BEA WebLogic Server R
WS-Security and Policy

Version: 10.3 Tech Preview

Document Date: October 2007

Table of Contents

WS-Security and WS-SecurityPolicy

The WS-SecurityPolicy 1.2 is an OASIS Standard as of July 2007, http://www.oasis-open.org/specs/index.php#wssecpolv1.2 . WebLogic Server 10.3 supports this version WS-SecurityPolicy (WSSP) specification, and the adoption of the following new namespace:

```
http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702
```

New Build-in Policies

The following build-in policies for security are supported in this release of WebLogic Server.

Policy Name	Descriptions
Transport Level Policy	
Wssp1.2-2007-Https.xml	Https only and no authentication
Wssp1.2-2007-Https-BasicAuth.xml	Https with basic auth authentication
Wssp1.2-2007-Https-ClientCertReq.xml	Https with client side certificates too
Wssp1.2-2007-Https-UsernameToken-Digest.xml	Https with digested Username Token for authentication
Wssp1.2-2007-Https-UsernameToken-Plain.xml	Https with plain-text Username Token for authentication
SAML Token Profile Policy	
Wssp1.2-2007-Saml1.1-HolderOfKey-Wss1.0.xml	SAML 1.1 Token Profile 1.0 Holder of Key with asymmetric binding
Wssp1.2-2007-Saml1.1-HolderOfKey-Wss1.1-Asymmetric.xml	SAML 1.1 Token Profile 1.1 Holder of Key with asymmetric binding
Wssp1.2-2007-Saml1.1-SenderVouches-Wss1.0.xml	SAML 1.1 Token Profile 1.0 Sender Vouches symmetric binding
Wssp1.2-2007-Saml1.1-SenderVouches-Wss1.1.xml	SAML 1.1 Token Profile 11 Sender Vouches symmetric binding
Wssp1.2-2007-Saml2.0-Bearer-Https.xml	SAML 2.0 Token Profile 1.1 Bearer with Https binding
Wssp1.2-2007-Saml2.0-HolderOfKey-Wss1.1-Asymmetric.xml	SAML 2.0 Token Profile 1.1 Holder of Key with asymmetric binding
Wssp1.2-2007-Saml2.0-SenderVouches-Wss1.1-Asymmetric.xml	SAML 2.0 Token Profile 1.1 Sender Vouches with asymmetric binding
Wssp1.2-2007-Saml2.0-SenderVouches-Wss1.1.xml	SAML 2.0 Token Profile 1.1 Sender Vouches with symmetric binding
WSS 1.0 Policy	
Wssp1.2-2007-Wss1.0-X509-Basic256.xml	WSS 1.0 X509 with asymmetric binding
Wssp1.2-2007-Wss1.0-UsernameToken-Digest-X509-Basic256.xml	WSS 1.0 X509 with asymmetric binding and authentication with digested Username Token
Wssp1.2-2007-Wss1.0-UsernameToken-Plain-X509-Basic256.xml	WSS 1.0 X509 with asymmetric binding and authentication with digested Username Token
WSS 1.1 Policy	
Wssp1.2-2007-Wss1.1-X509-Basic256.xml	WSS 1.1 X509 with asymmetric binding
Wssp1.2-2007-Wss1.1-UsernameToken-Digest-X509-Basic256.xml	WSS 1.1 X509 with asymmetric binding and authentication with digested Username Token
Wssp1.2-2007-Wss1.1-UsernameToken-Plain-X509-	WSS 1.1 X509 with asymmetric binding and authentication with

Basic256.xml	plain-text Username Token
Wssp1.2-2007-Wss1.1-EncryptedKey-X509-SignedEndorsing.xml	WSS 1.1 X509 with symmetric binding and protected by signed endorsing supporting token
Wssp1.2-2007-Wss1.1-UsernameToken-Digest-EncryptedKey.xml	WSS 1.1 X509 with symmetric binding and authentication with digested Username Token
Wssp1.2-2007-Wss1.1-UsernameToken-Plain-EncryptedKey.xml	WSS 1.1 X509 with symmetric binding and authentication with plain-text Username Token
Wssp1.2-2007-Wss1.1-DK-X509-SignedEndorsing.xml	WSS 1.1 X509 with derived key symmetric binding and protected by signed endorsing supporting token
Wssp1.2-2007-Wss1.1-UsernameToken-Digest-DK.xml	WSS 1.1 X509 with derived key symmetric binding and authentication with digested Username Token
Wssp1.2-2007-Wss1.1-UsernameToken-Plain-DK.xml	WSS 1.1 X509 with derived key symmetric binding and authentication with plain-text Username Token
WS-SecureConversation 1.3	
Wssp1.2-2007-Wssc1.3-Bootstrap-Https-BasicAuth.xml	WSSC 1.3 using Https BasicAuth for bootstrap
Wssp1.2-2007-Wssc1.3-Bootstrap-Https-ClientCertReq.xml	WSSC 1.3 using Https plus Client Certificate for bootstrap
Wssp1.2-2007-Wssc1.3-Bootstrap-Https.xml	WSSC 1.3 using Https only for bootstrap
Wssp1.2-2007-Wssc1.3-Bootstrap-Wss1.0.xml	WSSC 1.3 using Wss1.0 standard for bootstrap
Wssp1.2-2007-Wssc1.3-Bootstrap-Wss1.1.xml	WSSC 1.3 using Wss1.1 symmetric binding for bootstrap
Encryption and Signature Policy	
Wssp1.2-2007-SignBody.xml	Sign Body part policy
Wssp1.2-2007-EncryptBody.xml	Encrypt Body part policy
Wssp1.2-2007-Sign-Wsa-Headers.xml	Sign WSA header parts policy

The existing build-in policies will still be supported.

New Policy Assertions

The following policy assertions are now supported in this release:

WS-SP Spec Sectn	**Assertion**	**Supported New Assertions**	**Comments**
4.1.2 4.2.2 4.2.3 4.3.1 4.3.2	SignedElements EncryptedElements ContentEncryptedElements RequiredElements RequiredParts	<sp:SignedElements> <sp:EncryptedElements> <sp:ContentEncryptedElements> <sp:RequiredElements> <sp:RequiredParts>	Support both XPathFilter20 and XPath. For signature
5.4.3	X509Token	<sp:WssX509Pkcs7Token10> <sp:WssX509Pkcs7Token11> <sp:WssX509PkiPathV1Token10> <sp:WssX509PkiPathV1Token11>	

5.4.6	SecurityContextToken		WS-SC 1.3 assertion items.
5.4.7	SecureConversationToken	<sp:IssuerName> <sp:RequireExternalUriReference>	
5.4.8	SamlToken		Refer to SAML TP document
6.3	ProtectionOrder	<sp:EncryptBeforeSigning>	<sp:SignBeforeEncrypting> is supported in 1.0
6.7/7.2	Security Header Layout Property	<sp:Strict> will be supported in a Essex	Indigo Interop item.
8.3 8.4 8.5	EndorsingSupportingTokens SignedEndorsingSupportingTokens SignedEncrtptedSupportingTokens	<sp:RequireDerivedKeys/> is support inside the EndorsingSupportingTokens.	
10.1	Trust13 Assertion	<sp:Trust13>	support both Trust10 and Trust13 assertions in this release

The following policy assertions will NOT be supported in Essex:

WS-SP Spec Sectn	**Assertion**	**Not Support in Essex**	**Comments**
5.1.1	TokenInclusion	includeTokenPolicy=Once is not supported	
5.4.1	UsernameToken	Only <sp:UsernameToken11> and Password Derived Keys is not supported in Essex. Other Username Tokens assertions are supported.	
5.4.2	IssuedToken	WS-Trust Policy assertion is not supported in ESSex	Refer to WS-Trust func spec.
5.4.3	X509Token	Support all token types, except X509V1. <sp:WssX509V1Token10> and <sp:WssX509V1Token11> are not supported.	
5.4.4	KerberosToken	Will be supported in a future release	
5.4.5	SpnegoContextToken	May be supported in a future release based on real-world use cases and customer preferences.	
5.4.9	RelToken	No plan for supporting this token in the near future	
5.4.11	KeyValueToken	Will be supported in a future release	
6.5	Token Protection	Token Protection in cases where includeTokenPolicy="Never" or in cases where the Token is not in the Message is not supported	
7.1	AlgorithmSuite	/sp:AlgorithmSuite/wsp:Policy/sp:XPathFilter20 assertion, /sp:AlgorithmSuite/wsp:Policy/sp:XPath10 assertion and /sp:AlgorithmSuite/wsp:Policy/sp:SoapNormalization10 will be supported in the future release	
8.1	SupportingTokens	../sp:SignedParts assertion, ../sp:SignedElements assertion, ../sp:EncryptedParts assertion and ../sp:EncryptedElements	

		assertion will NOT be supported in Essex	
8.2 8.3 8.4 8.5	SignedSupportingTokens EndorsingSupportingTokens SignedEndorsingSupporting Tokens SignedEncrtptedSupporting Tokens	../sp:SignedParts assertion, ../sp:SignedElements assertion, ../sp:EncryptedParts assertion, and ../sp:EncryptedElements assertion will NOT be supported in Essex	
		The runtime will not be able to endorse the supporting token in cases where the Token is not in the Message (such as for includeTokenPolicy=Never/Once).	
8.6	EncryptedSupportingTokens	Not supported, except the UserName Token , i.e. UserName Token is the only EncryptionSupportingTokens will be supported in Essex. Other type of tokens will not be supported.	
8.7	EndorsingEncryptedSupporting Tokens	Not supported in Essex	
8.8	SignedEndorsingEncrypted SupportingTokens		
9.1	WSS10 Assertion	<sp:MustSupportRefExternalURI> and <sp:MustSupportRefEmbeddedToken> is not supported	
9.2	WSS11 Assertion		
10.1	Trust13 Assertion	MustSupportClientChallenge, MustSupportServerChallenge are not supported in Essex. We only support this assertion in the WS-SecureConversation policy.	

Smart Policy Selection

Multiple policy alternatives for any given Web Services are supported. In this release, many new features are provided to make the policy selection smarter and thus allowing the same service to support the use cases with multiple policy alternatives:

1. Different version of the standard. For example, WSRM 1.0 and WSRM 1.1, WSS1.0 and WSS 1.1, WSSC 1.1 and WWSSC 1.2, SAML 1.1 or SAML 2.0.
2. Different credentials for authentication. For example, allow either Username Token, X509, or SAML token for authentication.
3. Different security requirements for internal and external. For example, external authentication requires a SAML token while internal employee authentication only requires a Username token for authentication.

The Web Services client can also handle multiple policy alternatives. The same client can interoperate with different services that have different policy or policy alternatives.

For example, the same client can talk to one service that requires SAML 1.1 Token Profile 1.0 for authentication while another service requires SAML 2.0 Token Profile 1.1 for authentication.

WS-Policy and WS-SecurityPolicy Versioning

In WebLogic Server 10.3, the version independent policy is supported. WS-Policy is the foundation for WS-SecurityPolicy as well as WS-RM and MTOM. WebLogic Server 10.0 only supported WS-Policy 1.2 with the namespace of:

```
http://schemas.xmlsoap.org/ws/2004/09/policy
```

WS-Policy 1.5 is now W3C standard with a namespace of:

```
http://www.w3.org/ns/ws-policy
```

WS-SecurityPolicy has similar situation. In WebLogic Server 10.0, the following namespace was supported for WSSP:

```
http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200512
```

In this release, the following OASIS S-SX TC WSSP namespace is supported:

```
http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702
```

In most of the cases, the policy assertions are identical for either namespaces with the following exceptions:

- Trust10 and Trust13 assertion
- SC10SecurityContextToken and SC13SecurityContextToken
- Derived Key using deferent WSSC versions

In a production environment, the policy can come from different sources with different namespaces. At run-time, the merged policy file may contain two or more differentWSP namespaces.

In WebLogic Server 10.3, the policy and security policy can be either namespace. Regardless whether the policy is in a WSP 1.2 or WSP 1.45 namespace, or a WSSP 1.2 or pre 1.2 namespace, it will be processed and the result is the same SOAP Message output.

Optional Policy

WebLogic Server 10.3 supports the `Optional` WS-Policy function. For example, the following policy assertion is now supported:

```
(P001)    <sp:SgnedEncryptedSupportingTokens>
(P002)        <wsp:Policy>
(P003)          <sp:UsernameToken
(P004)            sp:IncludeToken="…/IncludeToken/AlwaysToRecipient"
wsp:Optional="true" >
(P005)            <wsp:Policy>
(P006)              <sp:WssUsernameToken10/>
(P007)            </wsp:Policy>
(P008)          </sp:UsernameToken>
(P009)        </wsp:Policy>
(P010)      </sp:SignedEncryptedSupportingTokens>
```

Specifying the Username Token for authorization is now optional. The client will continue if it cannot generate the Username Token (meaning when the user is anonymous or when there is no security context). The server also should be able to process messages that do not have the Username Token instead of rejecting the message.

During the Security Policy enforcement process, the message will not be rejected if the missing element has the Policy assertion with the attribute of `wsp:Optional="true"`.

The following security policy assertions are now supported by the `Optional` policy assertion:

- Username Token
- SAML Token
- Signature parts or signature elements
- Encryption parts or encryption elements
- Derive Key Token
- Signature Confirmation

Smart Policy Selection

In this release, WebLogic Web Services can make smarter policy selection for both outbound message building and inbound policy enforcement process.

Policy Selection Preference

In this release of WebLogic Server, Policy Selection preferences can now be configured in the WebLogic Server Administration Console. The following preferences are supported:

- Security
- Performance
- Compatibility

The availability of the credential (for example, `UNT`, `X509`, or SAML) should be queried at runtime by the credential provider.

The policy selection preference is set via MBean API or Stub properties. The property name is `weblogic.wsee.policy.selection.preference`. The following values are supported:

- `SCP`
- `SPC`
- `CSP`
- `CPS`
- `PCS`
- `PSC`
- `NONE (default)`

where

`S`–Security or functionality

`C`–Compatibility

`P`–Performance

The policy selection preference can be set via stub property or thru the MBean for the given client. The following example sets the stub property for security, compatibility, and performance preferences:

```
stub._setProperty(PolicyConstants.POLICY_SELECTION_PREFERENCE,
            PolicyConstants.PREFERENCE_SECURITY_COMPATIBILITY_PERFORMANCE);
```

If the policy selection preference is not set, then the default preference is used.

If there are multiple policy choices, the system will use the configured preference list, the availability of the credential and setting of the optional function to determine the best selection policy.

Selection Rules

If multiple policy alternatives exist for a client, the following selection rules are used:

- If the preference is not set, the first policy alternative will be picked, except if the policy alternative is defined as `wsp:optional=true`.
- If the preference is set to security first, then the policy that has the most security features is selected.

- If the preference is set to compatibility/interop first, then the policy that has lowest version is selected.
- If the preference is set to performance first, then the policy that the least security features is selected.

For the optional policy assertions, the selection rules are used:

- If the `default` policy selection preference is set, then the optional attribute on any assertion is ignored.
- If the `Interop` or `Performance` preferences is set, then any assertion with an optional attribute is ignored (the attribute is not ignored, therefore the assertion is ignored).
- If the `Security` policy selection preference is set, optional assertions are included and alternative assertions are never generated.