EMI Messaging Protocol for Accounting

Introduction

It has been agreed that accounting systems in EMI should use a common messaging protocol, for ease of interoperation.

We will use STOMP as the underlying protocol. All messages sent by clients are sent to one destination. It is essential that the destination is configured to be persistent, so that clients can assume that a message acknowledged by the broker is received successfully.

The protocol

Message Headers

Key	Value	Mandatory?
'destination'	<destination>	Yes
'msg-id'	Unique ID per message and client	No

Accounting systems may use additional headers for filtering or redirection.

Message Body

The raw content of the message is not covered here. Any content may be sent using this protocol.

The raw message MUST be signed.

The signed message MAY be encrypted.

Encryption and signatures

Each message MUST be signed and MAY be additionally encrypted. For all purposes the S/MIME (RFC 5751) MUST be used. To guarantee interoperability, security and simplified processing the following requirements MUST be followed. Note that the rules are fully independent from the actual content of the message.

Messages should be signed using a host certificate and key local to the client site. If messages are encrypted, the encryption should use the certificate of the accounting server. Certificate distribution is not covered in this document.

1) General requirements

The order of operations applied to the message being prepared for sending MUST be as follows:

1. (mandatory) signing
2. (optional) encryption

Each S/MIME structure MUST contain the smime-type parameter in Content-Type header (in the [SMIME] specification this is suggested but not enforced). The smime-type parameter MUST follow the rules of the [SMIME] specification.

2) Signing requirements

The message MUST be signed using the multipart/signed content type as described in the 3.4.3 section of [SMIME]. The micalg parameter MUST be specified and the algorithm used must be either SHA-1 or one of the SHA-2 algorithms, i.e. one of sha-224, sha-256, sha-384 or sha-512.

3) Encryption requirements

The encryption algorithm MUST be one of the AES algorithms. The key encryption algorithm is certificate dependent.

Example

Example using openssl 1 with both steps applied separately:

```
openssl cms -sign -inkey pk.pem -signer cert.pem -md sha256 -text <input.txt >input-sig.txt

openssl cms -encrypt -aes256 cert.pem <input-sig.txt >final.txt
```

Parsing

It should be possible to parse the message content using the MIME headers. From the Content-Type (together with smime-type for encrypted), one can know what is inside without parsing the binary ASN.1 CMS data:

Encrypted Data

```
Content-Type: application/pkcs7-mime; smime-type=enveloped-data; name="smime.p7m"
```

Signed Data

```
Content-Type: multipart/signed; protocol="application/pkcs7-signature";...
```

Notes

- SMIME headers are part of the message body. STOMP headers are separate to the message body.
- Use of detached signatures creates smaller messages; these messages are also human readable.
- You should add a 'receipt' header to each message. This will cause the broker to reply with a RECEIPT frame to indicate that the broker has received the message.