Servicios Web Seguros

Sesión 7: Seguridad a nivel de mensaje. Encriptación y envío de login

Puntos a tratar

- Encriptación de mensajes SOAP
- Encriptar la firma
- Encriptación de respuesta
- Envío de información de login

Encriptación de mensajes SOAP

- ...
- </sp:SignedParts>
- <sp:EncryptedParts>
- <sp:Body />
- </sp:EncryptedParts>

Firma del texto sin encriptar

- La firma se genera a partir del contenido no encriptado del mensaje.
- Se podría construir un diccionario de contenidos, asociados a firmas para así obtener indicios sobre el texto original a partir de la firma.
- Soluciones:
  - Encriptar también la firma
  - Encriptar antes de firmar
  - Añadir ruido “nonce”

Encriptar la firma

```
<sp:EncryptedElements>
    <sp:XPath>
            //*[local-name()='Signature']
    </sp:XPath>
</sp:EncryptedElements>
```

Encriptar antes de firmar

- <sp:EncryptBeforeSigning/>
- El mensaje encriptado es cada vez diferente, aunque sea para el mismo texto. Por tanto la firma también será diferente.
- Lo normal sería firmar el contenido visible por el usuario, pero firmar el contenido encriptado no representa ningún problema de seguridad.

Añadir ruido o “nonce”

- Añadir ruido aleatorio a cada mensaje
- y/o la hora exacta del instante actual
- Así cada vez la firma es diferente, a pesar de que el contenido significativo del mensaje sea el mismo

Encriptación de la respuesta

- El servidor recibe el mensaje encriptado junto con el certificado del cliente (con su clave pública)
- El servidor puede usar ese mismo certificado para encriptar la respuesta (sólo el cliente con la clave privada lo podrá desencriptar).
- Para indicar eso a rampart, en lugar del alias del cliente, se indica con “useReqSigCert”
  - Servidor: properties.put(SecurityConstants. ENCRYPT_USERNAME, "useReqSigCert");
  - Cliente: Servidor: properties.put(SecurityConstants. ENCRYPT_USERNAME, "servidor1");

Cuerpo del SOAP sin encriptar

```
<soap:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
  oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="Id-18227730">
   <ns2:saluda xmlns:ns2="http://jtech.ua.es/Seguro/">
      <nombre>Boyan</nombre>
      <apellido>Bonev</apellido>
   </ns2:saluda>
</soap:Body>
```

Cuerpo del SOAP encriptado

```
<soap:Body xmlns:wsu="http://docs.oasis-open.org/4d5wss/2004/01/
  oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="Id-16865950">
   <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
     Id="EncDataId-1" Type="http://www.w3.org/2001/04/xmlenc#Content">
      <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/
        xmlenc#tripledes-cbc" />
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
         <wsse:SecurityTokenReference xmlns:wsse="http://docs.oasis-open.org/
            wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
            <wsse:Reference xmlns:wsse="http://docs.oasis-open.org/wss/2004/
               01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
              URI="#EncKeyId-42419DE053BAC35578127413324278 72" />
         </wsse:SecurityTokenReference>
      </ds:KeyInfo>
      <xenc:CipherData>
         <xenc:CipherValue>575x5QxJXVErqWnjzpc5aL7p1UqI/zIUSbnOz4pYDMXxD37IV
           Qq2OLf0YYU1xwQXYrOQcdpQ8p/nJsEbFojhktOS+nn0u/V62tg+/MCGhz7FK3tu6h
           EdXYZrASs9eOHwUga8aQebnHConML+lHlwo4Bxzhs/jv+SuTkKFCOpGhCAf6oled0
           zd5ayTmh25y69EW95Gh+fw6a+mQQNEUVFUXAW43Fj9qBHuVmhVEZgN4Uo4Q6aUGf7
           cbcBCDkTONND/NGhHP8l1vNvLm0Bq3rL6STD0hxXLmP7ksR4RFUcyisJZK3O0zcP+
           TqubqyJG7Y2/GfBqKCAd1Fah7C8NER4oLq3LiQAPaNfsns/v+NtMUVm6kDwmbQ+mr
           wFPYFI1i0yhHw9yN50nQjQb36DpEDau4J9FPKRQhBhx4IB79sFco4=
         </xenc:CipherValue>
      </xenc:CipherData>
   </xenc:EncryptedData>
</soap:Body>
```

Información de login

- Token de seguridad con información de login
- Firmado para asegurar su autenticidad

```
<sp:SignedSupportingTokens>
    <wsp:Policy>
       <sp:UsernameToken sp:IncludeToken="http://docs.oasis-open.org/ws-sx/
         ws-securitypolicy/200702/IncludeToken/AlwaysToRecipient"/>

       </wsp:Policy>
</sp:SignedSupportingTokens>
```

Enviar login desde el cliente

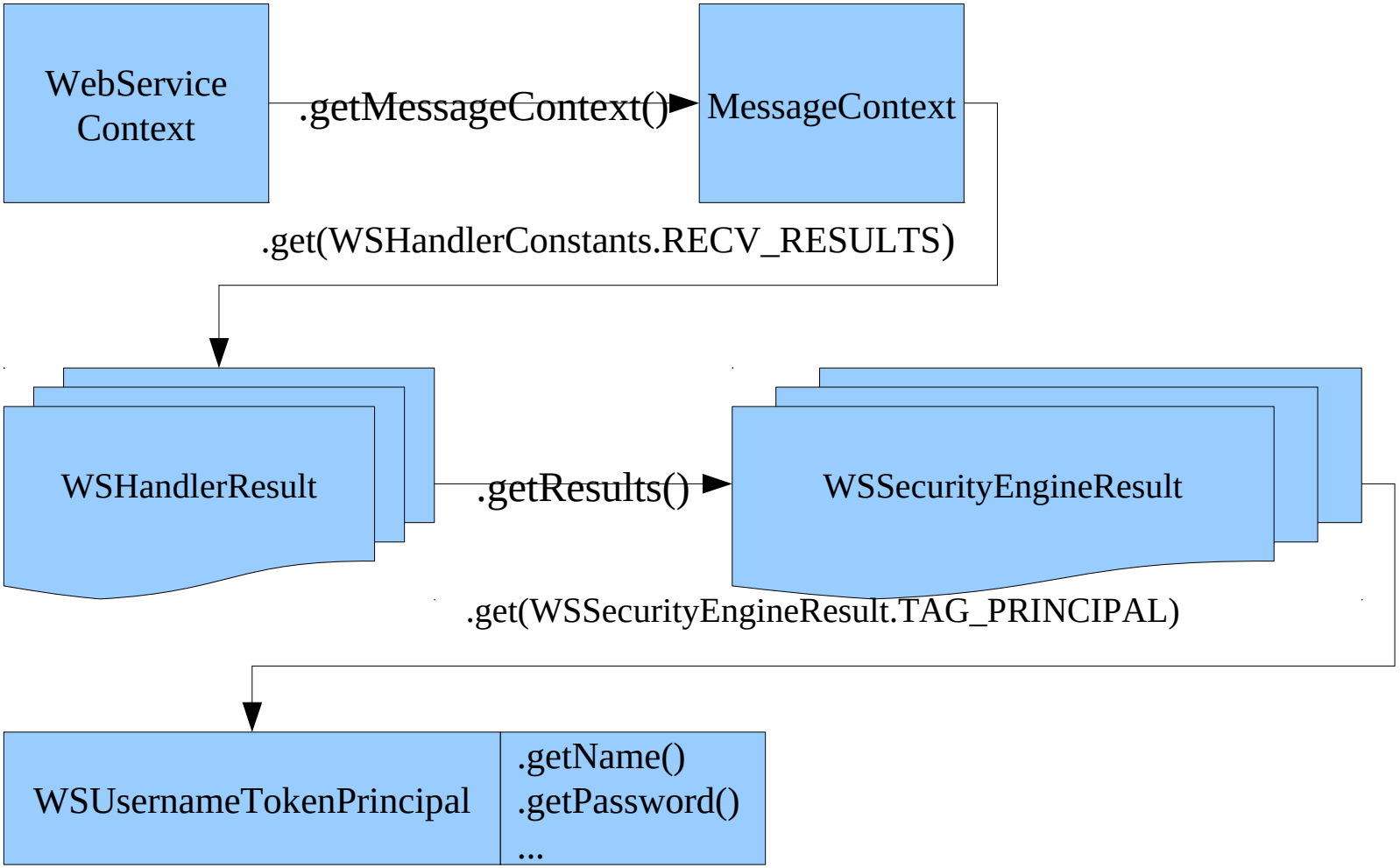
```
properties.put(SecurityConstants.USERNAME,
"pepe");

properties.put(SecurityConstants.PASSWORD,
"pepe-pass");
```

Recibir información de login

```
Vector<WSHandlerResult> handlerResults = (Vector<WSHandlerResult>)
  wsContext.getMessageContext().get(WSHandlerConstants.RECV_RESULTS);
for(WSHandlerResult handlerResult : handlerResults){
        for(WSSecurityEngineResult handler :
          (Vector<WSSecurityEngineResult>)handlerResult.getResults()){
                int action = ((Integer)handler.get(WSSecurityEngineResult.
                  TAG_ACTION)).intValue();
                if(action == WSConstants.UT){
                        WSUsernameTokenPrincipal utp = (WSUsernameTokenPrincipal)
                          handler.get(WSSecurityEngineResult.TAG_PRINCIPAL);
                        if(utp!=null){
                                if(utp.getName().equals("pepe") &&
                                   utp.getPassword().equals("pepe-pass")){
                                        return true;
                                }
                        }
                }
        }
}
```

Recibir información de login



¿Preguntas...?