Upgrading to Shibboleth IdP V3

Upgrading to Shibboleth Identity Provider V3

In May 2015, the Shibboleth Consortium formally announced the end-of-life of Shibboleth IdP V2. A staged end-of-life process began on December 31, 2015 and ended on July 31, 2016.

> **Shibboleth IdP V2 End-of-Life**
>
> Shibboleth IdP V2 reached end-of-life on July 31, 2016. In the future, no bug fixes, not even security-related bug fixes, will be issued.

If you are still running Shibboleth IdP V2, you should be actively planning your upgrade path to Shibboleth IdP V3 or alternative solutions. In preparation for an upgrade, IdP operators in the InCommon Federation should download and install a pre-production instance of Shibboleth IdP V3 as soon as possible.

At any given time, the appropriate version to use will vary but will always be the latest version available from the Shibboleth project.

A test deployment of Shibboleth IdP V3 should mimic your production deployment of V2. Be sure to point your test IdP at the InCommon Preview Aggregate, one of three Metadata Aggregates in the metadata pipeline.

> **Upgrading Your IdP**
>
> For the best possible results:
>
> 1. *Make sure your `entityID` does **not** change*
> 2. *Use a copy of your production SAML signing key*
> 3. *Make sure your SAML protocol endpoints do **not** change*
>
> Avoid making any changes to your metadata, at least until after the migration to V3 is complete. The best plan is to make **no changes to metadata** at all! If you have questions how to do that, contact us at admin@incommon.org

Contents

> **Contribute to the Shibboleth Community Knowledgebase**
>
> Please document your migration experience on the Shibboleth IdP V3 HowTo page in the shibboleth.net wiki.

Getting Started

Spend some time getting your house in order before beginning the upgrade:

1. **Fill all organizational roles**. An InCommon participant in good standing will have a designated *InCommon Executive and two (2) Federation Site Administrators*. Read more about Organizational Roles...
2. **Refresh metadata at least daily**. An interoperable IdP will automatically *refresh and verify metadata at least daily*. An optimal configuration will attempt to *refresh metadata every hour*. Read more about Metadata Consumption...
3. **Remove unnecessary certificates from metadata**. An IdP may support multiple signing keys, but usually an IdP has exactly one such key, and therefore an IdP typically has exactly one signing certificate in metadata. All *extraneous certificates should be removed* from metadata. Read more about Certificates in Metadata...
4. **Expose one (and only one) HTTP-Redirect endpoint**. Every IdP must expose a TLS-protected `<md:SingleSignOnService>` endpoint that supports the *SAML V2.0 HTTP-Redirect binding*. Read more about Endpoints in Metadata...
5. **Re-evaluate SAML protocol support**. A deployment strategy that forces *all protocol traffic over the front channel* is much easier to troubleshoot, manage, and maintain. Read more about Protocol Support for New IdPs...

Know who your SP partners are. In particular, know which SP partners automatically refresh metadata and which don't. SPs that do not automatically refresh metadata make your job more difficult. If you don't know about a partner, it's likely they're relying on your published metadata and so they won't present a significant problem to your migration anyway. It's usually the partners for which you had to manually load metadata yourself that need special care.

If you have questions, by all means raise them on the Shibboleth support list. Many deployers have already gone through complex migrations (from V1 to V2 or from SAML1 to SAML2) and will be happy to advise. Ask before something breaks, not after. If you don't fully understand the implications of a change, ask first.

Basic Migration Strategy

The basic strategy is quite simple: *Use production metadata for testing purposes*. In particular, do **not** introduce a second entity descriptor into InCommon metadata. Such an entity descriptor cannot have the same `entityID` as your production IdP, so by definition that entity descriptor describes a completely new IdP. You generally can't test your existing relationships and integrations with a new IdP. Indeed, you don't **want** a new IdP, you want to upgrade the one you have.

Install Shib IdP V3 on a new physical host (or virtual machine) using the same web path as your production IdP (assuming that path is not going to change). This will allow you to configure Shib IdP V3 to have the same protocol endpoints as your production IdP once your upgrade is complete.

Generally speaking, configure the new V3 instance to be identical to your production IdP (same `entityID`, same SAML signing key and certificate, same endpoints, same metadata sources, same attribute release policy rules, etc.). The two systems should be identical in every way.

In most cases, this is best accomplished by copying your old configuration to the new host as though you were deploying V2 there, and then using the IdP's upgrade support to install the new version on top of the old. This will maintain files that can be maintained to the greatest extent practical.

> **Stabilize eduPersonTargetedID**
>
> If your IdP supports a *computed version of* `eduPersonTargetedID`, don't forget to migrate the secret salt to the new V3 platform (which, per above, will happen automatically if you upgrade). Failure to do so will cause different values of `eduPersonTargetedID` to be generated, which will break interoperability with SP partners. See the FAQ below for more info.

Entity ID in Metadata

The most important piece of advice we can give is: do **not** introduce a new `entityID`.

> **Use Your Production IdP entityID**
>
> Configure Shib IdP V3 to have the same `entityID` as your production IdP so that the two are indistinguishable by relying parties. Consequently, a single entity descriptor in metadata is sufficient to describe both IdPs. Any SP that consumes that metadata will interoperate with either your test IdP or your production IdP.

If you change your `entityID`, you are essentially starting from scratch. In that case, there is no migration to speak of.

Certificates in Metadata

Every IdP has at least one signing certificate (`use="signing"`) in metadata. This certificate contains a public key that complement's the IdP's all-important SAML signing key, the private key used to sign SAML responses and assertions. The latter must be carefully protected at all times.

Most IdPs have exactly one signing certificate in metadata. If you have more than one signing certificate in metadata, it's probably a holdover from a key migration at some point in the past. You should clean this up before getting too deep into the migration by verifying whether the additional certificate(s) are actually in use and for which relying parties.

Of all the bits in metadata, the SAML signing certificate is one thing that **can** be changed without affecting interoperability (at least with SPs that are relying on the metadata). By following a well-defined migration process, you can introduce a new certificate into metadata without negative side effects. As long as your SP partners automatically refresh metadata, you won't have any problems. That said, if you need to migrate a new certificate into metadata, do so at a later time, after you've upgraded to Shib IdP V3.

> **Use Your Production IdP Signing Key for Testing**
>
> If possible, use your production signing key to test Shib IdP V3, but always **handle your private key with care**!

The simplest thing to do is copy the production signing key and certificate files onto the physical machine that hosts Shib IdP V3. Of course you must do this without exposing the private key. In this case, there is no need to update your metadata.

If you are unable to use the production signing key and certificate on the test system, then you'll have to generate a new signing key for testing purposes. The Shibboleth install (but not upgrade) process will do this for you automatically. This new signing key must be kept no less secure than your production signing key. The certificate corresponding to this new signing key may be added to the IdP's entity descriptor in metadata so that there are two certificates in metadata, one for the production IdP and one for the test IdP. See the FAQ below for more info.

Keys and Certificates

Read the Security and Networking topic in the Shibboleth wiki, especially the section on "Keys and Certificates." The Shibboleth IdP installer will generate three pairs of keys and certificates for you automatically but if you copy your production signing key and certificate to the test machine (which is highly RECOMMENDED), you won't need any of them.

Endpoints in Metadata

All browser-facing endpoints in IdP metadata are non-indexed (this is a SAML term, but it simply means that more than one of the exact same type and character can't be used). Consequently, there may be at most one `SingleSignOnService` endpoint per HTTP binding. (The same is true of `SingleLogoutService` endpoints.) At best, adding a second browser-facing endpoint with a redundant binding to metadata serves no purpose.

Changing an endpoint in metadata is even more problematic. While it is possible to migrate to new endpoint locations (see the FAQ below), it is not something to be taken lightly. Depending on your SP partners, it may take months to safely perform such a migration, much like changing a key. It is best to avoid that pain.

Use Your Production IdP Endpoints for Testing

Install the Shibboleth IdP V3 software on a new virtual host but make sure the web application path is exactly the same as your production IdP. Do **not** change the endpoint locations in production metadata.

You are now ready to begin testing.

Testing Strategy

Assuming your test IdP and your production IdP share the same path (on different hosts), your best testing strategy is to *map the DNS name of the production IdP to the IP address of the test IdP using /etc/hosts on a client/browser machine*. Using either SP-initiated or IdP-initiated SSO, systematically test selected partner SPs using the client machine. If the `entityID` and the signing credentials are both unchanged, any problems that occur are a sign of configuration differences that can be investigated and fixed before the cutover occurs.

Avoid back-channel protocols

Except for the back channel, you can fully simulate real transactions against real SPs through simple manipulation of client /etc/hosts files. Back-channel protocols, such as artifact resolution and attribute query, are much more difficult to test. This is one reason why an IdP deployment strategy that forces *all protocol traffic over the front channel* is highly desirable. Avoid back-channel protocols if you can.

Frequently Asked Questions

What if I change my entityID? Will that cause an outage?

The `entityID` is permanent. If you change it, you have in effect removed the old entity and added a new entity. Thus *changing your `entityID` is **not** an option* unless you want to start over. In that case there is no migration to speak of.

What if I change my SAML signing key? Will that cause an outage?

In general, if your SAML signing key is compromised, you should immediately generate a new key pair and introduce the resulting signing certificate into metadata. Yes, this will cause an outage but that's inevitable when your signing key is compromised. Consequently, you should protect your signing key at all costs.

In all other situations, you can migrate a new signing certificate into IdP metadata with relatively little or no pain. However, do not couple this to your software upgrade. First complete the migration to V3 and then migrate a new signing certificate into metadata.

If you absolutely must introduce a new signing key in conjunction with the migration to V3, you can add the corresponding signing certificate to production metadata while the migration is in progress. Any SP that relies on your metadata will accept a response signed by either key. In this way, you can freely drop in and out of test mode regardless of the signing key in use.

What if I change my endpoint locations in metadata? Will that cause an outage?

Yes, changing the endpoint locations in IdP metadata will cause an outage for as long as it takes your SP partners to update their metadata. It will also cause extended outages for any SPs that are not metadata-aware, thereby causing a large amount of disruption and unwanted attention. Don't do this. At the very least, don't use the upgrade as an excuse to do this. It's much harder to change your endpoint locations than to actually upgrade the IdP software, and you don't want to link two complex changes together.

But I really do need to change my endpoints. How do I do this?

If this is a hard requirement, once again, don't couple this to your software upgrade. One does **not** make the other easier. First complete the migration to V3 and then change your endpoints.

To change your endpoints, you need to run two instances of the Shibboleth IdP software on separate servers/hostnames while maintaining single sign-on between them. This allows you to gradually complete a migration from the old hostname to the new hostname by running both in production with traffic mixed between them.

SPs that do not support metadata consumption can be migrated one at a time to the new endpoint(s), usually after direct conversation with the SP owner. At whatever time is convenient, your InCommon metadata can be changed to migrate metadata-consuming SPs en masse to the new endpoint(s). Eventually all SPs will either have moved over, or you'll know there are stragglers requiring special attention using simple log analysis.

In short, endpoints can change, but you do **not** want to perform such a cutover as a weekend activity. It will take weeks, sometimes months to change your endpoints. If you try and shortcut this process, you will have outages.

My IdP supports eduPersonTargetedID. Will upgrading to Shib IdP V3 affect that support?

Implementations of `eduPersonTargetedID` by the Shibboleth software are either stored in a database or computed on the fly. If your IdP supports a *computed version of `eduPersonTargetedID`*, either use the documented upgrade process, or don't forget to migrate the secret salt to the V3 platform. Failure to do so will cause different values of `eduPersonTargetedID` to be generated, which will break interoperability with SP partners.

What about SAML2 Persistent NameID? Is that a concern?

SAML2 Persistent NameID is an alternative syntax to the `eduPersonTargetedID` SAML Attribute, so the question is the same in both cases: Is it stored or computed? If it's computed, you must upgrade properly, or migrate the secret salt, to maintain interoperability.

My IdP supports the R&S category. Will upgrading to Shib IdP V3 affect that support?

No, as long as you don't change your IdP `entityID`, the Research & Scholarship (R&S) entity attribute will remain in your metadata. You should use the upgrade process, or faithfully reproduce the relevant attribute release policy rules in Shib IdP V3 of course, but beyond that there is nothing further you need to do.

I have a network load balancing appliance or strategy. Can this help with my upgrade?

Yes! If you have a network load balancer, hardware or software, or something else that fronts the DNS for your IdP service, it's likely that you can use that facility to ease the transition to V3. This usually involves a step during the final production deployment of your new IdP infrastructure, where you have new IdP node(s) ready for deployment (***following all the critical advice in this document about not changing entityID, endpoint locations or keys***). Once you have your new nodes deployed but not yet in production, find a reasonable time to do the maintenance to put the new nodes into production. You can usually set your load balancing facility to put the old node(s) in 'drain connections' or similar. Start adding in the new IdP nodes and set the old nodes to drain connections. This will allow new traffic to connect to your new IdP node(s) without an interruption in service. ***Do this only after thorough testing of your new infrastructure.***

References

1. Upgrading Shibboleth from V2 to V3
2. Security and Networking for Shibboleth IdP V3
3. InCommon Shibboleth Training materials:
    a. Linux Identity Provider IdPv3
    b. Windows Identity Provider IdPv3
4. Shibboleth IdP 3 Installation Guide (SWITCH)
5. Installing a Shibboleth 3.x IdP (Tuakiri)