IDP Proxy Automation Design Document
-Mrudula Gaidhani

This document is divided into two parts, qatest framework related changes & proposed IDP proxy testcases.

1. qatest framework change:

1. Currently there are two properties in Configurator-server-anme.properties related to multi-protocol hub which refer to two different SP's. The name of those properties are wsfed_sp & idff_sp. These need to be renamed to host1 & host2. In case of multi-protocol hub these will be used as SP's talking different protocols, while in case of samlv2 & idff IDP proxy module they will be used as IDP proxy and IDP discovery service host.
   1. host1
   2. host2
2. IDP proxy details will be added to Samlv2TestConfigData.properties & idffTestConfigData.properties files. This change will be done in ClientConfigCreator.java
   1. IDP proxy protocol
   2. IDP proxy host
   3. IDP proxy port
   4. IDP proxy uri
   5. IDP proxy sp entity name
   6. IDP proxy sp metaalias
   7. IDP proxy idp entity name
   8. IDP proxy idp metaalias
   9. IDP proxy amadmin user name
   10. IDP proxy amadmin password
3. IDP Discovery details will be added too. This change will be done in ClientConfigCreator.java
4. New constants need to be added to TestConstants.java related to IDP proxy, IDP Discovery.

2. SAMLv2IDPProxy & IDFFIDPProxy modules:

New modules SAMLv2IDPProxy & IDFFIDPProxy will be created to add the testcases related to proxy. SAMLv2Common.java, IDFFCommon.java, MultiprotocolCommon.java will be used for common methods which are already defined. New common methods will be defined in the protocol specific common java class.

1. ConfigureUnconfigureIDPProxy.java: Based on the parameter passed through testng xml, this class will perform IDP proxy setup, IDP proxy with Discovery service setup. It will be done in BeforeSuite class called ConfigureUnconfigureIDPProxy.java. Parameter “setupParam” can have values: idpProxy, idpDiscovery. In future we can add new value if needed.
2. testng xml: For samlv2 as well as IDFF the profile value will be sent through testng xml. (Similar to current IDFF testcases). This class will give the flexibility to add testcases without much coding efforts. Also transient & persistance federation value will be sent through this xml.
   1. SSOProfile: artifact, post
   2. SLOProfile: http, soap
   3. TermProfile: http, soap
   4. RegProfile: http, soap (in case of IDFF)
   5. NameIDManagementProfile: http, soap (in case of samlv2)

   6. federationType: Transient, persistence

3. IDPProxyTests.java: This class will contain the SSO, SLO & termination in IDP Proxy setup with profile value picked up from testng xml. Both SP init & IDP init tests will be covered.
4. IDPProxyAutoFedTests.java: This will contain the auto-federation testcases in IDP proxy setup. Same tests will be run with transient, persistence federation.
5. IDPProxyAutoFedDynamicUserTests.java: This class will contain the auto-federation testcases where SP side user profile is set to Dynamic. Same tests will be run with transient, persistence federation. Both SP init & IDP init tests will be covered.
   1. Set user profile to dynamic only at the SP.
   2. Set user profile to dynamic only at the proxy.
   3. Set user profile to dynamic at SP & proxy.
   4. Set user profile to ignore only at the SP.
   5. Set user profile to ignore only at the proxy.
   6. Set user profile to ignore at SP & proxy.
6. SAMLv2IDPProxyRelayStateTests.java: This class will contain the testcases related to relay state supplied in the metadata as well as in the url parameter. We need to make sure that after successful SSO, SLO & termination proper relay state is loaded. Both SP init & IDP init tests will be covered.
   1. Set default Relay state in metadata.
   2. Set Relay State in URL
   3. Set Relay State in metadata & in URL. URL Relay state should take the precedence.
7. IDFFIDPProxyDefaultURLTests.java: This class will contain the testcases related to relay state supplied in the metadata. We need to make sure that after successful SSO, SLO, registration & termination proper relay state is loaded. Both SP init & IDP init tests will be covered.
8. IDFFIDPProxyIsPassiveTests.java: This class will contain the isPassive attribute related tests in IDP proxy setup.
   1. Set isPassive attribute to true at the proxy
   2. Set isPassive attribute to true at the final IDP.
9. IDFFIDPProxyForceAuthNTests.java: This class will contain the forceAuthN attribute related tests in IDP proxy setup.
   1. Set forceAuthN attribute to true at the proxy
   2. Set forceAuthN attribute to true at the final IDP.