Spring WS ve Güvenlik

harezmi bilişim çözümleri

Web Servis Çağrılarının Güvenliği

- Web servis çağrılarının ve mesaj içeriğinin güvenliğinin sağlanmasında **üç temel konu** vardır
  - **Kimliklendirme**
  - Mesajın **imza**lanması
  - Mesaj içeriğinin **kripto**lanması
- Spring WS bu üç alanda da çözümler sunmaktadır

EndpointInterceptor ile Güvenlik

- Spring WS'deki güvenlik mekanizması **EndpointInterceptor** üzerine kuruludur
- **İki farklı implemantasyon**u vardır
  - Wss4jSecurityInterceptor: Apache WSS4J tabanlı
  - XwsSecurityInterceptor: Sun XWSS tabanlı
- **Transport protokol bağımsız** çalışma imkanı sunarlar
- Ancak kullanımları **karmaşıktır**

Wss4jSecurityInterceptor

JAVA Eğitimleri

- Apache **WSS4J**'i temel alır
- Harici **konfigürasyon dosyası** kullanmaz
- **Property**'ler ile **konfigüre** edilir
- İstemci tarafında **securementActions,** sunucu tarafında ise **validationActions** property'lerine **securement** ve **validation** action'ları tanımlanarak konfigüre edilir
- Bu action'lar hem istemci, hem de sunucu tarafında **aynı sıra ile tanımlanmalı**dır

İstemci Tarafı için Securement Action'ları

- **NoSecurity:** herhangi bir securement action uygulanmaz
- **UsernameToken:** username token ekler
- **UsernameTokenSignature:** username token ve bunun secret key'ini ekler
- **Timestamp:** timestamp ekler
- **Encrypt:** mesajı kriptolar
- **Signature:** mesajı imzalar

Wss4JSecurityInterceptor Konfigürasyonu - İstemci

```
<bean id="securityInterceptor"
class="org.springframework.ws.soap.security.wss4j.Wss4jS
ecurityInterceptor">
    <property name="securementActions"
value="UsernameToken"/>
    <property name="securementUsername" value="user1"/>
    <property name="securementPassword" value="secret"/>
    <property name="securementPasswordType"
value="PasswordText"/>
    <property name="securementUsernameTokenElements"
value="Nonce Created"/>
</bean>
```

Alabileceği değerler :PasswordText, PasswordDigest (default)

Wss4JSecurityInterceptor Konfigürasyonu - İstemci

```
<bean id="webServiceTemplate"
class="org.springframework.ws.client.core.WebServiceTemplate">
...
    <property name="interceptors">
        <array>
            <ref bean="securityInterceptor"/>
        </array>
    </property>
</bean>
```

Bir önceki adımda tanımlanan securityInterceptor bean'i WebServiceTemplate'ın **ClientInterceptor**'leri arasına eklenmelidir

Sunucu Tarafı için Validation Action'ları

- **NoSecurity:** herhangi bir işlem yapmaz
- **UsernameToken:** username token'ı validate eder
- **Timestamp:** timestamp'i validate der
- **Decrypt:** kriptolu mesajı çözümler
- **Signature:** imzayı validate eder

Wss4JSecurityInterceptor Konfigürasyonu - Sunucu

```
<bean id="securityInterceptor" class="org.springframework.ws.soap
        .security.wss4j.Wss4jSecurityInterceptor">
    <property name="validationActions" value="UsernameToken"/>
    <property name="validationCallbackHandler"
        ref="validationCallbackHandler"/>
</bean>
```

Sunucu tarafındaki konfigürasyon ValidationCallbackHandler'a ihtiyaç duyar. Bu callback handler'in farklı implemantasyonları mevcuttur

```
<bean id="validationCallbackHandler"
class="org.springframework.ws.soap.security.wss4j.callback.SimplePassw
ordValidationCallbackHandler">
    <property name="users">
        <props>
            <prop key="user1">secret</prop>
               <prop key="user2">123456</prop>
        </props>
    </property>
</bean>
```

Wss4JSecurityInterceptor Konfigürasyonu - Sunucu

```
<web-services:interceptors>
    <ref bean="securityInterceptor"/>
</web-services:interceptors>
```

Bir önceki adımda tanımlanan securityInterceptor bean'i sunucu tarafında Spring WS'e endpoint interceptor olarak eklenmelidir

Spring Security Entegrasyonu

- Kimliklendirme Spring Security tarafından yürütülür
- İşlem başarılı ise AuthenticationToken **SecurityContextHolder**'a set edilir
- Web servis request'inin sonunda **SecurityContext** temizlenmektedir
- Spring Security'ye özel **ValidationCallbackHandler**'lardan uygun olan birisi üzerinden gerçekleşir

Spring Security Entegrasyonu

```
<bean id="securityInterceptor" class="org.springframework.ws.soap.security
        .wss4j.Wss4jSecurityInterceptor">
    <property name="validationActions" value="UsernameToken" />
    <property name="validationCallbackHandler" ref="validationCallbackHandler" />
</bean>

<bean id="validationCallbackHandler" class="org.springframework.ws.soap.security
        .wss4j.callback.SpringSecurityPasswordValidationCallbackHandler">
    <property name="userDetailsService" ref="userDetailsService"/>
</bean>
```

Kullanılan ValidationCallbackHandler'a göre **UserDetailsService** veya **AuthenticationManager** enjekte edilebilir

```
<security:user-service id="userDetailsService">
    <security:user name="user1" password="secret" authorities=""/>
    <security:user name="user2" password="123456" authorities=""/>
</security:user-service>
```

Spring WS ve HTTP Basic Auth Kullanımı

- EndPointInterceptor yaklaşımına alternatif diğer yöntem **HTTP BASIC Auth** kullanılmasıdır
- Konfigürasyonu ve **kullanımı basit**tir
- **HTTPS** ile birlikte kullanıldığı vakit pek çok kullanım durumundaki gereksinimi karşılar
- Spring Security ile birlikte **kolayca konfigüre edilebilir**

Spring WS ve HTTP Basic Auth Kullanımı

- Ancak **transport protokol bağımlı**dır
- Sadece transport protocol olarak **HTTP kullanıldığı vakit** kullanılabilir

HTTP Basic Auth – Sunucu Tarafı

<security:http-basic/> elemanı basic authentication'ı aktive eder. <security:intercept-url> tanımı ile de WS request'lerinin kimliklendirmeye tabi tutulması sağlanır.

```
<security:http auto-config="false">
    <security:csrf disabled="true"/>
    <security:http-basic/>
    <security:intercept-url   pattern="/ws/**"
                              access="isFullyAuthenticated()"/>
</security:http>


<security:authentication-manager>
    <security:authentication-provider
            user-service-ref="userDetailsService"/>
</security:authentication-manager>

<security:user-service id="userDetailsService">
    <security:user name="user1" password="secret" authorities=""/>
    <security:user name="user2" password="123456" authorities=""/>
</security:user-service>
```

HTTP Basic Auth – İstemci Tarafı

İstemci tarafında WebServiceTemplate kullanılıyor ise default MessageSender yerine Apache Commons HttpClient'ı kullanan HttpComponentsMessageSender konfigüre edilmelidir

```
<bean id="webServiceTemplate"
class="org.springframework.ws.client.core.WebServiceTemplate">
...

<property name="messageSender">
        <bean class="org.springframework.ws.transport
                            .http.HttpComponentsMessageSender">
            <property name="credentials">
                <bean class="org.apache.http
                      .auth.UsernamePasswordCredentials">
                    <constructor-arg value="user1:secret"/>
                </bean>
            </property>
        </bean>
    </property>
</bean>
```

İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- http://www.java-egitimleri.com
- info@java-egitimleri.com