Step-by-Step guide to Configure Single sign-on for HTTP requests using SPNEGO web authentication

Summary

Abstract

This guide want to explain how install, and configure, Security Directory server to synchronize user Password between AD 2012 and IBM Portal 8.5 Credential Vault.

IBM WebSphere Portal Server 8.5
Red Hat Enterprise Linux 6.0 update 3
DB2 10.5
Active Directory 2012 R2 mixed mode
IBM HTTP Server 8.0
Security Directory Integrator 7.2
Security Directory Server 6.3.1

Windows/Unix Differences

This guide was written using Linux as the base operating system, however the steps/concepts listed in this guide are independent of operating system.

The only significant difference is that for Windows, you must use the batch file commands instead of the UNIX shell commands listed in this guide.

For example:

UNIX: ./startServer.sh WebSphere_Portal
Windows: startServer.bat WebSphere_Portal

Or

UNIX: ./ConfigEngine.sh cluster-node-config-cluster-setup
Windows: ConfigEngine.bat cluster-node-config-cluster-setup

Hostnames Used in this Guide

To avoid confusion with my own hostnames, I've replaced each instance of the hostnames of my Servers with a sample value that corresponds to the server it belongs to so that it may be easier to understand which server I'm referring to in my examples.
I use the following values:

Database Server:	dbstore.ondemand.com
LDAP Server:	ldap.ondemand.com
IBM HTTP Server:	portal.ondemand.com
SDI Server:	sdi.ondemand.com

Main Guide

Pre check

Verify have more then 5GB on temporary directory /tmp

Open terminal and verify if your system is reachable using fully qualified hostname

[root@serv01 /]# ping first.ondemand.com

In the same terminal, execute

[root@serv01 /]# ping localhost

To verify the "localhost" network settings are configured properly on your machine.

Linux/UNIX environments only.
If in your environment do not use IPV6 verify that is disable in each machine.

In the same terminal, execute

[root@serv01 /]# cat /etc/sysconfig/network

And verify if your NETWORKING_IPV6 is set to "no"

Ensure have sufficient file open limit, is set to 10240 or higher.

ulimit -n 10240

Web Content Manager only: Complete the following steps to remove any file size limits: Use the ulimit -f command to set the maximum size of files that can be created.

Following library is needed during installation process, if you do not configure X environment verify you can use export display to use each wizard, in this guide I use this method to execute installation.

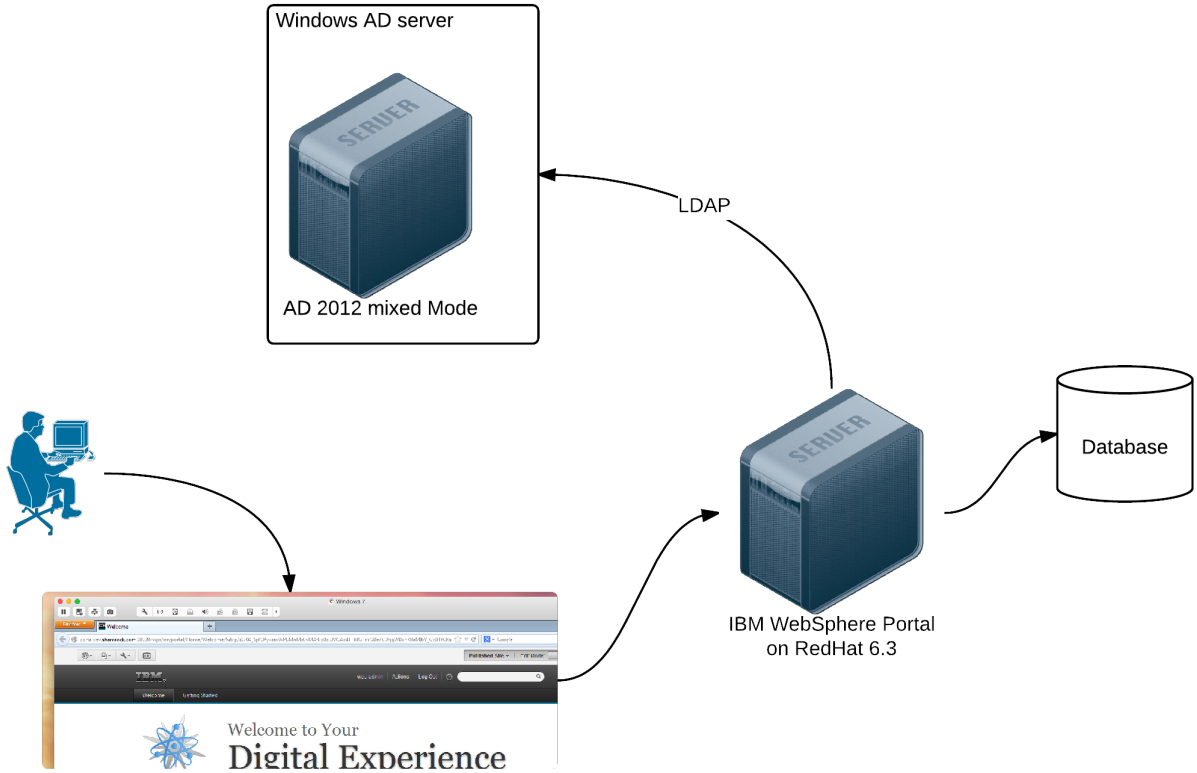
gtk2-2.18.9-6.el6.x86_64.rpm
glib2-2.22.5-6.el6.x86_64.rpm
libXtst-1.0.99.2-3.el6.x86_64.rpm
compat-libstdc++-33-3.2.3-69.el6.x86_64.rpm
openmotif22-2.2.3-19.el6.x86_64.rpm
pam-1.1.1-10.el6.x86_64.rpm
libXp-1.0.0-15.1.el6.x86_64.rpm
libXmu-1.0.5-1.el6.x86_64.rpm
kernel-headers-2.6.18-238.19.1.el5.x86_64.rpm
compat-glibc-headers-2.3.4-2.26.x86_64.rpm
compat-glibc-2.3.4-2.26.x86_64.rpm
libgtk-x11-2.0.so.0
libgtk-x11-2.0.so.0
libcanberra-gtk-module.so
glibc-2.12-1.47.el6.i686.rpm

compat-libstdc++-33-3.2.3-69.el6.x86_64.rpm
compat-libstdc++-33-3.2.3-69.el6.i686.rpm
yum search -1.0.0-15.1.el6.i686.rpm
libXp-1.0.0-15.1.el6.x86_64.rpm
openmotif-2.3.3-4.el6.i686.rpm
xterm
xkeyboard-config
tigervnc-server-1.0.90-0.17.20110314svn4359.el6.x86_64.rpm
xorg-x11-twm-1.0.3-5.1.el6.x86_64.rpm
xorg-x11-font*

Architectural Scenario

In this scenario, we have one AD, and ours Portal Environment.

The idea is: when user open portal page using his browser the can come in without insert his credential because WebSphere Portal will be configure to accept SPNEGO authentication.



Single sign-on for HTTP requests using SPNEGO web authentication

What is SPNEGO?

SPNEGO is a standard specification that is defined in The Simple and Protected GSS-API Negotiation Mechanism (IETF RFC 2478).

The authentication of HTTP requests is triggered by the user (the client-side), which generates an SPNEGO token. WebSphere Application Server receives this token. Specifically, the SPNEGO web authentication decodes and retrieves the user identity from the SPNEGO token. The identity is then used to make authorization decisions.

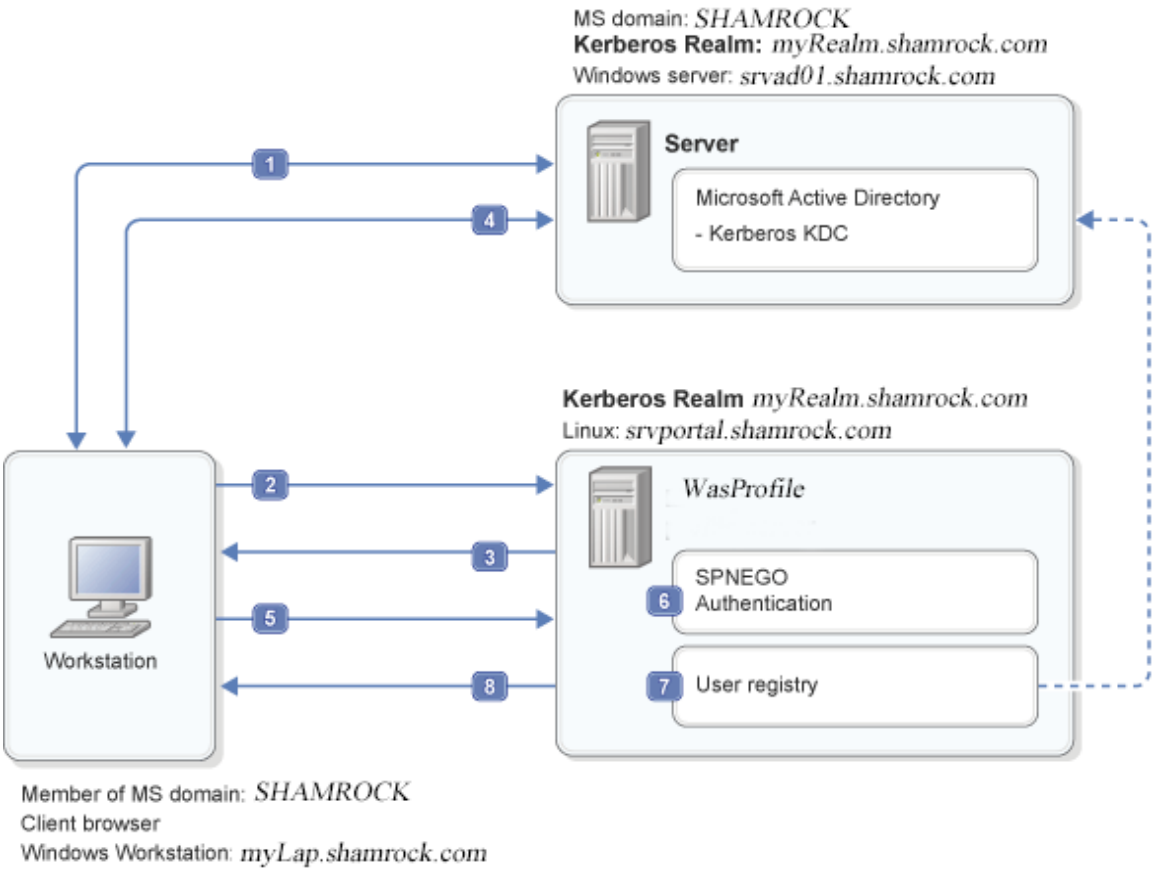
SPNEGO web authentication is a server-side solution in the WebSphere Application Server. Client-side applications are responsible for generating the SPNEGO token for use by SPNEGO web authentication. The user identity in the WebSphere Application Server security registry must be identical to the identity that the SPNEGO web authentication retrieves. An identical match does occur when Microsoft Windows Active Directory server is the Lightweight Directory Access Protocol (LDAP) server that is used in WebSphere Application Server. A custom login module is available as a plug-in to support custom mapping of the identity from the Active Directory to the WebSphere Application Server security registry.

WebSphere Application Server validates the identity against its security registry. If the validation is successful, the client GSS delegation credential is retrieved and placed in the client subject, and a Lightweight Third Party Authentication (LTPA) security token is created. It then returns the LTPA cookie to the user in the HTTP response. Subsequent HTTP requests from this same user to access more protected resources in the WebSphere Application Server use the LTPA security token that is previously created to avoid repeated login challenges.

SPNEGO web authentication in a single Kerberos realm

SPNEGO web authentication is supported in a single Kerberos realm (domain). The challenge-response handshake process is shown in the following figure:

Figure 1. SPNEGO web authentication in a single Kerberos realm



In the previous figure, the following events occur:

1. To begin, the user logs on to the Microsoft domain controller `SHAMROCK` from the workstation.
2. Next, the user attempts to access the Web application. The user requests a protected Web resource using a client browser, which sends an `HTTP GET` request to the WAS profile server.
3. SPNEGO authentication in the WAS profile server answers the client browser with an `HTTP 401` challenge header that contains the `Authenticate: Negotiate` status.
4. The client browser recognizes the negotiate header because the client browser is configured to support integrated Windows authentication. The client parses the requested URL for the host name. The client uses the host name to form the target Kerberos service principal name (SPN) `HTTP/srvPortal.shamrock.com` to request a Kerberos service ticket from the Kerberos ticket-granting service (`TGS`) in the Microsoft Kerberos KDC (`TGS_REQ`). The `TGS` then issues a Kerberos service ticket (`TGS_REP`) to the client. The Kerberos service ticket (SPNEGO token) proves both the user's identity and permissions to the service (Liberty profile server).

5. The client browser then responds to the WAS profile server Authenticate: Negotiate challenge with the SPNEGO token that is obtained in the previous step in the request HTTP header.
6. SPNEGO authentication in the Was profile server sees the HTTP header with the SPNEGO token, validates the SPNEGO token, and gets the identity (principal) of the user.
7. After the WAS profile server gets the identity of the user, it validates the user in its user registry and performs the authorization checks.
8. If access is granted, the Liberty profile server sends the response with an `HTTP 200`. The WAS profile server also includes an LTPA cookie in the response. This LTPA cookie is used for subsequent requests.

Configure Windows 2012 for extend SSO to WebSphere

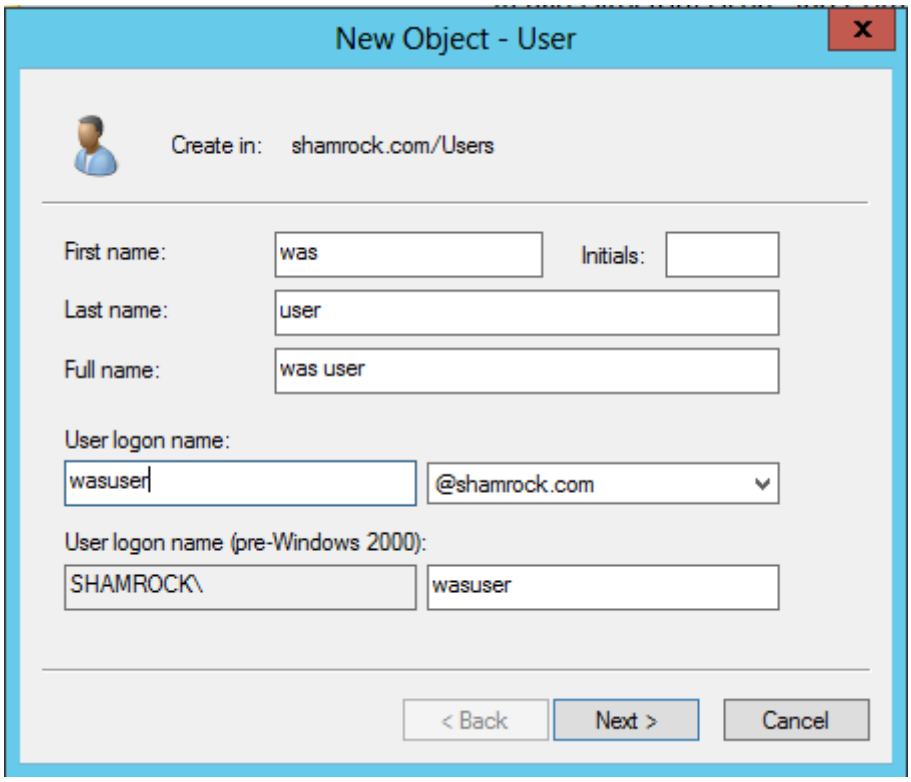
In this section, you will applying each configuration on your Windows AD to permit Spnego SSO with WebSphere

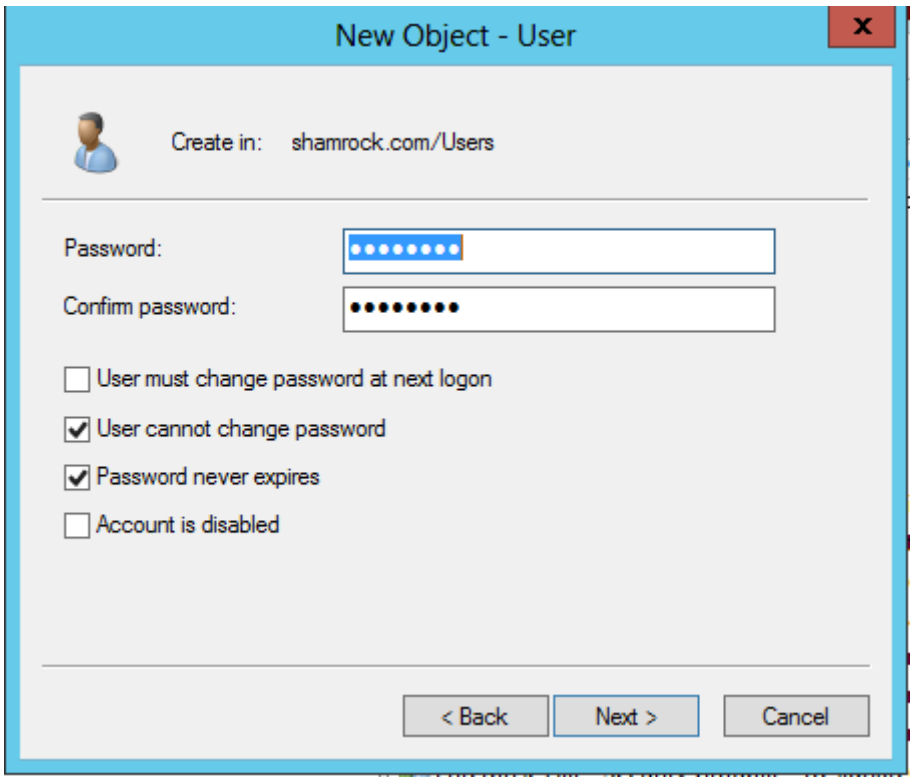
This task is performed on the active directory domain controller machine. Complete the following steps to ensure that the Windows 2012 Server that is running the active directory domain controller is configured properly to the associated key distribution center (KDC).

1. Create a user account in the Microsoft Active Directory for the WebSphere Application Server.

   Click **Start->Programs->Administrative Tools->Active Directory Users and Computers**.

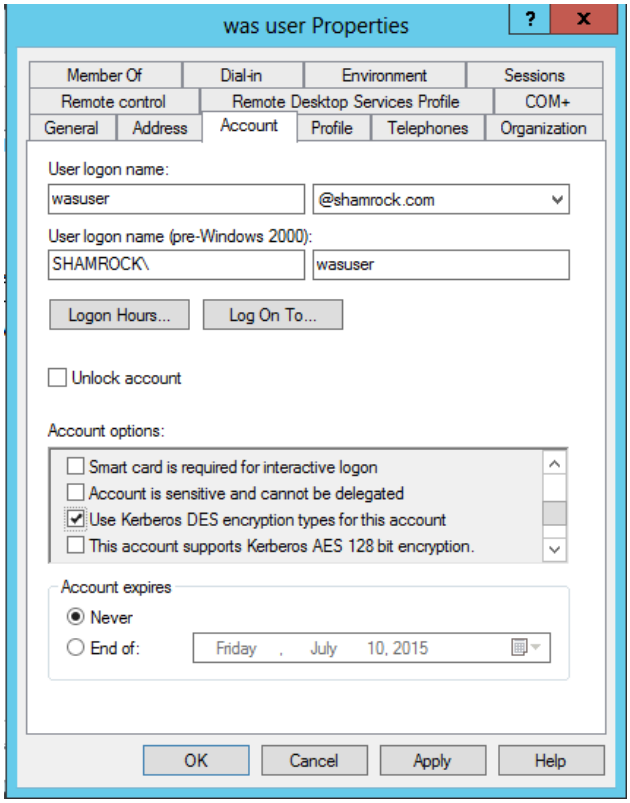
   Use the name for WebSphere Application Server. For example, if the application server you are running on the WebSphere Application Server machine is called `srvPortal.shamrock.com`, create a new user in an active directory called `WasUser`.





Insert your password, and flag “User cannont change password” and “Password never expired”

Now we must add another Account parameter, choose user properties and Account tab, in Account options sections select “User Kerberos DES encryption types for this account”



And choose OK

Make sure that you do not have the computer name `wasuser` under Computers and Domain Controllers. If you already have a computer named `wasuser`, then you must create a different user account name.

- Click **Start -> Programs -> Administrative Tools -> Active Directory Users and Computers->Computers**.
- Click **Programs -> Administrative Tools -> Active Directory Users and Computers->Domain Controllers**.

Register Kerberos service principal name

Use the **setspn** command to map the Kerberos service principal name, `<service name>/<fully qualified host name>`, to a Microsoft user account.

The service name for SPNEGO web authentication must be HTTP. However, the service name for Kerberos authentication can be any strings that are allowed by the KDC.

An example of the **setspn** command usage for SPNEGO web authentication is as follows:

From \windows\System32\ run

setspn -A HTTP/portaldev.shamrock.com wasuser

to register link between user/server

to verify your setting use

setspn –l wasuser

```
C:\Windows\System32>setspn -l wasuser
Registered ServicePrincipalNames for CN=was user,CN=Users,DC=shamrock,DC=com:
        HTTP/portaldev.shamrock.com

C:\Windows\System32>_
```

Note: The host name must be a fully-qualified host name.

Important: Make sure that you do not have the same service principle names (SPNs) mapping to more than one Microsoft user account. If you map the same SPN to more than one user account, the web browser client can send an NT LAN manager (NTLM) token instead of a SPNEGO token to WebSphere Application Server.

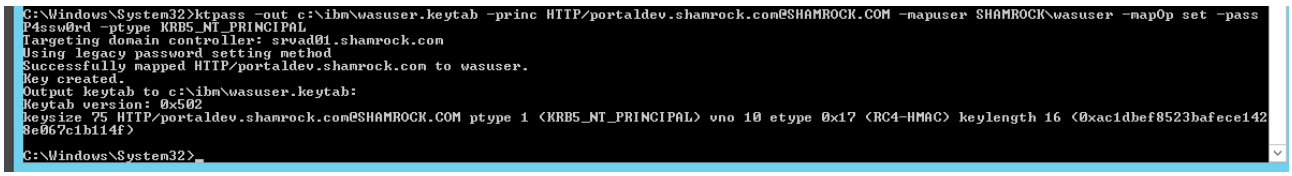
Create the Kerberos keytab file

Create the Kerberos keytab file and make it available to WebSphere Application Server. Use the **ktpass** tool from the Windows Server toolkit to create the Kerberos keytab file (krb5.keytab) for the SPN.
Note: A Kerberos keytab file contains a list of keys that are analogous to user passwords. It is important for hosts to protect their Kerberos keytab files by storing them on the local disk.

Use the **ktpass** tool from the Windows Server toolkit to create the Kerberos keytab file for the service principal name (SPN). Use the latest version of the **ktpass** tool that matches the Windows server level that you are using.

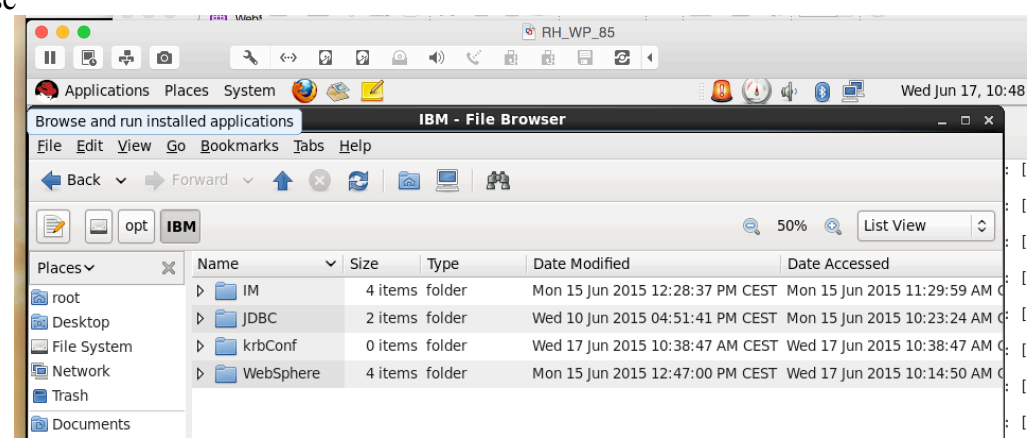
To determine the appropriate parameter values for the **ktpass** tool, run the `ktpass -?` command from the command line. This command lists whether the **ktpass** tool, which corresponds to the particular operating system, uses the `-crypto RC4-HMAC` or `-crypto RC4-HMAC-NT` parameter value. To avoid warning messages from the toolkit, you must specify the `-ptype KRB5_NT_PRINCIPAL` parameter value.

ktpass -out c:\ibm\wasuser.keytab -princ HTTP/portaldev.shamrock.com@SHAMROCK.COM -mapuser SHAMROCK\wasuser -mapOp set -pass P4ssw0rd -ptype KRB5_NT_PRINCIPAL



Configure your WAS to accept Kerberos and SPNEGO Authentication

Loggin in your Was Server and create a directory where will copy each configuration file
In my case



/opt/ibm/krbConf

copy from your Windows server your keytab file and ini file

Now create the Kerberos configuration file, this file contains some information including the location of KDC for realms of iterest. Use wsadmin utility to create a Kerberos configuration file for your WAS.

Go to <WAS_Home>/bin and run wsadmin.sh

```
[root@localhost bin]# ./wsadmin.sh -lang jython -user waslocal -password P4ssw0rd
```

to get help for createKrbConfigFile command use

print AdminTask.help('createKrbConfigFile')

```
wsadmin>print AdminTask.help('createKrbConfigFile')
WASX8006I: Detailed help for command: createKrbConfigFile

Description: This command creates a Kerberos configuration file (krb5.ini or krb5.conf).

Target object:   None

Arguments:
  *krbPath - Supply directory location and file name of the configuration (krb5.ini or krb5.conf) file.
  *realm - Supply Kerberos realm name.
  *kdcHost - Supply host name of the Kerberos Key Distribution Center.
  kdcPort - Supply port number of the Kerberos Key Distribution Center (default: 88).
  *dns - Supply the Domain Name Service (DNS).
  *keytabPath - Supply directory location and file name of the Kerberos keytab file.
  encryption - Supply encryption type (default: rc4-hmac des-cbc-md5).

Steps:
  None


wsadmin>
```

in ours case use:

AdminTask.createKrbConfigFile('[-krbPath /opt/ibm/krbConf -realm SHAMROCK.COM -kdcHost srvad01.shamrock.com -dns srvad01.shamrock.com -keytabPath /opt/IBM/krbConf/wasuser.keytab]')

If you prefer cna use command in interactive mode running :

AdminTask.createKrbConfigFile(['-interactive'])

```
root@localhost:/opt/IBM/WebSphere/AppServer/bin
File  Edit  View  Search  Terminal  Help
wsadmin>AdminTask.createKrbConfigFile(['-interactive'])
Create Kerberos configuration file

This command creates a Kerberos configuration file (krb5.ini or krb5.conf).

*Filesystem location of the Kerberos configuration file (krbPath): /opt/ibm/krbConf
*Kerberos realm name in Kerberos configuration file (realm): SHAMROCK.COM
*Host name of the Kerberos Key Distribution Center (kdcHost): srvad01.shamrock.com
Port number of the Kerberos Key Distribution Center (kdcPort):
*A list of the Domain Name Service, seperated by a pipe character (austin.ibm.com|raleigh.ibm.com) (dns): srvad01.shamrock.co
m
*Filesystem location of the keytab file (keytabPath): /opt/ibm/krbConf/wasuser.keytab
Encryption type (encryption):

Create Kerberos configuration file

F (Finish)
C (Cancel)

Select [F, C]: [F] F
WASX7278I: Generated command line: AdminTask.createKrbConfigFile('[-krbPath /opt/ibm/krbConf -realm SHAMROCK.COM -kdcHost srv
ad01.shamrock.com -dns srvad01.shamrock.com -keytabPath /opt/ibm/krbConf/wasuser.keytab]')
'/opt/ibm/krbConf has been created.'
wsadmin>
```

the result will be

[libdefaults]

default_realm = SHAMROCK.COM

default_keytab_name = FILE:/opt/IBM/krbConf/wasuser.keytab

default_tkt_enctypes = rc4-hmac des-cbc-md5

```
        default_tgs_enctypes = rc4-hmac des-cbc-md5
        forwardable  = true
        renewable  = true
        noaddresses = true
        clockskew  = 300
[realms]
        SHAMROCK.COM = {
                kdc = srvad01.shamrock.com:88
                default_domain = srvad01.shamrock.com
        }
[domain_realm]
        .srvad01.shamrock.com = SHAMROCK.COM
```

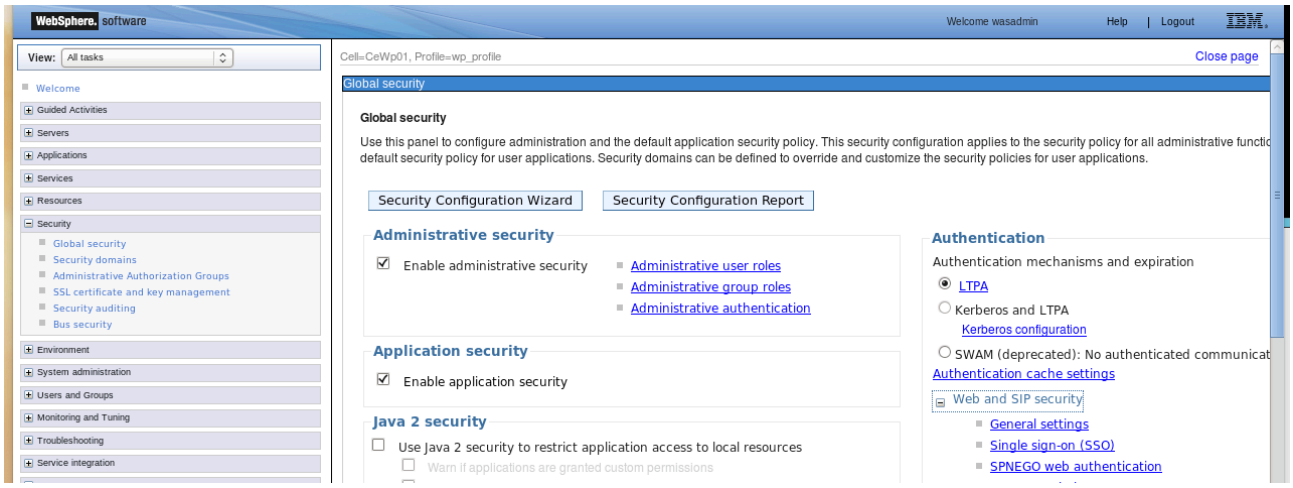
set the file permission to 644 !

chmod –R 664 /opt/IBM/krbConf

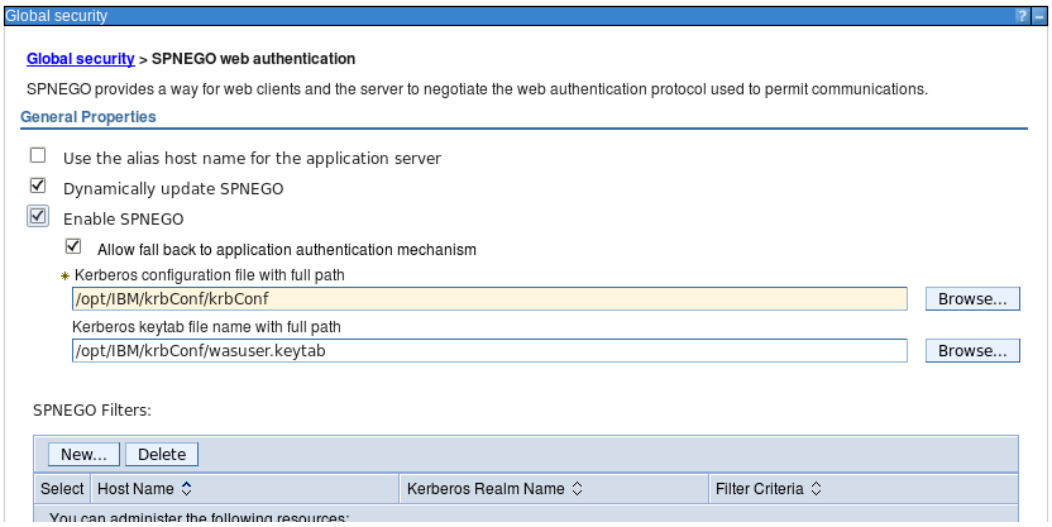
```
[root@localhost krbConf]# ll
total 8
-rw-rw-r-- 1 root root 427 Jun 17 11:32 krbConf
-rw-rw-r-- 1 root root  77 Jun 10 13:01 wasuser.keytab
[root@localhost krbConf]#
```

Configure SPNEGO as the authentication mechanism using ICS Console

Login in your ICS and navigate to **Security > Global Security** from Authentication section, expand **Web and SIP Security** and clik **SPNEGO web Authentication**

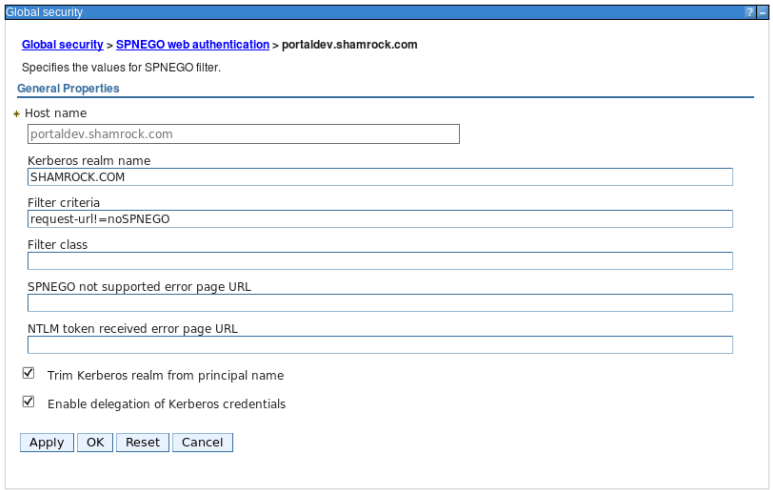


and populate filed with your value



set Dynamically update SPNEGO to true
set Enable SPNEGO to true
set Allow fail back….. to true

and create SPNEGO Filter, clik new



insert HostName with your FQDN
insert Kerberos Realm and remember Kerberos Realm Name must be Capital Letter, it's case sensitive
insert Filter Criteria in my case *request-url!=noSPNEGO*

in SPNEGO not supported….and in NTLM Token…. You can map your courtesy page in case of error, you must map full uri eg. http://portaldev.shamrock.com/error/sp.html or nt.html….

Set Trim Kerberos realm from principal name to true
Set Enable delegation of kerberos credentials to true
Apply and save

Restart jvm

courtesy page sample:

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.0 Transitional//EN">
```

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html">
<html>
 <head>
  <script language="javascript">
    var origUrl=""+document.location;
    if (origUrl.indexOf("noSPNEGO")<0) {
       if (origUrl.indexOf('?')>=0) origUrl+="&noSPNEGO";
       else origUrl+="?noSPNEGO";     }

function redirTimer() {
        self.setTimeout("self.location.href=origUrl;",0);     }
</script>
<META HTTP-EQUIV = "Pragma" CONTENT="no-cache">
<script language="javascript">
document.write("<title> Redirect to "+origUrl+ " </title>");
</script>
</head>
<body onLoad="redirTimer()"/>
</html>
```

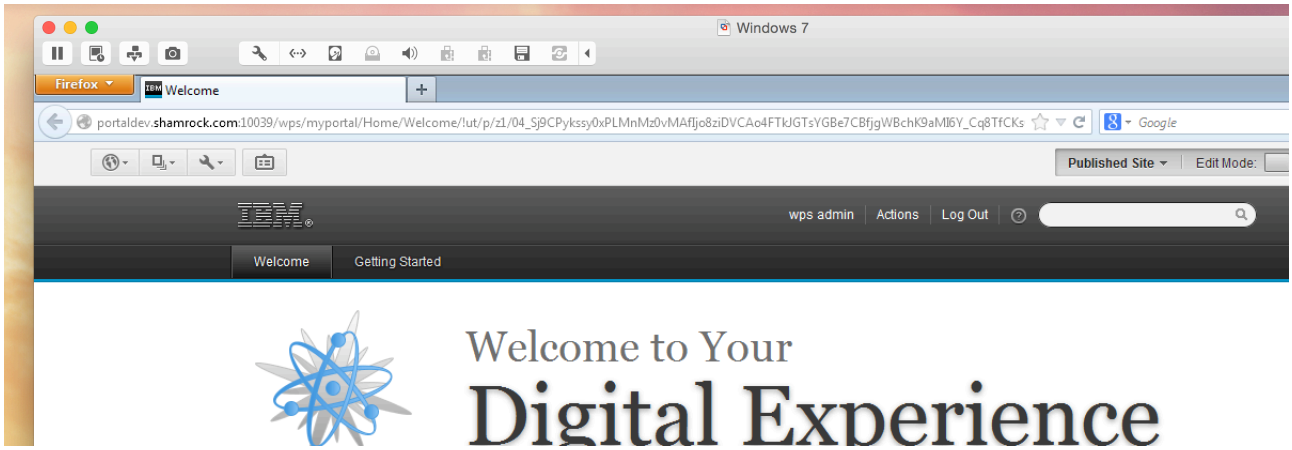
Configuring web browsers to support SPNEGO

Now you must configure your browser to accept SPNEGO authentication, and when you call a private url you can comein in SSO with windows, but only if you are authenticated to Windows domain.

Do one of the following:

- Microsoft Internet Explorer:
  a. From the Internet Explorer menu, select Tools > Internet Options and then click the Security tab.
  b. Click the Local intranet icon and then click Sites.
  c. Click Advanced and then add the web address of the host name of your IBM Connections server into the Add this website to the zone field. For example: *.enterprise.example.com. Click Add.
  d. Enter the host name of your IBM HTTP Server into the Add this website to the zone field and click Add. For example: http://*<IHS_host>* or https://*IHS_host>*".
  e. Click OK to save the change and return to the main Security page.
  f. Click Custom level, scroll to find User Authentication > Logon, and select Automatic logon only in Intranet zone. Click OK to save the change and return to the main Security page.
  g. Click the Advanced tab, scroll to find Security, and then select the Enable Integrated Windows Authentication check box. Click OK to save the change.
  h. Restart the web browser to apply the configuration changes.
- Mozilla Firefox:
  a. Open Firefox and type `about:config` into the location bar.
  b. Type `network.n` into the Filter field and double-click network.negotiate-auth.trusted-uris.
  c. Enter the address of the server that hosts IBM Connections, for example, `enterprise.example.com`.
  d. Click OK to save the change.
  e. If the deployed SPNEGO solution is using the advanced Kerberos application of Credential Delegation, double-click network.negotiate-auth.delegation-uris. This preference defines the sites for which the browser can delegate user authorization to the server. Enter a comma-delimited list of trusted domains or URLs.
  f. Restart Firefox to apply the configuration change.

Now if you call private url like http://protaldev.shamrock.com/wps/**my**portal you can entrer in SSO



Behind the scenes you can see in your logs file:

```
[6/17/15 23:17:30:206 CEST] 00000001 WsServerImpl  A   CWWSR0001I: Server WebSphere_Portal open for e-business
[6/17/15 23:18:00:337 CEST] 000000d3 ServerCredent I com.ibm.ws.security.spnego.ServerCredentialsFactory initializeServer CWSPN0016I: Ready to process host: portaldev.s
hamrock.com.
[6/17/15 23:18:00:337 CEST] 000000d3 TrustAssociat I com.ibm.ws.security.spnego.TrustAssociationInterceptorImpl initialize CWSPN0006I: SPNEGO Trust Association Intercep
tor initialization is complete. Configuration follows:
        SPNEGO Web Authentication:
        enabled = true
        dynamically update = true
        allowAppAuthMethodFallback = true
        krb5Config = /opt/IBM/krbConf/krbConf
        krb5Keytab = /opt/IBM/krbConf/wasuser.keytab
        Server configuration:
        Kerberos ServicePrincipalName=HTTP/portaldev.shamrock.com@SHAMROCK.COM
        com.ibm.ws.security.spnego.SPN.filter=request-url!=noSPNEGO
        com.ibm.ws.security.spnego.SPN.filterClass=com.ibm.ws.security.spnego.HTTPHeaderFilter@879b9b68
        com.ibm.ws.security.spnego.SPN.NTLMTokenReceivedPage=null
        com.ibm.ws.security.spnego.SPN.spnegoNotSupportedPage=null
        cannonicalSupport=false
[6/17/15 23:18:00:397 CEST] 000000d3 SpnegoHandler W com.ibm.ws.security.spnego.SpnegoHandler handleRequest CWSPN0021E: No delegated credentials were found for user: wp
sadmin@SHAMROCK.COM.
[6/17/15 23:18:01:077 CEST] 000000d3 authz         I   CWWIM2000I Initialization of the authorization component completed successfully.
[6/17/15 23:18:01:116 CEST] 000000d3 ServerCache   I   CWWDY1001I: WebSphere Dynamic Cache instance named ws/WSSecureMapNotShared initialized successfully.
[6/17/15 23:18:01:117 CEST] 000000d3 ServerCache   I   CWWDY1071I: The cache provider "default" is being used.
[6/17/15 23:18:01:169 CEST] 000000d3 ServerCache   I   CWWDY1001I: WebSphere Dynamic Cache instance named ws/com.ibm.wps.devicesupport.client2deviceclass initialized su
ccessfully.
[6/17/15 23:18:01:169 CEST] 000000d3 ServerCache   I   CWWDY1071I: The cache provider "default" is being used.
```

Author:

Andrea Fontana
IBM Champion for WebSphere on 2012, 2013, and 2014
IBM Champion for Collaborative Solution on 2015

DevloperWorks Contributor Author

Can be contacted at: a.fontana@net2action.com
afontana@sowre.com