Metadata Server

Updated Metadata Host

On Monday, August 7, 2017 @ 3:00 pm EDT, the metadata host in Bloomington, Indiana (140.182.44.53) was permanently decommissioned and a new host in Los Angeles, California (163.253.32.9) has taken its place. Read the full announcement for more information. If you have any questions or concerns, please contact us at admin@incommon.org

Metadata Distribution Server

InCommon metadata is served from vhost `md.incommon.org`, a name that resolves to one of two identical servers, either in Michigan (207.75.165.125) or California (163.253.32.9). Be aware that *the actual server used at any given point in time is unspecified* and left to the discretion of InCommon Operations. If one of the physical servers goes down or requires maintenance, the other can be brought up within minutes, with minimal disruption of services.

Configure your SAML software by name

Ensure both your SAML implementation and your metadata refresh processes are configured with hostname `md.incommon.org` (as opposed to an IP address).

Depending on your environment, you may have to poke a hole in an outbound firewall to allow your metadata client to reach the metadata server. In that case, you will actually want to *poke two holes in your outbound firewall* since there are actually a pair of metadata servers as described above.

Configure your outbound firewall by address

Ensure your outbound firewall (if any) is configured with both IP addresses (207.75.165.125 and 163.253.32.9).

Server Configuration

To facilitate frequent updates, the metadata server supports HTTP Conditional GET, which has important security benefits. For efficiency, the metadata server also supports HTTP Compression, specifically the `gzip` compression algorithm.

Security Considerations

The authenticity and integrity of InCommon metadata is based on document-level security mechanisms. In particular, all metadata files are signed using XML Signature. The signature on downloaded metadata files must be verified before the metadata is trusted.

A trusted metadata refresh process is bootstrapped with an authentic copy of the signer's Metadata Signing Certificate. Using the public key in the certificate, a secure client verifies the signature and validates the expiration date on all downloaded metadata files. Since little (if any) security is provided by downloading metadata via a secure channel, *TLS is not supported on the metadata server*.

Finally, administrative access to the metadata server is strictly controlled.