Kerberos Configuration on Liberty

Document History

Document revision history

Revision	Date	Author	Summary of Changes
0.2	07/07/18	Jagdish Komakula	First draft version

Document approval history

Name	Signatur	Date	Comments

Contents

1. Introduction

This document describes in detail how to configure single sign-on with an existing deployment of IFE Application. There are many different single sign-on technologies. This document defines Kerberos. Authentication single sign-on setup with workstations that are members of the same Microsoft Active Directory domain. IFE Application uses the users and groups in Active Directory to determine the authorization of users.

Prerequisites :

The instructions assume that you have the following prerequisites installed and accessible:

1. A Microsoft Windows® Server running an Active Directory Domain Controller and associated Kerberos Key Distribution Center (KDC). Example host for such a domain controller is *srv-dc08.ibm.com.* The domain controller name is *ibm.com* and the Kerberos realm name is *IBM.COM,* which is the domain controller name in all uppercase letters.

2. A Microsoft Windows® domain member (client) with a web browser that supports the SPNEGO authentication mechanism. Example host for the client is *myClientMachine.ibm.com*.

3. A working deployment of IFE App, that can be accessed by users in Active Directory. Liberty server host is *srv-t-ife-lib01.ibm.com*.

Intended audience

This document is intended for Enercon IT operation, IBM development team and IBM support team who are familiar with configuring and managing domain controllers, Microsoft Active Directory, and have an understanding of Kerberos Authentication for single sign-on.

1.1 IFE Application with Single Sign-On

IFE application on Liberty Server is currently configured to use user names and passwords that are stored in a file-based registry. By configuring the deployment to use Kerberos single sign-on, a user is logged in through the domain client workstation that they are logged in to.

After a user logs in to a single sign-on environment, they are authenticated with any systems that they have access to. IFE application can be configured to allow authentication through Kerberos single sign-on, with authorization through Active Directory.

Kerberos single sign-on enables users to log in to a Microsoft domain controller, and be authenticated within the single sign-on environment. In Kerberos single sign-on, to change the user that is logged in to IFE, the user must log out of the workstation, and a new user must log in to the workstation.

Configuring IFE application to use Kerberos single sign-on, changes the way that users authenticate with the platform. A deployment that uses Kerberos single sign-on requires the user to access application on a workstation that is a member of the same domain.

Authentication

When IFE application is configured to use Kerberos single sign-on, the authentication sequence between the client and the platform matches the following steps and the associated diagram:

1. The client attempts to connect to WebSphere Application Server Liberty profile with an *HTTP*/*Post*/*Get* request.

2. WebSphere Application Server Liberty profile returns HTTP 401 with a *Negotiate header.*

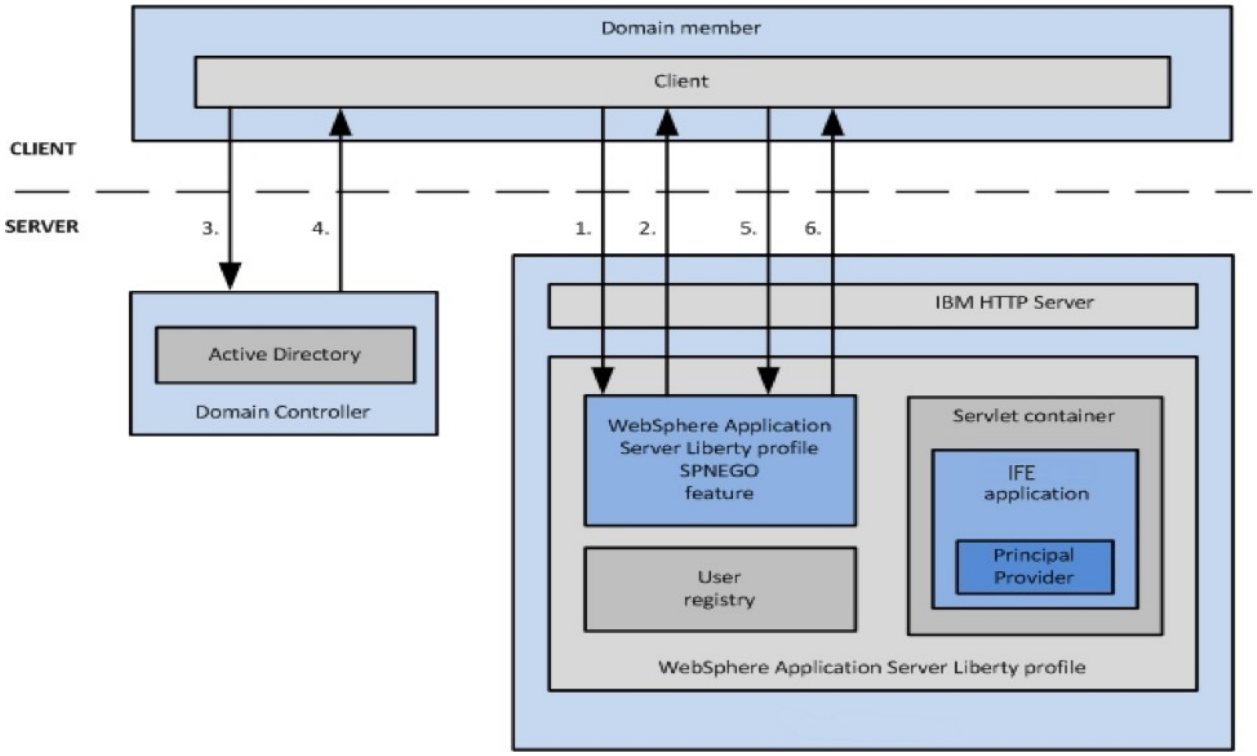
3. The client requests a SPNEGO token from the domain controller.

4. The domain controller returns a SPNEGO token to the client.

5. The client attempts to connect to WebSphere Application Server Liberty profile with an *HTTP*/*Post*/*Get* request and the SPNEGO token.

6. On successful authentication, the client receives a Lightweight Third-Party Authentication (LTPA) token in a cookie.

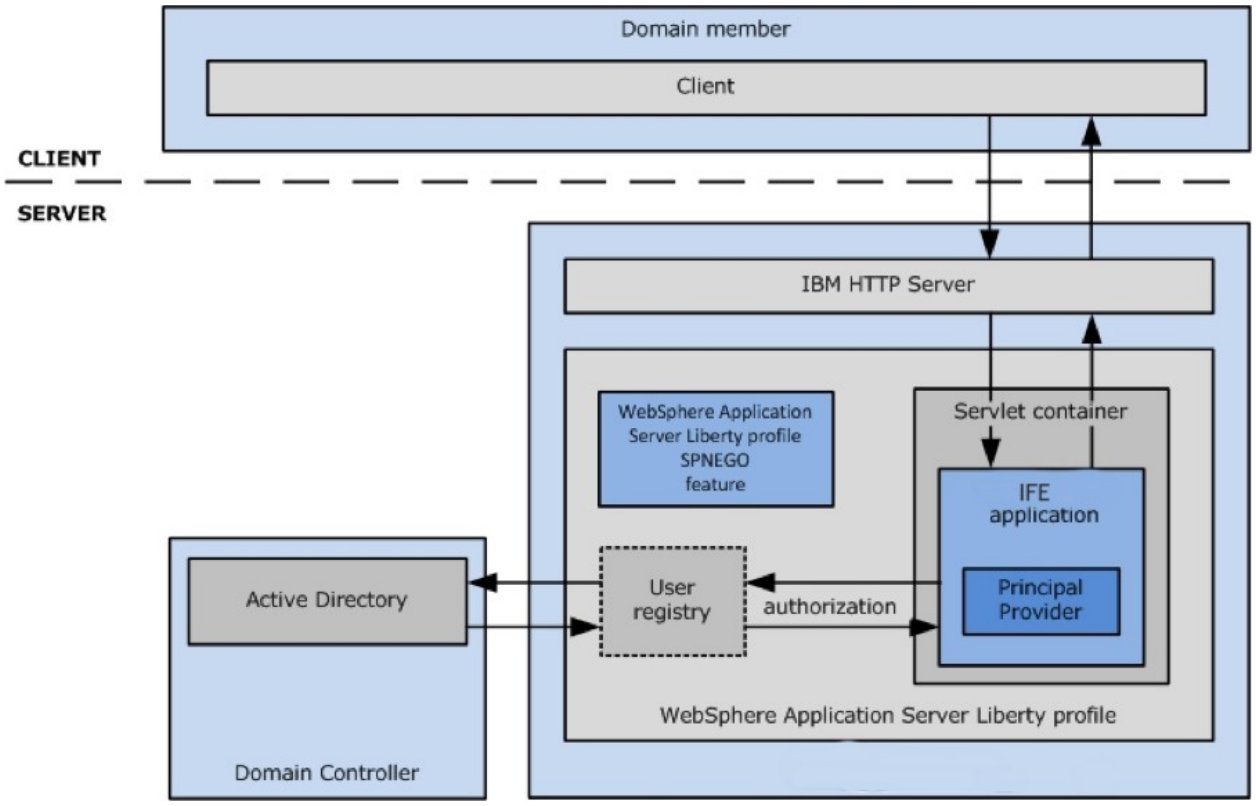
7. After the user is authenticated, they are logged in to IFE Application. To define the data that the user has access to, the user must be authorized by IFE Application.



Authorization

For authorization, the IFE application communicates with Active Directory, through the WebSphere Application Server Liberty profile user registry APIs to retrieve information about the current user. The principal provider then maps the retrieved information to security roles.

The following diagram shows how authorization works in IFE App:


Domain member
Client
CLIENT
SERVER
IBM HTTP Server
WebSphere Application Server Liberty profile SPNEGO feature
Servlet container
IFE application
Principal Provider
Active Directory
User registry
authorization
WebSphere Application Server Liberty profile
Domain Controller

1.2 Implementing Kerberos SSO

Following are the three major steps that needs to be implemented to setup Kerberos Single Sign-On on Liberty Server.

1. Populate Active Directory with the correct users and groups for your environment.
2. Complete the configuration steps for WebSphere Application Server Liberty profile and the deployed IFE application.
3. Redeploy and connect to IFE application from a client workstation within the same domain.

1.2.1 Create Users/Groups in Microsoft Active Directory

The users that are in Microsoft Active Directory are used to authenticate with IFE Application. The groups that are in Active Directory are used for authorization in IFE Application

In a single sign-on setup, the following users must be created in Active Directory:

a) Create user **IFELiberty_http_T** with password as **ibmpassw0rd** for the server that hosts the IFE application, that is later mapped to a Kerberos Service Principal Name (SPN).

b) Create users **IFEAdmin01, IFEUser01, IFEUser02** that are used to log in to IFE App.

To authorize users, the following groups must be created in Active Directory:

a) Create group **IFEUsers**
b) Create group **IFEAdmins** for administrators.

Make each user a member of the respective groups.

a) Add **IFEUser01, IFEUser02** to **IFEUsers** group.
b) Add **IFEAdmin01** to **IFEAdmins** group.

1.2.2 *Configuring WebSphere Application Server Liberty profile*

1) Configure WebSphere Application Server Liberty profile to use Kerberos single sign-on by using the following steps.

   a) Run the Microsoft setspn command to map the user account to a Kerberos SPN. This user account represents the Liberty server as being a Kerberos service with the KDC.

```
C:\> setspn -a HTTP/srv-t-ife-lib01.ibm.com IFELiberty_http

Registered ServicePrincipalNames for
CN=IFELiberty_http_T,OU=IFE_TEST,OU=IFE,OU=Technical,OU=Divisions,DC=ib
m,DC=com: HTTP/srv-t-ife-lib01.ibm.com

Updated object
```

   Note: *Ensure that the host file on the Active Directory server uses the full host name, including the domain name, for the IFE Liberty server. Remove any entries that use only the short name for the IFE Liberty server. The value in the host file must match the value that is used for the SPN.*

   b) Create the Kerberos keytab file by using the Microsoft *ktpass* tool. The default name for this file is *krb5.keytab*.

```
C:\> ktpass -out krb5.keytab -princ HTTP/srv-t-ife-lib01.ibm.com@IBM.COM -mapUser
IFELiberty_http_T -mapOp set -pass passw0rd -crypto AES256-SHA1 -ptype
KRB5_NT_PRINCIPAL

Targeting domain controller: srv-dc06.ibm.com
Using legacy password setting method
Successfully mapped HTTP/srv-t-ife-lib01.ibm.com to IFELiberty_http_T.
Key created.
Output keytab to krb5.keytab:
Keytab version: 0x502
keysize 93 HTTP/srv-t-ife-lib01.ibm.com@IBM.COM ptype 1 (KRB5_NT_PRINCIPAL) vno
6 etype 0x12 (AES256-SHA1) keylength 32
(0x5df7eb84a45d7a9b5726370ecfca64ac9617e01b107cc877232ab938)
```

   c) Make sure that there is not a duplicated SPN in the Microsoft forest by using one of these commands:

```
C:\>setspn -X HTTP/srv-t-ife-lib01.ibm.com

Processing entry 0
found 0 group of duplicate SPNs.
```

d) On the Liberty server machine (srv-t-ife-lib01.ibm.com), enable the Kerberos keytab and configuration files and SPNEGO web authentication.

i) Copy the Kerberos keytab file from the domain controller to the Liberty server machine. The default name of this file is *krb5.keytab* and the default location is */etc/krb5.keytab* same directory as the Kerberos configuration file.
ii) Create a Kerberos configuration file.

The Kerberos configuration file contains client configuration information. This information includes the locations of KDCs for the realms of interest, defaults for the current Kerberos realm, and mappings of host names onto Kerberos realms. For Liberty servers, it must be created manually.

The default location is */etc/krb5.conf*.

Here is a sample Kerberos configuration file (based on the default keytab location):

```
[libdefaults]
   default_realm = IBM.COM
   clockskew = 300
   default_keytab_name = FILE:/etc/krb5.keytab
   default_tgs_enctypes = aes256-cts-hmac-sha1-96
   default_tkt_enctypes = aes256-cts-hmac-sha1-96
   permitted_enctypes = aes256-cts-hmac-sha1-96
   forwardable = false
   proxiable = true
   noaddress = false
   dns_lookup_realm = false
   dns_lookup_kdc = false
   allow_weak_crypto = false
   default_ccache_name = FILE:/tmp/krb5cc_%{uid}

[domain_realms]

[logging]
   kdc = FILE:/var/log/krb5/krb5kdc.log
   admin_server = FILE:/var/log/krb5/kadmind.log
   default = SYSLOG:NOTICE:DAEMON

[domain_realm]
   .ibm.com = IBM.COM

[appdefaults]
   pam = {
      ticket_lifetime = 1d
      renew_lifetime = 1d
      forwardable = true
      proxiable = false
      minimum_uid = 1
      external = sshd
      use_shmem = sshd
   }

[realms]
   IBM.COM = {
      kdc = srv-dc08.ibm.com
      default_domain = ibm.com
      admin_server = srv-dc08.ibm.com
```

```
}
```

e) Verify the Kerberos configuration and keytab files.

You can use the JDK command **klist** to list the SPN in the keytab file.

```
klist -k -t /etc/krb5/krb5.keytab
```

You can use the JDK command **kinit** to validate the SPN in the keytab file and the Kerberos configuration file.

```
kinit -k -t /etc/krb5.keytab HTTP/srv-t-ife-lib01.ibm.com
```

```
root@srv-t-ife-lib01 usr]# kinit -k -t /etc/krb5.keytab HTTP/srv-t-ife-lib01.ibm.com@IBM.COM
Done!
New ticket is stored in cache file /root/krb5cc_root
[root@srv-t-ife-lib01 usr]#
```

After the **kinit** command, you can use the klist command to list the Kerberos ticket. If you get the Kerberos ticket, then the Kerberos keytab and configuration are valid.

f) Configure and enable SPNEGO web authentication on the Liberty server. You can enable SPNEGO web authentication by enabling the *spnego-1.0* feature of Liberty.

i. Add the `spnego-1.0` feature to the server.xml file.

```
<featureManager>
        <feature>spnego-1.0</feature>
        <feature>appSecurity-2.0</feature>
        ...
</featuremanager>
```

2) Configure WebSphere Application Server Liberty profile to use the Microsoft Active Directory registry by using the instructions in Configuring LDAP user registries with Liberty as a reference.

a) Add the `appSecurity-2.0` and `ldapRegistry-3.0` Liberty features to the *server.xml*

b) Add Microsoft Active Directory Server configuration to the *server.xml*

```
<ldapRegistry id="ldap" realm="ADRealm"
    host="server.ibm.com" port="389" ignoreCase="true"
    baseDN="cn=users,dc=server,dc=ibm,dc=com"
    bindDN="cn=testuser,cn=users,dc=server,dc=ibm,dc=com"
    bindPassword="testuserpwd"
    ldapType="Microsoft Active Directory"
    <actived Filters
```

```
            userFilter="(&amp;(sAMAccountName=%v)(objectcategory=user))"
            groupFilter="(&amp;(cn=%v)(objectcategory=group))"
            userIdMap="user:sAMAccountName"
            groupIdMap="*:cn"
           groupMemberIdMap="memberOf:member" >
      </actived Filters>
   </ldapRegistry>
```

c) If AD-LDAP is configured for SSL communication

(i) Imported SSL certificates from AD-LDAP i.e srv-dc08

```
echo -n | openssl s_client -connect 172.16.89.16:636 |
sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > ldapserver.jks
```

(ii) Copy SSL certificates to Liberty Keystore i.e $WLP_HOME/resources/security

```
keytool -importcert -keystore key.jks -file ldapserver.jks
```

(iii) Modify server.xml <ldapRegistry> to include SSL Configuration.

```
<ldapRegistry id="ldap" realm="SampleLdapADRealm"
    host="172.16.89.16" port="636" ignoreCase="true"
    baseDN="DC=ibm,DC=com"
    bindDN="CN=IFELiberty_http_T,OU=IFE_TEST,OU=IFE,OU=Technical,OU=Divisions,DC=ibm,DC=com"
    bindPassword="gQcutA6KKgnab3qapnue"
    ldapType="Microsoft Active Directory"
    sslEnabled="true"
    sslRef="LDAPSSLSettings">
    <actived Filters
    userFilter="(&amp;(sAMAccountName=%v)(objectcategory=user))"
    groupFilter="(&amp;(cn=%v)(objectcategory=group))"
    userIdMap="user:sAMAccountName"
    groupIdMap="*:cn"
    groupMemberIdMap="memberOf:member" />
</ldapRegistry>

<ssl id="LDAPSSLSettings" keyStoreRef="LDAPKeyStore" trustStoreRef="LDAPTrustStore" />

<keyStore id="LDAPKeyStore" location="${server.config.dir}/resources/security/key.jks"
          type="JKS" password="passw0rd" />
<keyStore id="LDAPTrustStore" location="${server.config.dir}/resources/security/key.jks"
          type="JKS" password="passw0rd" />
```

d) By default, all requests to access protected resources use SPNEGO authentication. As the previously deployed IFE App is with basic authentication, you must ensure that the basic registry is not present in the *server.xml* file.

e) **Remove or comment** out the complete *<basicRegistry>* element in the *server.xml* file.

```
<!-- <basicRegistry id="basic" realm="WebRealm">
     <user name="Bob" password="${BobuserPassword}"/>
    <user name="user1" password="${user1userPassword}"/>
    <user name="user2" password="${user2userPassword}"/>
       <group name="admins">
         <member name="Bob"/>
       </group>
       <group name="users">
        <member name="Bob"/>
         <member name="user1"/>
         <member name="user2"/>
       </group>
 </basicRegistry> -->
```

1.2.3 *Testing IFE Application with Kerberos single sign-on on the client workstation*

You must be logged in to the client workstation as one of the users in the domain controller, who is in at least one group per security dimension in the IFE security schema.

Your web browser must be configured on the client workstation before accessing the IFE application

Configuration for Chrome :

a. Add the *--auth-server-whitelis*t parameter to the google-chrome command.
b. For example, to run Chrome from a Linux prompt, run the google-chrome command as follows:

> google-chrome --auth-server-whitelist = "*.ibm.com"

Now Log in to the client workstation as users with different access levels. For each user, complete the steps to demonstrate that authorization is working correctly when you are using Kerberos Authentication for single sign-on.

1.3 Troubleshooting

Enable logging in Liberty server

Add the following line to server.xml and look for trace.log in the logs folder

```
<logging traceSpecification="com.ibm.ws.wim.*=all:com.ibm.ws.security.*=all::com.ibm.ws.webcontainer.security.*=all" />
```

2. Issues and Resolution

2.1 Firewall Rules

Liberty Server should have access to KDC on TCP/UDP Port 88 (Bi-directional)
Liberty Server should have access to LDAPS on TCP Port 636 (Bi-directional)

Client should have access to KDC on TCP/UDP Port 88
Client should have access to Liberty Server on TCP Port 9080, 9443

2.2 Invalid Key Size

```
[jkomakula@srv-t-ife-lib01 bin]$ ./kinit IFEUser02_T
Password for IFEUser02_T@IBM.COM:

com.ibm.security.krb5.internal.crypto.KrbCryptoException, status code: 0
     message: java.security.InvalidKeyException: Illegal key size
```

Solution: Copied JCE jars to JAVA_HOME/lib/security

https://www.ibm.com/support/knowledgecenter/en/SSZJPZ_11.7.0/com.ibm.swg.im.iis.found.admin.common.doc/topics/lmt_scr_downloading_installing_jce_policyfiles.html

2.3 Clock skew too great

[root@srv-t-ife-lib01 bin]# ./kinit IFEUser02_T
Password for IFEUser02_T@IBM.COM:

com.ibm.security.krb5.KrbException, status code: 37
     message: Clock skew too great

Solution : The time difference between the Liberty Single Sign-On server and the Key Distribution Center (KDC) (or ActiveDirectory domain controller) is too great. Normally, the time difference should not be great than 5 minutes.

Need to enable NTP or synchronize system's time (srv-t-ife-lib01) to the domain controller (srv-dc05.ibm.com)

2.4 Pre-authentication information was invalid

```
[root@srv-t-ife-lib01 etc]# kinit IFELiberty_http_T
Password for IFELiberty_http_T@IBM.COM:

com.ibm.security.krb5.KrbException, status code: 24
     message: Pre-authentication information was invalid


Solution: Password for the user might be changed. Reset the Password.
```

2.5 Client not found in database

```
[root@srv-t-ife-lib01 ~]# kinit -k -t /etc/srv-t-ife-lib01.keytab HTTP/srv-t-ife-
lib01.ibm.com@IBM.COM
com.ibm.security.krb5.KrbException, status code: 6
       message: Client not found in Kerberos database

Solution:
```

Check in your KDC to make sure that you only have one SPN and not multiple?
setspn -X HTTP/srv-t-ife-lib01.ibm.com@IBM.COM

Fix your key tab (regenerate one) that has the only SPN you plan on using.
Use AES256 crypto as recommended.

ktpass -out krb5.keytab -princ HTTP/srv-t-ife-lib01.ibm.com@IBM.COM -mapUser `IFELiberty_http_T` -mapOp
set -pass passw0rd -crypto AES256-SHA1 -ptype KRB5_NT_PRINCIPAL

3. Kerberos Configuration in a Clustered Environment

1. Integrate your clustered environment with MS Active Directory. It is simply.

2. Restart and check domain's users authorization.

3. Create users for linking to Kerberos service principal name (SPN).
You must create one user for any node and one user for your Web front server.
Thereby, for CE with two nodes you must create three users!

3. Create keytab file for each user. Use different names of servers.
Remember, one server - one keytab - one SPN!

4. Copy all our keytabs to some Liberty server.

5. Unite our keytabs using the command:

    ktab -m <keytab file1> <keytab file2>

Repeat the command for uniting all the keytabs.

6. You must copy your united keytab and krb5.conf in the same folders in each of your servers.
The folder must have the same path for all your servers!
In my case, /etc/krb5.conf and /etc/*.keytab