# Security Considerations for the HPC Profile

Blair Dillaway, Microsoft
Marty Humphrey, University of Virginia
Jim Basney, NCSA/UIUC

In this paper, we describe interoperable security mechanisms which HPC Basic Profile compliant implementations must support. These mechanisms are limited to those necessary to address the requirements of the "Base Case" (Section 2) of the HPC Use-Case Document [HPC-U]. Compliant implementations may support additional security mechanisms required for extended functionality as discussed in Section 3 of [HPC-U].

## *1.1 Security Requirements of the HPC Basic Profile*

The environment in which an HPC Basic Profile service/client will operate is described below along with the requirements for securing the HPC Basic Profile messages.

### 1.1.1 Environment Assumptions

In addressing the Base Case some common assumptions are made about the environment and relationships between the users and BES web service schedulers. The security mechanisms defined in this specification build on this environment.

1. There is an identity management infrastructure deployed for provisioning users and services with identity credentials.
    - Web services are provisioned with X.509 [RFC 3280] service certificates following industry standard practice.
    - It is required that users be provisioned with username-password credentials or X.509 certificates. If an organization uses X.509 client certificates, username-password credentials may also be utilized but are not required.
2. Trust relationships are pre-configured and uniform
    - Users trust the CA(s) issuing X.509 service certificates and services trust the authority provisioning username-password credentials or the CA(s) issuing X.509 user certificates.
    - All BES Web services are fully trusted with respect to managing and executing activities within the environment and safeguarding any confidential user and activity information.
    - Users may not fully trust each other. They may require their activities be free from tampering by other users, or in some cases that the details of their activities (job type, data source, ..) not be exposed to other users.
3. X.509 certificate revocation may be supported using industry standard mechanism such as CRLs [RFC3280] and OCSP [RFC 2560] responders. It is up to the relying party whether to take advantage of revocation information.
4. It is assumed BES services are well-known to users and other services and may be located using commonly deployed mechanisms such as DNS (Domain Name Service) or UDDI (Universal Description Discovery and Integration) look-ups.
5. Authorization is based on authenticated user/service identities and attributes carried in the provisioned identity credentials. The authorization mechanism employed is outside the scope of this specification.

### 1.1.2 Securing the HPC Profile Messages

There is a need to secure messages exchanged between users and BES scheduler services to support the Base Case. The security mechanisms must support required message sender authentication (BES requests and responses), integrity protection, and confidentiality. These are summarized below:

**BES Request Message Authentication** – BES services require authentication of clients (may be a user or other service) invoking their services to ensure only authorized actions are performed. This includes, limiting who may create an activity, cancel an activity, and query an activity's status.

**BES Response Message Authentication** – Entities requesting BES services will require authentication of the responding service. This is needed to ensure that returned status information or faults can be relied upon.

**Integrity Protection** – High assurance message integrity is necessary to prevent attackers from modifying activity definitions for purposes such as creating incorrect billing or denial of service.

**Confidentiality** - In some environments, activity details and status information will be considered confidential. As such, it will be mandatory to encrypt the BES messages to prevent disclosure to unauthorized entities. Confidentiality of this information may not be critical in other environments, though message encryption is still acceptable.

## *1.2 HPC Basic Profile Message Security*

This specification takes the position that security interoperability for the Base Use case is best achieved through a few widely deployed, standards-based, technologies and vetted implementation guidance. It is not a goal of this specification to innovate in the security area or drive adoption of new technologies.

To that end, use of TLS/SSL transport layer security as the basis for interoperable secure messages is adopted. This provides greater functionality that absolutely required for some environments, but minimizes the number of mechanisms which must be supported. It is not believed the tools, and supporting infrastructure, for interoperable message-level security (based on the WS-* family of specifications) have reached the level of adoption and deployment needed to rely on their use as the primary security mechanism for this profile.

The HPC Basic Profile builds on the "WS-I Basic Security Profile" (WS-I BSP) [WS-I Basic Security Profile Version 1.0, Working Group Draft,2006-08-17] as the foundation for interoperable message security. In particular, the transport layer security mechanisms identified in Section 4 of that specification are used. The more restrictive cipher suite guidelines specified in the "OGSA Basic Security Profile 1.0 - Secure Channel" (OGSA BSP-SC) are also adopted as described in Section 1.3.

(Note: The "OGSA Basic Security Profile 1.0 - Core" specification is not used as that addresses the binding of key information to an endpoint reference [in WS-Addressing], which is not relevant when using transport layer security.)

The HPC Basic Profile message security mechanisms and requirements are defined in Section 1.3 and 1.4. Compliant implementations are required to fully implement one of these mechanisms, though they may support both. The terminology of the WS-I BSP is used to define compliant implementations. Specifically, a conforming INSTANCE is "software that implements a wsdl:port or a uddi:bindingTemplate".

## *1.3 TLS/SSL using X.509 Certificate Based Mutual Authentication*

This specification supports use of the Transport Layer Security (TLS 1.0 and TLS 1.1)[RFC2246 and RFC 4346] or Secure Sockets Layer (SSL 3.0) protocol for BES message security with mutual authentication of the sender and receiver based on X.509 v3 certificates. This is done in accordance with the recommendations of WS-I BSP and the more restrictive cipher suite guidance of the OGSA BSP-SC specification. Faults shall be handled in accordance with the TLS/SSL specifications.

Specific requirements of this specification are:

R0501: An INSTANCE MUST support TLS 1.0, SHOULD support SSL 3.0, and SHOULD support TLS 1.1.

R0502: An INSTANCE MUST support the FIPS-140 compliant

Ciphersuites TLS_RSA_FIPS-WITH_3DES_EDE_CBC_SHA and TLS_RSA_FIPS_WITH_3DES_EDE_CBC_SHA

R0503: An INSTANCE SHOULD support TLS_RSA_WITH_AES_128_CBC_SHA and TLS_RSA_WITH_AES_128_CBC_SHA.

R0504: An INSTANCE MUST support X.509 v3 certificates using RSA cryptographic keys and RSA/SHA-1 (http://www.w3.org/2000/09/xmldsig#**rsa-sha1)** digital signatures.

R0505 An INSTANCE SHOULD support X.509 v3 certificates using RSA cryptographic keys and RSA/SHA-256 (http://www.w3.org/2001/04/xmldsig-more#**rsa-sha256)** digital signatures.

R0506: An INSTANCE must use TLS/SSL encryption key agreement based on the RSA algorithm. Diffie-Helman key agreement shall not be used.

R0507 An INSTANCE MUST support server authentication using X.509 v3 certificates.

R0508: An INSTANCE MUST support client authentication using X.509 v3 certificates.

## *1.4 TLS/SSL with Username-Password Client Authentication*

This specification supports use of the TLS or SSL protocol for BES message security with x.509 server authentication and username-password based client authentication. When using this mechanism, a secure TLS/SSL session with the BES service must be first established. This is done in conformance with the recommendations contained in the WS-I BSP and requirements R0501 through R0507 above. That is, service authentication is done using an X.509 service certificate and a channel encryption key negotiated using RSA key transport.

Once an encrypted and integrity protected transport layer channel has been established, the client may transmit an HPC Basic Profile supported request messages, including their username-password authentication information as specified in the Username Token Profile 1.1 specification [Web Services Security UsernameToken Profile, Working Draft 2, OASIS, 23 Feb 2003].

Specific requirements of this specification are:

R0509: An INSTANCE MUST support client authentication using username/password credentials with cleartext (http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText') type encoding.

R0510: An INSTANCE MAY support client authentication using username/password credentials with digest (http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordDigest) type encoding.

Since all password information is communicated within a secure transport layer by compliant implementations, this specification does not specify use of message-level encryption. Also, use of nonces or creation times to prevent replay attacks is not required by this specfication and these may be omitted from a password digest calculation.

Faults occuring during TLS/SSL negotiation shall be handled in accordance with the TLS/SSL specifications. If faults arise based on processing of the clients username-password credential by the service, the service may silently drop the request message or respond with a SOAP fault message. When responding with a fault message, if the service is unable to validate the supplied credentials a SOAP fault with faultcode 'Client' should be returned otherwise a fault with faultcode 'Server' shall be returned. Compliant BES service implementations may wish to implement mechanisms to limit the number of invalid authentication attempts for a given username to prevent password guessing attacks.

An example CreateActivity message, including a username and digest password is shown below.

```
<s11:Envelope
  xmlns:s11="http://schemas.xmlsoap.org/soap/envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:bes-"factory="http://schemas.ggf.org/bes/2006/08/bes-factory"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" >
  <s11:Header>
    <wsse:Security>
      <wsse:UsernameToken xmlns:wsse='http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd' >
        <wsse:Username>Bert</wsse:Username>
        <wsse:Password  Type='http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-username-token-profile-1.0#PasswordText'>Ernie</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
    <wsa:Action>
       http://schemas.ggf.org/bes/2006/08/bes-factory/GetActivitiesStatus
    </wsa:Action>
    <wsa:To s11:mustUnderstand=1>
        http://www.bes.org/BESFactory
    </wsa:To>
  </s11:Header>
  <s11:Body wsu:Id='TheBody'>
        <bes-factory:CreateActivity>
            <bes-factory:activityDescriptionDocument>
                <bes-factory:ActivityDocument>
                    {Any valid JSDL document}
                </bes-factory:ActivityDocument>
            </bes-factory:activityDescriptionDocument>
        </bes-factory:CreateActivity>
  </s11:Body>
</s11:Envelope>
```