# Configuring Apache Knox SSO with Ipsilon using SAML2

**Contributed by Leonardo Dias**

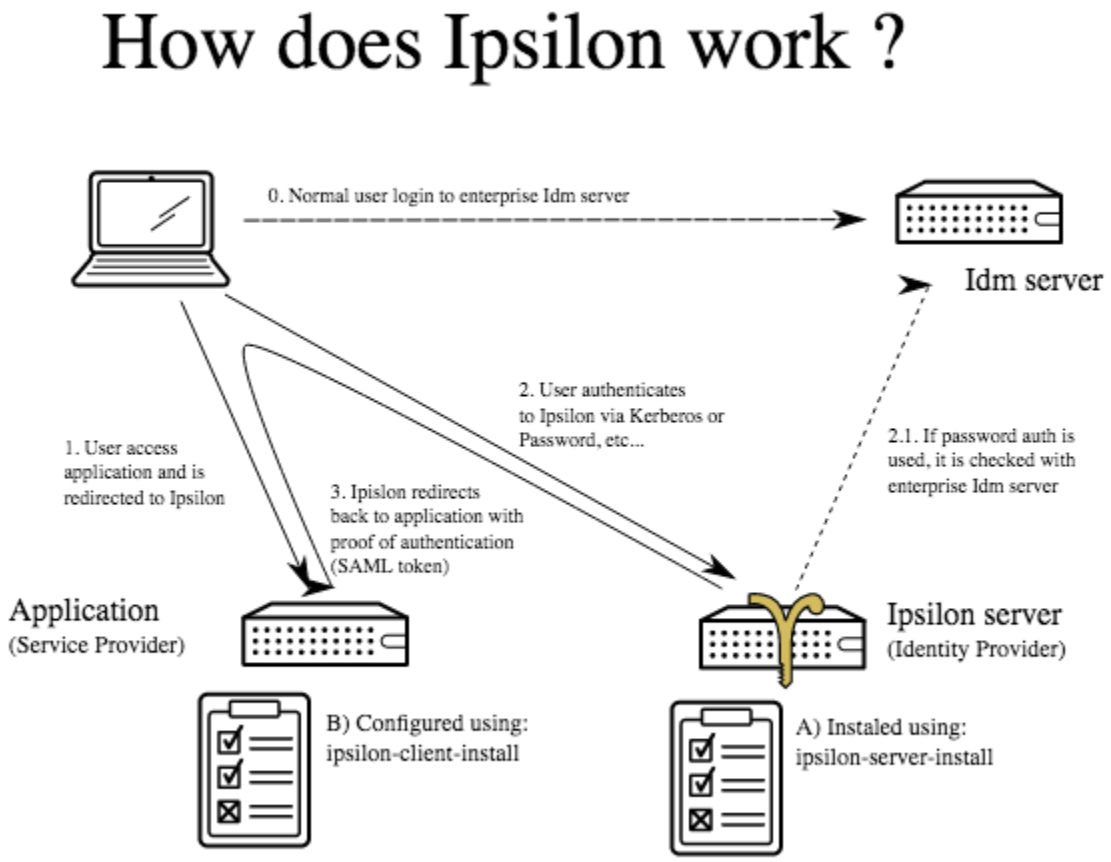
This tutorial will provide the steps to configure and integrate Ipsilon with Knox SSO, using SAML2.

## Ipsilon Overview

Ipsilon is a server and a toolkit to configure Apache-based Service Providers. The server is a pluggable selfcontained mod_wsgi application that provides federated SSO to web applications. User authentication is always performed against a separate Identity Management system (for example a FreeIPA server), and communication with application is done using a federation protocol like SAML, OpenID, etc..

Knox SSO will be a Service Provider of Ipsilon.

The following picture illustrates how Ipsilon works and you can see where KnoxSSO is within the architecture as the Application (Service Provider):




## Requirements

1. Apache Knox installed
2. Apache Ambari Server installed (for Ambari UI SSO)
3. Apache Ranger installed (from Ranger UI SSO)
4. IPA installed and configured
5. Ipsilon binaries already installed on system

## Configuration

1. Configure Ipsilon Server with SAML2 support:

   ```
   ipsilon-server-install --saml2=yes --form=yes --gssapi=yes --ipa=yes  --info-sssd=yes
   ```

2. Patch the Ipsilon Server to fix NameIDPolicy bug (https://pagure.io/ipsilon/pull-request/44). Patch is not included on version 1.0.0 of Ipsilon that can be downloaded from EPEL.

```
From e23eead22c21258c3a0ef22a65f8e1aebc115b77 Mon Sep 17 00:00:00 2001
From: Rob Crittenden <rcritten@redhat.com>
Date: Oct 21 2015 14:52:38 +0000
Subject: Don't crash if no NameIdPolicy is requested


This fixes two problems:

1. Logging was done before a None check was completed
2. The None check was insufficient because the whole object
   could be None

Signed-off-by: Rob Crittenden <rcritten@redhat.com>

https://fedorahosted.org/ipsilon/ticket/189

---

diff --git a/ipsilon/providers/saml2/provider.py b/ipsilon/providers/saml2/provider.py
index 6cbf5ab..6d46ad2 100644
--- a/ipsilon/providers/saml2/provider.py
+++ b/ipsilon/providers/saml2/provider.py
@@ -254,10 +254,12 @@ class ServiceProvider(ServiceProviderConfig):
         self.load_config()

     def get_valid_nameid(self, nip):
-        self.debug('Requested NameId [%s]' % (nip.format,))
-        if nip.format is None:
+        if nip is None or nip.format is None:
+            self.debug('No NameId requested, returning default [%s]'
+                       % SAML2_NAMEID_MAP[self.default_nameid])
             return SAML2_NAMEID_MAP[self.default_nameid]
         else:
+            self.debug('Requested NameId [%s]' % (nip.format,))
             allowed = self.allowed_nameids
             self.debug('Allowed NameIds %s' % (repr(allowed)))
             for nameid in allowed:
```

3. Configure Ipsilon Client on Knox Server. Command is using "ipsilon.example.com" as the Ipsilon Server FQDN.

```
ipsilon-client-install --saml-idp-url https://ipsilon.example.com/idp
 --saml-sp-name knox
 --saml-auth "/gateway/knoxsso/api/v1/websso?pac4jCallback=true&client_name=SAML2Client"
 --saml-no-httpd --saml-sp-post "/gateway/knoxsso/api/v1/websso?pac4jCallback=true&client_name=SAML2Client"
 --saml-sp "/gateway/knoxsso/api/v1/websso?pac4jCallback=true&client_name=SAML2Client"
 --saml-sp-logout="/gateway/knoxsso/api/v1/websso?pac4jCallback=true&client_name=SAML2Client"
 --port 8443 --saml-secure-setup=falseThis command will configure a Service Provider on Ipisilon and
generate three files on the current directory:
 certificate.pem, certificate.key and metadata.xml
```

| Property Name | Description |
|---|---|
| --saml-idp-url | URL for Ipsilon IDP Server |
| --saml-sp-name | Alias to Knox Service Provider |
| --saml-auth | Should match saml.serviceProviderEntityId on KnoxSSO Topology, without server:port |
| --saml-no-httpd | Generate metadata and certificates local and not configure HTTPD as Service Provides (Knox will be SP) |
| --saml-sp-post | Should match saml.serviceProviderEntityId on KnoxSSO Topology, without server:port |
| --saml-sp | Should match saml.serviceProviderEntityId on KnoxSSO Topology, without server:port |
| --saml-sp-logout | Should match saml.serviceProviderEntityId on KnoxSSO Topology, without server:port |
| --port | Knox port |
| --saml-secure-setup | Disable two way SSL |

4. Export IPA/Ipsilon certificate and root certificate to file:

```
openssl x509 -in <(openssl s_client -connect ipsilon.example.com:443 -prexit 2>/dev/null) > ipa.pem
cat /etc/ipa/ca.crt >> ipa.pem
```

5. Import IPA/Ipsilon certificate and root certificate to Java Truststore:

```
$JAVA_HOME/bin/keytool -import -keystore $JAVA_HOME/jre/lib/security/cacerts -file ipa.pem -alias ipa
```

6. Deploy Knox SSO Topology (`/etc/knox/topologies/knoxsso.xml` or use Ambari to configure it). Template below consider Knox Server on `hdp24.example.com`:8443 and Ipsilon Server on `ipsilon.example.com/idp`

```
<topology>
   <gateway>
     <provider>
         <role>federation</role>
         <name>pac4j</name>
         <enabled>true</enabled>
         <param>
               <name>pac4j.callbackUrl</name>
              <value>https://hdp24.example.com:8443/gateway/knoxsso/api/v1/websso</value>
         </param>

         <param>
           <name>clientName</name>
           <value>SAML2Client</value>
         </param>

         <param>
           <name>saml.identityProviderMetadataPath</name>
           <value>https://ipsilon.example.com/idp/saml2/metadata</value>
         </param>

         <param>
            <name>saml.serviceProviderMetadataPath</name>
           <value>/tmp/hdp24.example.com/metadata.xml</value>
         </param>

         <param>
           <name>saml.serviceProviderEntityId</name>
           <value>https://hdp24.example.com:8443/gateway/knoxsso/api/v1/websso?pac4jCallback=true&amp;client_name=
SAML2Client</value>
         </param>
     </provider>
     <provider>
         <role>identity-assertion</role>
         <name>Default</name>
         <enabled>true</enabled>
     </provider>
   </gateway>

   <service>
       <role>KNOXSSO</role>
       <param>
         <name>knoxsso.cookie.secure.only</name>
         <value>false</value>
      </param>
      <param>
        <name>knoxsso.token.ttl</name>
        <value>100000</value>
      </param>
      <param>
         <name>knoxsso.redirect.whitelist.regex</name>
         <value>^https?:\/\/(hdp24\.example\.com|localhost|127\.0\.0\.1|0:0:0:0:0:0:0:1|::1):[0-9].*$</value>
      </param>
   </service>
</topology>
```

Below you can find a description of each parameter set:

| Property Name | Description |
|---|---|
| pac4j.callbackUrl | URL used by pack4j. Should match the knoxsso topology URL |
| clientName | `SAML2Client` is used for SAML2 client on pac4j |
| saml.identityProviderMetadataPath | It's the IDP Metadata URL, on Ipsilon use directory `saml2/metadata` on URL (ex: `https://ipsilon.example.com/idp/saml2/metadata`) |

| | |
|---|---|
| saml.serviceProviderMetadataPath | This parameter is configured to workaround an existing bug, it can be configured to any folder. If you don't configure this entry properly, you will get a NullPointerException on Java |
| saml.serviceProviderEntityId | This is the ID of the Service Provider on Ipsilon, it should be `pac4j.callbackUrl` + `?pac4jCallback=true&client_name=SAML2Client` |
| identity-assertion | Default rule will work. If you want to map some users to different usernames, this is the parameter to be changed |
| knoxsso.cookie.secure.only | `True` if HTTPS is enabled on all URL provided by Knox, otherwise must be `false` |
| knoxsso.token.ttl | Time to live of the cookie in seconds, after this time cookie will be invalid and a new authentication from Ipsilon will be required. |
| knoxsso.redirect.whitelist.regex | Regex that should be matched for Knox to redirect URL to Ipsilon |

8) Extract the Knox Certificate from Gateway Keystore, which will be used on Ambari and Ranger configuration.

```
JAVA_HOME/bin/keytool -export -alias gateway-identity -rfc -file knox.pem -keystore /usr/hdp/current/knox-server
/data/security/keystores/gateway.jks
```

7) Configure Ambari UI for KnoxSSO. On Ambari Server Host run the command:

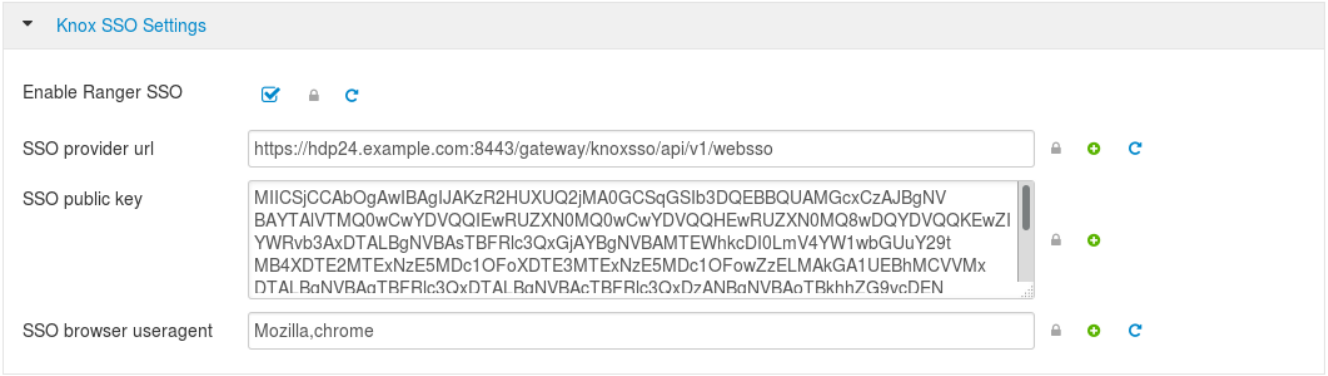
```
root@hdp24 ~]# ambari-server setup-sso
Using python  /usr/bin/python
Setting up SSO authentication properties...
Do you want to configure SSO authentication [y/n] (y)?y
Provider URL [URL] (https://hdp24.example.com:8443/gateway/knoxsso/api/v1/websso):
Public Certificate pem (stored) (empty line to finish input):
MIIDCzCCAfOgAwIBAgIJAOE0SBrVjLOaMA0GCSqGSIb3DQEBCwUAMBwxGjAYBgNV
BAMMEWhkcDI0LmV4YW1wbGUuY29tMB4XDTE2MTExODEzMzIwNloXDTIxMTExNzEz
MzIwNlowHDEaMBgGA1UEAwwRaGRwMjQuZXhhbXBsZS5jb20wggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQDC6bBZAdQPUik1ahpD3kiKpEjJ0L/yZOvbxGPJ
ig8StHR4pSxV7e15blKg5r+LGxvh63tKb/Ju7e66hVto0W2M3phXDR1WhQBhLg+2
UGBypYVQaSibjkmXMXeNx514T+2rXmdcrVuZBCzMf67dsa5lPTWwLjWZh+RlLmhM
iVWfGiN9RRDCmBunCPyZRPE1vqDK8LZVtNTjgPcbMM+Zd9ozegTZhMiyw3YmYu2z
iG29VbOQV52zyv2jQSpXZayNDSWJxKm5g3oSU54PYCV0Psl+YKKbCr7NwAAuih11
hHqIN9k1oWAfrEz56BOuNbbKtWx2hU9Dnrll7kJ72wJewdeBAgMBAAGjUDBOMB0G
A1UdDgQWBBRcmkaQywf/BMuRyDW4wVT5Mv5fQDAfBgNVHSMEGDAWgBRcmkaQywf/
BMuRyDW4wVT5Mv5fQDAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQBd
GwZXq5Es0IzDT9cUeFVBhubdhyn0rMuQckdarRI0iisaCwrSQwBICSVvszckLYcx
fuzhfpjmAtsj5zJ0oT7eZJWNuHznPQMAiAz9BKnarHV21TvSkVMHYOUL2KoiWPxo
2t7SjmfukX5zAxSCeyimOMEhwT8AXJldEHPTCSleDfFz8HPJsB3mwhhf8GHK6V99
YQC2zpRfb+u1SK+EjVXgsdc7Y0pfa08oNgGK/WzvSYaol4ejy55wyG5hcwhVr5hr
QMFGI0Q4MxyU54W6wPu3mm/vNcFSgoyLO9UHG3dUeuEHS/dzG9dfNcNe6MTYquvZ
iQM5c1FKPrAQS5qPZWaR

Do you want to configure advanced properties [y/n] (n) ?
Ambari Server 'setup-sso' completed successfully.
[root@hdp24 ~]#
```

NOTE: Do not paste the Header and Footer of certificate when asked for Public Certificate, which is the Knox Certificate. NOTE: LDAP Authentication is required for SSO to work properly, and need to be configured before setting SSO.

8) Restart Ambari Server to apply new configuration

9 Configure Ranger UI for KnoxSSO. On Ambari Server, go to Ranger -> Configs -> Advanced and set Knox SSO Settings:



NOTE: Do not paste the Header and Footer of certificate when asked for Public Certificate, which is the Knox Certificate.

10) Restart Ranger to apply new configuration

After those steps, all login requests will be redirected to Ipsilon and after authentication, will allow access to both Ranger and Ambari UI.

FINAL NOTES:

- On Ambari Server, to login with local users access the url: http://:8080/#/login/local
- On Ranger, to login with local users access the url: http://:6080/locallogin