Phase 1 Implementation Plan FAQ

Phase 1 Implementation Plan: Frequently Asked Questions

In September 2013, the Metadata Distribution Working Group submitted its Phase 1 Recommendations to the InCommon Technical Advisory Committee. This FAQ anticipates questions and concerns regarding a Phase 1 Implementation Plan to be initiated December 2013.

General Questions

Is it really true that the HTTP location of InCommon metadata is changing?

Yes, for the first time since the formation of the InCommon Federation, the location of InCommon metadata is changing. This is a big deal, we know, but the time is ripe for change.

InCommon Operations will deploy three new metadata aggregates at the following **permanent** HTTP locations:

- http://md.incommon.org/InCommon/InCommon-metadata.xml (production)
- http://md.incommon.org/InCommon/InCommon-metadata-fallback.xml (fallback)
- http://md.incommon.org/InCommon/InCommon-metadata-preview.xml (preview)

The new locations will replace the current HTTP location of production metadata:

- http://wayf.incommonfederation.org/InCommon/InCommon-metadata.xml (legacy)

Moving forward, all new metadata services will be deployed on vhost `md.incommon.org`. Legacy vhost `wayf.incommonfederation.org` will be phased out.

Why are you moving the metadata to a new vhost?

Multiple, heterogeneous services currently run on legacy vhost `wayf.incommonfederation.org`, namely, InCommon Metadata Services and the Discovery Service. To provide better quality of service, these services need to be segregated onto separate vhosts (`md.incommon.org` and `ds.incommon.org`, resp.). Note: The InCommon Federated Error Handling Service is already running on `ds.incommon.org`.

If this were the *only* reason to move to a new vhost, we probably wouldn't be doing it (which is why we haven't done it until now). There are other, more significant reasons for making this change. See the Phase 1 Implementation Plan for a complete list of drivers, some of which are discussed below.

What will happen to the legacy vhost?

All metadata services on legacy vhost `wayf.incommonfederation.org` will be decommissioned on March 29, 2014. At that time, we will install a redirect from the legacy metadata aggregate to the new *fallback metadata aggregate*. That redirect will remain in place indefinitely.

> ⚠ **All deployments should migrate ASAP**
>
> All SAML deployments shall migrate to one of the new metadata aggregates ASAP but **no later than March 29, 2014**.

Is the InCommon metadata signing certificate expiring again?

Yes, traditionally the InCommon metadata signing certificate has been set to expire every three years. It will next expire on May 2, 2014. More importantly, the InCommon metadata signing certificate is signed by a legacy CA whose certificate expires on March 29, 2014. This is why we chose the above migration deadline.

Moving forward, all new metadata aggregates will be signed using a new self-signed signing certificate set to expire on December 18, 2037:

- https://ds.incommon.org/certs/inc-md-cert.pem

In the future, we don't intend to re-sign the new self-signed metadata signing certificate unless it's absolutely necessary.

> **The trusted InCommon metadata signing key will not change**
>
> Although the metadata signing certificate is new, the metadata signing key is not.

Like the vhost issue, if this were the *only* reason for making these changes, we probably wouldn't be doing it. Even *together with* the new vhost, it's debatable whether or not these changes are warranted, but in any case, an emerging SHA-2 requirement is the straw that breaks the camel's back.

What does SHA-2 have to do with metadata?

XML Signature uses a so-called digest algorithm to compute a hash of the signed XML node. The current metadata signing process relies on the SHA-1 digest algorithm. We are updating the process to use SHA-2 instead of SHA-1.

Out of the chute, two of the new metadata aggregates will use different digest algorithms (the third aggregate is for future use):

- The new *production metadata aggregate* will be signed using a SHA-2 digest algorithm (specifically, SHA-256).
- The new *fallback metadata aggregate* will be signed using the SHA-1 digest algorithm (which is what we use now).
- The new *preview metadata aggregate* will be identical to the *production metadata aggregate*.

Why switch from SHA-1 to SHA-2?

Currently the XML signature on InCommon metadata uses the deprecated (and soon-to-be disallowed) SHA-1 digest algorithm:

- NIST deprecated the use of SHA-1 in conjunction with digital signatures on January 1, 2011.
- NIST disallows the use of SHA-1 in conjunction with digital signatures after January 1, 2014.
- See: NIST SP 800-57 Part 1, Revision 3 (July 2012), Tables 3 and 4

This is motivating the migration to a SHA-2 digest algorithm.

> **All deployments should be compatible with SHA-2**
>
> All SAML deployments shall consume metadata signed with a SHA-2 digest algorithm by **June 30, 2014**.

Why are there THREE new metadata aggregates?

Multiple metadata aggregates allow us to deploy changes to InCommon metadata more quickly, easily, and safely. Metadata consumers choose exactly one of the aggregates depending on the immediate requirements of their deployment.

> **Each deployment chooses one aggregate**
>
> Each deployment in the InCommon Federation shall migrate to one (and only one) of the new metadata aggregates no later than **March 29, 2014**.

What is a "fallback metadata aggregate?"

The *fallback metadata aggregate* comes into play when a breaking change is inadvertently introduced into InCommon metadata (such as SHA-2). When a change is made to the production metadata aggregate, and that change breaks a downstream metadata process, an affected deployment can temporarily migrate to the fallback metadata aggregate. This gives the deployment time to adjust to the breaking change.

The fallback metadata aggregate is *transient* in the sense that backward compatibility is provided for a limited, predetermined period of time. In the case of SHA-2, deployments have approximately three months (beyond the March 29 milestone) to become compatible with SHA-2. At that time, the *fallback metadata aggregate* will be synced with the *production metadata aggregate*, which forces all deployments to conform.

> **Eventually all published metadata will be signed with SHA-2**
>
> All SAML deployments shall migrate to the new *production metadata aggregate* (or the *preview metadata aggregate*) ASAP but **no later than June 30, 2014**. From that day forward, all metadata aggregates published by InCommon will be signed using a SHA-2 digest algorithm.

What is a "preview metadata aggregate?"

Like the fallback metadata aggregate, the *preview metadata aggregate* helps manage the introduction of potentially breaking changes into InCommon metadata. Before such a change is deployed in production, it is first introduced in preview mode. Any issues that surface are addressed before the change is moved to production.

The preview metadata aggregate is designed for deployments on the leading edge, such as test and dev deployments. Such deployments are strongly encouraged to consume the preview metadata aggregate instead of the production metadata aggregate.

Choose one: production or preview?

An important decision point for each deployment is whether to migrate to the new *production metadata aggregate* or the *preview metadata aggregate*. Regardless of whether your deployment is compatible with SHA-2, determine the ultimate goal (production or preview) and plan accordingly. Depending on timing, you may have to temporarily migrate to the *fallback metadata aggregate* to reach your ultimate goal.

In the FAQs that follow, where it says "migrate to the *production metadata aggregate*," it should say "migrate to the *production metadata aggregate* or the *preview metadata aggregate*, whatever makes the most sense for your deployment."

Questions about SAML Software

How does this implementation plan affect Shibboleth IdP deployments?

Shibboleth IdP deployments in the InCommon Federation are *least affected* by this implementation plan.

The Shibboleth IdP is SHA-2 compatible

Being a pure Java implementation, the Shibboleth IdP software will verify an XML signature based on a SHA-2 digest algorithm if the underlying version of Java supports SHA-2. Therefore all Shibboleth IdP deployments based on Java version 1.4.2 (or later) should migrate to the new *production metadata aggregate* ASAP (but no later than March 29, 2014).

If you are running Shibboleth IdP version 2.0 (or later), you are almost certainly running Java 1.4.2 (or later). In that case, your software is compatible with SHA-2 and you can migrate to the new *production metadata aggregate* at your convenience (but no later than March 29, 2014).

There are two modifications you need to make to your IdP's metadata configuration:

1. The `metadataURL` XML attribute on the `<MetadataProvider>` element should point to the HTTP location of the new *production metadata aggregate*.
2. Securely download and install a copy of the new metadata signing certificate.

The second step above is optional (since the new signing certificate contains the same key as the old signing certificate) but it is a recommended practice nonetheless. See the Metadata Consumption wiki page for instructions how to securely obtain a copy of the new metadata signing certificate.

How does this implementation plan affect Shibboleth SP deployments?

Shibboleth SP version 2.0 (or later) supports SHA-2 but whether or not it can deliver that support depends on the version of OpenSSL in use by the SP. As a special case, since OpenSSL is bundled with the Shibboleth SP software on the Windows platform, SHA-2 support on Windows is assured.

The Shibboleth SP on the Windows platform is SHA-2 compatible

On the Windows platform, the Shibboleth SP software will verify an XML signature based on a SHA-2 digest algorithm. Therefore Windows-based Shibboleth SP deployments should migrate to the new *production metadata aggregate* (or the *preview metadata aggregate*) ASAP but no later than March 29, 2014.

Since OpenSSL is **not** bundled with the Shibboleth SP software on a non-Windows platform, SHA-2 compatibility depends on the version of OpenSSL in use by the SP.

Old versions of OpenSSL are not compatible with SHA-2

If your deployment depends on an old version of the OpenSSL crypto library, it may be unable to verify the signature on the new metadata aggregate. In particular, versions of OpenSSL prior to 0.9.8 are known to be incompatible with SHA-2 and therefore any platform that depends on OpenSSL 0.9.7 (or earlier) will not be able to verify an XML signature that uses a SHA-2 digest algorithm.

If the Shibboleth SP software is installed on an unsupported OS platform, it is likely you are running an old version of OpenSSL that doesn't support SHA-2. For example, RHEL 4 was built with OpenSSL version 0.9.7, which is known to be incompatible with SHA-2.

On Linux, the Shibboleth SP software depends on whatever version of OpenSSL is built into the underlying operating system, and so it is relatively easy to determine the version of OpenSSL in use:

```
$ /usr/bin/openssl version
OpenSSL 0.9.8y 5 Feb 2013
```

If you're running the Shibboleth SP software on any other operating system (Solaris, Mac OS X), you need to determine the OpenSSL dependency by some other means.

Bottom line: If your Shibboleth SP deployment depends on OpenSSL version 0.9.8 or later, you should migrate to the new *production metadata aggregate* (or the *preview metadata aggregate*); otherwise, you should migrate to the *fallback metadata aggregate*. In either case, there are two modifications you need to make to your SP's metadata configuration:

1. The `url` XML attribute on the `<MetadataProvider>` element should point to the HTTP location of the new metadata aggregate.
2. Securely download and install a copy of the new metadata signing certificate.

The second step above is optional (since the new signing certificate contains the same key as the old signing certificate) but it is a recommended practice nonetheless. See the Metadata Consumption wiki page for instructions how to securely obtain a copy of the new metadata signing certificate.

Deployments not compatible with SHA-2 should upgrade ASAP

If your Shibboleth SP deployment is **not** compatible with SHA-2, and you have to migrate to the *fallback metadata aggregate*, **start planning now** to upgrade your system so that you can migrate to the new *production metadata aggregate* by **June 30, 2014**.

On Linux, since the Shibboleth SP software depends on whatever version of OpenSSL is built into the underlying operating system, you have no choice but to upgrade to a supported platform.

How does this implementation plan affect simpleSAMLphp deployments?

The simpleSAMLphp metarefresh module will refresh and verify metadata automatically. Signature verification depends on the fingerprint of the metadata signing certificate, so the fingerprint configured in the metarefresh module must be updated before migrating simpleSAMLphp to one of the new metadata aggregates.

Old versions of simpleSAMLphp are incompatible with SHA-2

It is known that versions of simpleSAMLphp prior to version 1.11 are **not** compatible with SHA-2. You will need to upgrade to simpleSAMLphp 1.11 (or later) before migrating to the new *production metadata aggregate* (or the *preview metadata aggregate*).

If you're running simpleSAMLphp 1.11 (or later), your software is compatible with SHA-2 and you should migrate to the new *production metadata aggregate* (or the *preview metadata aggregate*); otherwise you should migrate to the *fallback metadata aggregate*. In either case, there are two configuration changes you need to make to simpleSAMLphp's metarefresh module:

1. The `src` array element should point to the HTTP location of the new metadata aggregate.
2. The `validateFingerprint` array element must reflect the fingerprint of the new signing certificate.

The second step is critical for simpleSAMLphp deployments (despite the fact that the new metadata signing certificate contains the same key as the old signing certificate). See the Metadata Consumption wiki page for instructions how to securely obtain the fingerprint of the new metadata signing certificate.

Deployments not compatible with SHA-2 should plan to upgrade

If your simpleSAMLphp deployment is **not** compatible with SHA-2, and you migrate to the *fallback metadata aggregate*, **start planning now** to upgrade your simpleSAMLphp installation and migrate to the new *production metadata aggregate* (or the *preview metadata aggregate*) ASAP but **no later than June 30, 2014**.

Note: simpleSAMLphp 1.12 is due to be released in December 2013. You may want to wait for this release since it includes significant improvements to the metarefresh module.

How does this implementation plan affect Microsoft AD FS 2.0 deployments?

Microsoft AD FS 2.0 is not able to consume the InCommon metadata aggregate, so the answer depends on the external process you use to refresh and verify metadata on behalf of AD FS. Some deployments use a tool called FEMMA, which does not verify the signature on the metadata. In this case the XML signature on the new metadata aggregate will have no adverse effect. An AD FS deployment that uses FEMMA is advised to migrate to the new *production metadata aggregate* (or the *preview metadata aggregate*) ASAP but no later than March 29, 2014.

FEMMA alone does not provide a secure metadata refresh process

If your AD FS deployment uses FEMMA alone to consume InCommon metadata, it is doing so insecurely. A secure metadata process will verify the signature on the metadata **and** check the `validUntil` attribute on the root element of the XML file. FEMMA does neither. See the Metadata Consumption wiki page for more information about the `validUntil` attribute.

An alternative to FEMMA called pysFEMMA *will* verify the signature on the metadata, however. Like the Shibboleth SP, an AD FS deployment that uses pysFEMMA will have difficulty migrating to the new metadata aggregate if the underlying crypto library is incompatible with SHA-2. See the AD FS Metadata Config wiki page for more detailed information about FEMMA and pysFEMMA.

How does this implementation plan affect deployments based on commercial products other than Microsoft AD FS 2.0?

AFAIK, no commercial product is able to consume the InCommon metadata aggregate, so the answer is more-or-less the same as it is for Microsoft AD FS 2.0: it depends on the tool chain used to refresh and verify metadata.

Miscellaneous Questions

What about certificates in metadata? Do I need to re-issue my self-signed certificates in metadata so that they use a SHA-2 digest algorithm?

No, certificates in metadata are not affected by this new policy. Remember, the certificate wrapper on keys in metadata is totally irrelevant. Only the public key bound to the certificate matters.