IdP Key Handling

Handling the IdP Private Key

This topic discusses the proper handling of the IdP's private signing key. The corresponding public key is bound to a self-signed certificate in IdP metadata. Note that a signing key may be used for more than just signing, as discussed in the Key Usage topic. See the TLS Server Certificates topic regarding keys and certificates used for browser-facing TLS.

The IdP's private signing key is used to sign SAML assertions transmitted to the SP. (The corresponding certificate in metadata contains the public key that is used by the SP to verify the signature on the assertion.) If the private key is lost or stolen, the holder has the power to issue arbitrary assertions to **an** SP. This is the absolute worst thing that can happen in a federated context.

> **Protect your private keys!**
>
> Positive control of your private keys must be maintained at all times. This includes the private keys used for browser-facing TLS as well as your private signing key.

The IdP's private signing key is necessarily an online key, that is, it must be available to the IdP software at runtime. An online key may be encrypted, but the password or passphrase used to decrypt the key generally has to be available in an unencrypted file so that the IdP service can be restarted in unattended fashion. Therefore an online key is considerably more vulnerable than an offline key, and must be protected accordingly.

If the signing key is stored in the file system as an ordinary file, it should have strict permissions to prevent unauthorized copying of the private key. For stronger protection, the signing key may be stored in a hardware security module (HSM) that prevents export of the private key.

Key Audit

You need to go back to day one of the total lifetime of each of your private keys and ask the following question: *Has this key been under my positive control at all times?* If the answer to that question is anything other than yes, the key should be considered compromised. This requires you to securely generate a new private key and to systematically migrate the corresponding public key certificate into metadata. See the Certificate Migration topic for safe instructions how to do this.

If there is reason to believe that the IdP's signing key has fallen into the wrong hands, it should be replaced immediately. In this case, there can be no orderly migration of the corresponding public key certificate in metadata, which should also be replaced immediately. This will break interoperability with SPs until such time as they have refreshed metadata (which is why regular, automated metadata refresh is important), so replace the IdP's signing key only under the most serious circumstances.

Key Generation

See the Key Generation topic for instructions how to generate a secure private signing key for your IdP.