagens neutras.

As novas especificações de segurança definem um conjunto de padrões para extensões SOAP ou para cabeçalhos de mensagens, utilizados para oferecer maior integridade e confidencialidade às aplicações de Web Services. O SOAP é um protocolo baseado na linguagem XML, com acesso a diferentes plataformas ou linguagens. O WS-Security oferece os mecanismos-padrão de segurança necessários para realizar o intercâmbio seguro de mensagens certificadas em um ambiente de Web Services.

Além da divulgação das especificações WS-Security, a IBM e a Microsoft anunciaram também a publicação do projeto "A Segurança no mundo dos Web Services" (*Security in a 49 Web Services World*). Esse documento define alguns recursos adicionais de segurança de Web Services que se enquadram no modelo estabelecido pelas especificações WS-Security, entre eles:

WS-Policy, WS-Trust e WS-Privacy - O WS-Policy define como os recursos e restrições das normas de segurança poderão ser expressos; o WS-Trust irá descrever um modelo para que se obtenha um relacionamento de confiança, tanto direto, quanto por meio de agentes (incluindo terceiros e intermediários); o WS-Privacy determinará de que forma os Web Services serão adotados e implementados.

WS-Secure Conversation, WS-Federation e WS-Authorization - O WS-Secure Conversation explica como autenticar e gerenciar a troca de mensagens, incluindo especificações para o contexto de segurança desse intercâmbio e a derivação de chaves de sessão; o WS-Federation gerencia os vários tipos de relacionamento em ambientes heterogêneos associados, incluindo o suporte à identidade dessas partes associadas; a WSAuthorization define como os Web services administrarão os dados e as normas de autorização.

3.3 XML-Signature Syntax and Processing

Este padrão W3C especifica sintaxe e regras de processamento de assinaturas digitais em XML [3]. O exemplo abaixo, trecho da assinatura de um arquivo XML, demonstra o uso desta especificação:

```
[s01] <Signature Id="MyFirstSignature"
 xmlns="http://www.w3.org/2000/09/xmldsig#">
[s02]  <SignedInfo>
[s03]  <CanonicalizationMethod
 Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
[s04]  <SignatureMethod
 Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
[s05]  <Reference
 URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
[s06]   <Transforms>
[s07]    <Transform
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
[s08]   </Transforms>
[s09]   <DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
[s10]
<DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
[s11]  </Reference>
[s12]  </SignedInfo>
[s13]  <SignatureValue>MC0CFFrVLtRlk=...</SignatureValue>
[s14]  <KeyInfo>
[s15a]   <KeyValue>
[s15b]    <DSAKeyValue>
[s15c]     <P>...</P><Q>...</Q><G>...</G><Y>...</Y>
[s15d]    </DSAKeyValue>
[s15e]   </KeyValue>
[s16]  </KeyInfo>
[s17] </Signature>
```

[s02-12] O elemento *SignedInfo* é a informação da assinatura.

[s14-16] O elemento *KeyInfo* indica a chave a ser usada para validar a assinatura. As formas possíveis para identificação incluem certificados, nomes das chaves e algoritmos.

3.4 XML Encryption Syntax and Processing

Este padrão W3C especifica o processo de criptografia de dados representado em formato XML [4].

Expressado numa forma curta, o elemento *EncryptedData* tem a seguinte estrutura (onde "?" significa zero ou uma ocorrência; "+" significa uma ou mais ocorrências; "*" significa zero ou muitas ocorrências; e um elemento tag vazio significa que o elemento será vazio):

```
  <EncryptedData Id? Type? MimeType? Encoding?>
    <EncryptionMethod/>?
    <ds:KeyInfo>
      <EncryptedKey>?
```

```
      <AgreementMethod>?
      <ds:KeyName>?
      <ds:RetrievalMethod>?
      <ds:*>?
    </ds:KeyInfo>?
    <CipherData>
      <CipherValue>?
      <CipherReference URI?>?
    </CipherData>
    <EncryptionProperties>?
  </EncryptedData>
```

Exemplo de um XML sem criptografia dos dados:

```
  <?xml version='1.0'?>
  <PaymentInfo xmlns='http://example.org/paymentv2'>
    <Name>John Smith</Name>
    <CreditCard Limit='5,000' Currency='USD'>
      <Number>4019 2445 0277 5567</Number>
      <Issuer>Example Bank</Issuer>
      <Expiration>04/02</Expiration>
    </CreditCard>
  </PaymentInfo>
```

Exemplo da mensagem anterior usando criptografia:

```
  <?xml version='1.0'?>
  <PaymentInfo xmlns='http://example.org/paymentv2'>
    <Name>John Smith</Name>
    <EncryptedData
Type='http://www.w3.org/2001/04/xmlenc#Element'
     xmlns='http://www.w3.org/2001/04/xmlenc#'>
      <CipherData>
        <CipherValue>A23B45C56</CipherValue>
      </CipherData>
    </EncryptedData>
  </PaymentInfo>
```

4. SOAP Message Security 1.0 (WS-Security 2004)

Esta especificação, definida pelo grupo OASIS, descreve avanços na mensagem SOAP para garantir integridade e confidencialidade [6]. O modelo especificado pode ser usado para acomodar uma variedade de modelos de segurança e tecnologias de criptografia. O exemplo abaixo mostra uma mensagem que usa segurança baseada em *tokens*, assinaturas e criptografia.

```
(001) <?xml version="1.0" encoding="utf-8"?>
(002)  <S11:Envelope  xmlns:S11="..."  xmlns:wsse="..."
xmlns:wsu="..." xmlns:xenc="..." xmlns:ds="...">
(003) <S11:Header>
(004) <wsse:Security>
(005) <wsu:Timestamp wsu:Id="T0">
(006) <wsu:Created>
(007) 2001-09-13T08:42:00Z</wsu:Created>
(008) </wsu:Timestamp>
(009)
(010) <wsse:BinarySecurityToken ValueType="...#X509v3"
wsu:Id="X509Token" EncodingType="...#Base64Binary">
(011) MIIEZzCCA9CgAwIBAgIQEmtJZc0rqrKh5i...
(012) </wsse:BinarySecurityToken>
(013) <xenc:EncryptedKey>
(014) <xenc:EncryptionMethod Algorithm=
"http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
(015) <ds:KeyInfo>
(016)                                <wsse:KeyIdentifier
EncodingType="...#Base64Binary"
ValueType="...#X509v3">MIGfMa0GCSq...
(017) </wsse:KeyIdentifier>
(018) </ds:KeyInfo>
(019) <xenc:CipherData>
(020) <xenc:CipherValue>d2FpbmdvbGRfE0lm4byV0...
(021) </xenc:CipherValue>
(022) </xenc:CipherData>
(023) <xenc:ReferenceList>
(024) <xenc:DataReference URI="#enc1"/>
(025) </xenc:ReferenceList>
(026) </xenc:EncryptedKey>
(027) <ds:Signature>
(028) <ds:SignedInfo>
(029) <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
(030) <ds:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
(031) <ds:Reference URI="#T0">
(032) <ds:Transforms>
(033) <ds:Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
(034) </ds:Transforms>
(035) <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
(036) <ds:DigestValue>LyLsF094hPi4wPU...
(037) </ds:DigestValue>
(038) </ds:Reference>
(039) <ds:Reference URI="#body">
(040) <ds:Transforms>
(041)                                    <ds:Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
(042) </ds:Transforms>
(043) <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
(044) <ds:DigestValue>LyLsF094hPi4wPU...
(045) </ds:DigestValue>
(046) </ds:Reference>
(047) </ds:SignedInfo>
(048) <ds:SignatureValue>
(049) Hp1ZkmFZ/2kQLXDJbchm5gK...
(050) </ds:SignatureValue>
(051) <ds:KeyInfo>
(052) <wsse:SecurityTokenReference>
(053) <wsse:Reference URI="#X509Token"/>
(054) </wsse:SecurityTokenReference>
(055) </ds:KeyInfo>
(056) </ds:Signature>
```

```
(057) </wsse:Security>
(058) </S11:Header>
(059) <S11:Body wsu:Id="body">
(060) <xenc:EncryptedData
Type="http://www.w3.org/2001/04/xmlenc#Element"
wsu:Id="enc1">
(061) <xenc:EncryptionMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-
cbc"/>
(062) <xenc:CipherData>
(063) <xenc:CipherValue>d2FpbmdvbGRfE0lm4byV0...
(064) </xenc:CipherValue>
(065) </xenc:CipherData>
(066) </xenc:EncryptedData>
(067) </S11:Body>
(068) </S11:Envelope>
```

Descrição de algumas seções deste exemplo:

As linhas (003)-(058) contêm o cabeçalho da mensagem SOAP.

As linhas (004)-(057) representam o bloco <*wsse:Security*> do cabeçalho. Este contém a segurança da informação para esta mensagem.

As linhas (005)-(008) especificam o momento da criação da mensagem.

As linha (010)-(012) especificam o *token* de segurança associado à mensagem. Neste caso é especificado um certificado X.509 que está codificado como *Base64*.

As linhas (013)-(026) especificam a chave usada para criptografar o corpo da mensagem.

As linhas (027)-(056) especificam a assinatura digital. Neste exemplo, a assinatura é baseada num certificado X.509.

A linha (039) referencia o corpo da mensagem.

As linhas (048)-(050) indicam o atual valor da assinatura que foi especificada na linha (043).

As linhas (052)-(054) indicam a chave usada na assinatura.

O corpo da mensagem está representada pelas linhas (059)-(067).

As linhas (060)-(066) representam os metadados da criptografia e formas do corpo usando *XML Encryption*.

As linha (063)-(064) contêm o corpo criptografado.

5. CRIANDO WEB SERVICES SEGUROS

Para criar Web Services seguros, conheça as ameaças associadas [5]. As principais ameaças aos Web Services são:

- Acesso não autorizado
- Manipulação de parâmetros
- Espionagem na rede
- Divulgação de dados de configuração
- Repetição de mensagens

A Figura 2 mostra as principais ameaças e ataques aos Web Services.

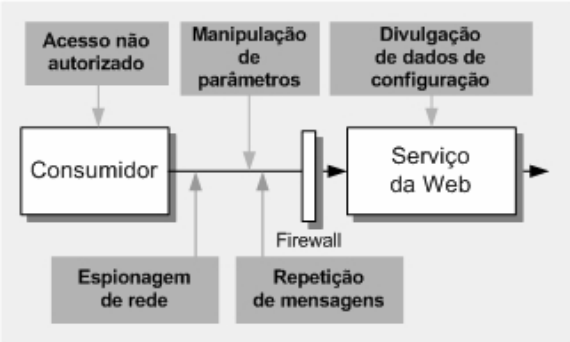

Figura 2 – Principais ameaças e ataques a Web Services

5.1 Acesso não autorizado

Vulnerabilidades

As vulnerabilidades que podem causar o acesso não autorizado por meio de um Web Service incluem:

- Falta de autenticação.
- Senhas de texto sem formatação passaram em cabeçalhos SOAP.
- Autenticação básica usada por um canal de comunicação sem criptografia.

Contramedidas

Para evitar o acesso não autorizado, podem ser usadas as seguintes contramedidas:

- Use resenha de senhas em cabeçalhos SOAP para autenticação.
- Use permissões Kerberos em cabeçalhos SOAP para autenticação.
- Use certificados X.509 em cabeçalhos SOAP para autenticação.
- Use autorização baseada em função para restringir o acesso aos Web Services. Isso pode ser feito com o uso da autorização de URL.

5.2 Manipulação de parâmetros

Vulnerabilidades

As vulnerabilidades que podem permitir a manipulação de parâmetros incluem:

- Mensagens que não são assinadas digitalmente e, portanto, não oferecem proteção contra violação.
- Mensagens que não são criptografadas e, portanto, não oferecem privacidade e proteção contra violação.

Contramedidas

Para evitar a manipulação de parâmetros, podem ser usadas as seguintes contramedidas:

- Assinar a mensagem digitalmente. A assinatura digital é usada no destinatário para verificar se a mensagem não foi violada enquanto estava em trânsito.

- Criptografar a carga da mensagem para oferecer privacidade e proteção contra violação.

5.3 Espionagem na rede

Vulnerabilidades

As vulnerabilidades que podem ativar a espionagem na rede com êxito incluem:

- Credenciais de texto sem formatação que passaram em cabeçalhos SOAP.
- Falta de criptografia no nível da mensagem
- Falta de criptografia no nível do transporte

Contramedidas

Para proteger mensagens SOAP confidenciais à medida que elas passam pela rede, podem ser usadas as seguintes contramedidas:

- Usar criptografia no nível do transporte, como SSL ou IPSec. Isso se aplica somente se houver controle em ambos os pontos de extremidade.
- Criptografar a carga da mensagem para oferecer privacidade. Essa abordagem funciona em situações nas quais a mensagem passa por rota de nós intermediários até o destino final.

5.4 Divulgação de dados de configuração

Vulnerabilidades

As vulnerabilidades que podem causar a divulgação de dados de configuração incluem:

- Arquivos WSDL irrestritos, disponíveis para download no servidor Web.
- Um Web Service irrestrito oferece suporte à geração dinâmica do WSDL e permite que consumidores não autorizados obtenham características do Web Service.
- Tratamento de exceções de baixa segurança

Contramedidas

Para evitar a divulgação indesejada de dados de configuração, podem ser usadas as seguintes contramedidas:

- Autorizar o acesso a arquivos WSDL através de permissões.
- Remover arquivos WSDL do servidor Web.
- Desativar os protocolos de documentação para impedir a geração dinâmica do WSDL.
- Capturar exceções e descartar uma exceção do tipo “SoapException” ou “SoapHeaderException” — que retorna somente informações mínimas e inofensivas — de volta para o cliente.

5.5 Repetição de mensagens

Vulnerabilidades

As vulnerabilidades que podem ativar a repetição de mensagens incluem:

- Mensagens não criptografadas.
- Mensagens que não são assinadas digitalmente e, portanto, não impedem a violação.
- Mensagens duplicadas não detectadas pela falta de identificação de mensagem exclusiva.

Ataques

Os tipos mais comuns de ataques de repetição de mensagens incluem:

- Ataque de repetição básico. O invasor captura e copia uma mensagem e, em seguida, repete a mesma mensagem e assume a identidade do cliente. Esse ataque de repetição não requer que o usuário mal–intencionado conheça o conteúdo da mensagem.
- Ataque de interceptadores. O invasor captura a mensagem, altera algum conteúdo, por exemplo, um endereço para remessa e, em seguida, a repete no Web Service.

Contramedidas

Para solucionar a ameaça de repetição de mensagens, podem ser usadas as seguintes contramedidas:

- Usar um canal de comunicação com criptografia, por exemplo, o SSL.
- Criptografar a carga da mensagem para oferecer privacidade de mensagem e proteção contra violação. Embora não impeça os ataques de repetição básicos, isso impede os ataques de interceptadores, nos quais o conteúdo da mensagem é modificado antes de ser repetido.
- Usar uma identificação de mensagem exclusiva ou um valor de uso único com cada solicitação para detectar duplicatas e assinar digitalmente a mensagem para oferecer proteção contra violação. **Observação:** um *valor de uso único* é um valor exclusivo usado criptograficamente para a solicitação. Quando o servidor responde ao cliente, ele envia uma identificação exclusiva e assina a mensagem, inclusive a identificação. Ao fazer outra solicitação, o cliente inclui a identificação com a mensagem. O servidor certifica–se de que a identificação enviada para o cliente na mensagem anterior seja incluída na nova solicitação do cliente. Se ela for diferente, o servidor rejeitará a solicitação e irá supor que está sujeito a um ataque de repetição. O invasor não pode falsificar a identidade da mensagem, pois ela está assinada. Observe que isso protege somente o servidor contra ataques de repetição iniciados pelo cliente usando a solicitação de mensagem e não oferece nenhuma proteção ao cliente contra respostas repetidas.

6. CONCLUSÃO

Como apresentado, os Web Services estão sendo utilizados a cada dia que passa como forma de integração entre sistemas heterogêneos utilizando a Internet como meio de transmissão. As formas apresentadas de se garantir segurança para os Web Services vão desde soluções simples, uso de HTTPS, até soluções mais sofisticadas e pouco complexas, uso de assinaturas digitais e criptografia de conteúdo. Várias tecnologias, como Java e .NET por exemplo, já possuem API’s que facilitam a implementação da segurança.

7. REFERÊNCIAS

[1] MENÉNDEZ, Andrés Ignácio Martinez. Uma ferramenta de apoio ao desenvolvimento de Web Services. Campina Grande, agosto de 2002.

[2] JAGIELLO, Iverson Lourenço & PERPÉTUO Júnior, Enio. Web Services - Uma Solução para Aplicações Distribuídas na Internet. Curitiba, 2003.

[3] XML-Signature Syntax and Processing - W3C Recommendation 12 February 2002 http://www.w3.org/TR/xmldsig-core/

[4] XML Encryption Syntax and Processing - W3C Recommendation 10 December 2002 http://www.w3.org/TR/xmlenc-core/

[5] Centro de Orientações de Segurança http://www.microsoft.com/brasil/security/guidance/topics/devsec/secmod85.mspx

[6] Web Services Security: SOAP Message Security 1.0 (WS-Security 2004) OASIS Standard 200401, March 2004 http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0

[7] The Java Web Services Tutorial http://java.sun.com/webservices/docs/1.5/tutorial/doc/index.html

[8] Web Services Security - Technology Evangelist Sun Microsystems http://developers.sun.com/dev/evangcentral/presentations/WebServicesSecurity.pdf

[9] Web Services Home http://msdn.microsoft.com/webservices/default.aspx