BasicTrustEngine

The Basic engine is found in ShibOnedotThree and extracts keys and certificates directly from MetaData to evaluate signatures or TLS credentials.

It has the following behavior, implications, and problems:

Validating Signatures

Each `<md:KeyDescriptor>` is resolved into a key. If the signature can be verified with one of the keys, then the engine returns success.

The following `<ds:KeyInfo>` children can be resolved into keys without additional plugin support:

- `<ds:KeyValue>/<ds:RSAKeyValue>`
- `<ds:KeyValue>/<ds:DSAKeyValue>`
- `<ds:X509Data>/<ds:X509Certificate>`

Note that under no circumstances is an X.509 certificate evaluated on any level when resolving a key. If it is a correctly encoded certificate, the signed key will be resolved. Valid or expired certificates issued by any signer with any sort of extensions are acceptable.

Revocation

Removing a `<md:KeyDescriptor>` from a metadata role is equivalent to revoking the enclosed key. The frequency of metadata update determines the window of exposure at any given provider.

Key Rollover

Since each key found is evaluated, new keys can be introduced by registering them in metadata, waiting a pre-defined period of time for the change to propagate, and then finally deploying the new signing key.

Known Issues

Versions of the ShibOnedotThree C++ ServiceProvider prior to the latest, 1.3.1, mistakenly ignore any `<md:KeyDescriptor>` without a `use` attribute set to "signing". 1.3.1 corrects this and permits descriptors with no `use` attribute to be applied.

Validating TLS and X.509 Credentials

> ***Note:*** *As of version 1.3.1 (currently IdP only), the behavior is now identical to the ExplicitKeyTrustEngine. Otherwise, the behavior is as described below.*

Each `<md:KeyDescriptor>` is resolved into a certificate chain. The first certificate in the chain (assumed to be the first one in order) is directly compared to the client or server TLS certificate presented. If they match exactly, then the engine returns success.

The following `<ds:KeyInfo>` children can be resolved into keys without additional plugin support:

- `<ds:X509Data>/<ds:X509Certificate>`

Note that under no circumstances is an X.509 certificate evaluated on any level **by Shibboleth** during the comparison. Valid or expired certificates issued by any signer with any sort of extensions are acceptable as long as they match what is presented exactly.

However, see the Known Issues below for caveats to this caused by other software.

Revocation

Removing a `<md:KeyDescriptor>` from a metadata role is equivalent to revoking the enclosed certificate. The frequency of metadata update determines the window of exposure at any given provider.

Key Rollover

Since each certificate chain found is evaluated, new certificates can be introduced by registering them in metadata, waiting a pre-defined period of time for the change to propagate, and then finally deploying the new certificate.

Known Issues

Since most other SAML implementations are likely to enforce the usual TLS name checking rules when evaluating a TLS server certificate, this checking is not disabled by Shibboleth. As a result, the certificates CN or subjectAltName(s) are checked against the expected hostname. This does not add any actual security to the system since a non-matching certificate won't be accepted anyway, but is still a requirement for deployers.

All known versions of Apache with mod_ssl have support for the `SSLVerifyClient optional_no_ca` setting to turn off certificate evaluation. However, even with this option, the depth of the certificate chain presented by the client is still wrongly evaluated in some strange fashion. Therefore, disabling verification must be accompanied by a high setting for `SSLVerifyDepth` .