RedirectGeneration

Web servers have built-in support in various ways for triggering redirections by returning new absolute URLs when resources move. Part of this functionality involves "normalizing" requests so that the scheme (http or https), hostname, port, and path of a request can be determined and used consistently when generating redirects.

When the Shibboleth ServiceProvider software issues AuthnRequest messages, a pair of absolute URLs are constructed representing the requested resource and the eventual AssertionConsumerService URL to return the requested assertion to. In some cases, the ShibbolethXml configuration can supply an absolute URL to use, but in most cases these URLs are constructed dynamically based on the normalized request URL. For this reason, it is **ess ential** that the system be configured with accurate information about the externally visible scheme, port, and hostname on which each virtual host runs.

Problems with redirect generation most frequently occur in accessing the SP's protocol handlers (e.g. `https://gotham.supervillain.edu /Shibboleth.sso/SAML/POST`) and in the final redirect back to the requested resource. Broken redirects most frequently manifest as 404 errors or RedirectLooping (or sometimes simply attempts to access a server over the wrong port). Rather than trying to address each of these in detail, it's more useful to discuss how these redirects are created and what must be properly configured.

Unfortunately, configuring this information is highly non-portable, and sometimes outright impossible. Web servers often claim to support virtual hosting, but this support has limits. It particularly falls apart in proxy configurations in which users access SSL-enabled front-ends to non-SSL web servers. This is **n ot** supported properly by most web servers and is **not** an advisable configuration. Modern servers have the CPU capacity to handle SSL traffic quite well and you will save a lot of time and effort by not needlessly optimizing yourself into a corner. At least, please read this topic before posting questions. If you don't understand it, then please ask yourself if you really need to complicate things by supporting a web server deployment that you don't fully understand.

When possible, Shibboleth pulls most of the information used to normalize these URLs from the web server and the request. The following configuration details highlight how this is done and the exceptions.

Apache-based web servers:

Most of the information needed can be set using standard Apache commands in the configurations that Apache supports.

- UseCanonicalName must be enabled. This forces Apache to return the hostname accessed in the full and proper form rather than a possible variation entered by the user. Failure to do so can result in inaccurate redirects and even exposure of resources if all possible variants are not accounted for in the RequestMap (depending a lot on how session requirements are expressed).
- The external DNS hostname of the web server must be properly stated in a `ServerName` directive either in the main configuration or the proper virtual host. This could be the hostname of a web proxy that is handling the actual requests from users.
- External port numbers are pulled from the `VirtualHost` definition in Apache 2.0 and the `Port` command in Apache 1.3. This is the port users must access, and not necessarily the port the virtual host is actually listening on.
- The correct URL scheme (`http` or `https`) is set based on whether SSL is active on the incoming request. If this isn't accurate, you cannot rely on the web server to properly generate redirects and it must be supplemented by Shibboleth configuration. This will **not** correct redirects from elsewhere in the environment. You can use the `ShibURLScheme https` Apache command to override the value determined from the request if all external access is handled with SSL through an offloaded device.

IIS Web Server:

IIS does not appear to support proper normalization of requests natively. Therefore, ShibbolethXml includes facilities for the routing of incoming requests based on IIS site identifiers and the incoming scheme, port, and hostname can be overridden using it.

- The `<ISAPI>` configuration element in ShibbolethXml defines the mapping from IIS Site/Instance IDs to virtual hosts.
- Each `<Site>` element contains mandatory `id` and `name` attributes specifying the mapping from site number to canonical hostname respectively.
- Optional attributes for `port` , `sslport` and `scheme` can be specified per-site to override the data supplied by the client request.
- SSL-offloading requires that `scheme` and `port` be set to properly normalize requests so that redirects will route back through the SSL front-end. Note that `sslport` is not used in this case (it applies to scenarios where both SSL and non-SSL ports are accessible for one virtual site/host, but not at the ports expected).

Tomcat:

- Unknown at this time (Tomcat is not yet supported as a web server), but the Connector syntax in the Tomcat configuration should include some support for virtualizing the host and specifying hostname or port. It's likely that the same SSL->nonSSL limitation present in Apache will have to be worked around somehow.