DOAG Vortrag über

# Interoperabilität zwischen Oracle WSM 11g und Microsoft WCF/.NET 4.0

| | |
|---|---|
| erstellt am: | 08.09.2014 |
| letzte Änderung: | 18.09.2014 |
| Version: | 1.0 |

# Inhalt

# 1 Einleitung

## 1.1 Ausgangssituation

Ein vorhandener Proxy Service im OSB (Oracle Service Bus) soll von einem Windows Client genutzt werden. Dabei soll zwischen dem Client und dem Service eine sichere Authentifizierung stattfinden. Hierfür soll die Anmeldung am Windows-Rechner genutzt werden (SSO – Single Sign On).

Dieses Tutorial basiert auf einem Projekt der SIV.AG. Die SIV.AG hat für Ihre Kunden ein Outlook Add In entwickelt, dass sich nahtlos in Outlook einbettet. Wird auf eine E-Mail geklickt, deren Absender in kVASy bekannt ist, dann werden zu diesem Absender Kundendaten angezeigt. Der Sachbearbeiter kann sich damit einen schnellen Überblick über den Kunden und seine letzten Aktivitäten machen.

kVASy ist das Hauptprodukt der SIV.AG und wird bei ca. 300 Kunden in der Energie- und Wasserwirtschaft eingesetzt. Siehe auch http://www.siv.de/ für weitere Informationen zur SIV.AG und ihren Produkten und Dienstleistungen.

Das kVASy Outlook Add In wird in diesem Tutorial nicht weiter behandelt. Stattdessen wird der Windows Client an Hand eines WCF .Net C#-Konsolenprogramms erklärt.

## 1.2 Zielsetzung

Das Windows-Konsolen-Programm soll die Windows-Anmeldung des Benutzers nutzen. Diese sollen dem Webservice in einer sicheren Art und Weise übertragen werden. Der Webservice soll den Benutzer überprüfen und im Fall einer unberechtigten Nutzung ablehnen.

## 1.3 Lösungsbeschreibung

Das Windows-Konsolen-Programm ist eine .NET Applikation (.NET Framework 4.5), die in der Programmiersprache C# erstellt wurde. Das .NET Framework bietet die Möglichkeit auf die Windows-Anmeldung über das Kerberosprotokoll zugreifen zu können. Die Anmeldedaten werden dabei in Form eines Kerberos-Tickets zur Verfügung gestellt. Dieses Ticket kann dem Webservice übergeben werden. Der Oracle Service Bus (OSB) ist, unter Verwendung von Oracle Web Service Manager (OWSM) Kerberos Policies, in der Lage diese Tickets zu verarbeiten und die Authentifizierung des Benutzers durchzuführen.

## 1.4 Was ist Kerberos

Für eine kurze Einführung siehe [12]. Eine genauere Beschreibung wie Kerberos in Windows implementiert ist, findet sich in [7]. Eine sehr verständliche Beschreibung findet sich in [13]. Und zum Schluss mein Favorit: Ein 10minütiges Video auf Youtube [15].

## 1.5 Beteiligte Systeme

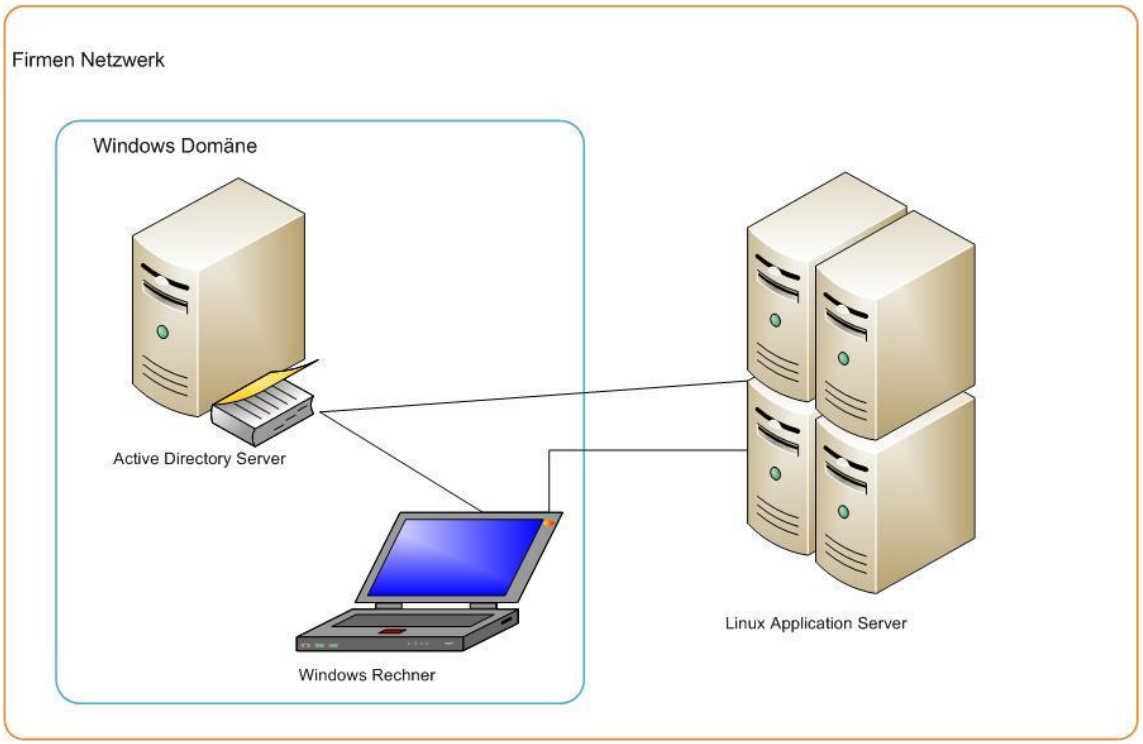



*Abbildung 1*

Abbildung 1 zeigt die beteiligten System:

1. Windows Rechner mit installierten .Net Client der auf einen Webservice auf dem Linux Applikationen-Server zugreifen will.
2. Linux-Applikation-Server mit installiertem Oracle OSB- , SOA-Server, kVASy5-Datenbank und kVASy5-Geschäftsanwendung.
3. Active Directory Server

Alle Systeme befinden sich in einem Netzwerk. Der Windows-Rechner und der ADS befinden sich in einer Windows-Domäne. Der Windows Rechner ist in der Windows Domäne registriert. Der Benutzer der Windows .Net Anwendung meldet sich an der Windows Domäne an und ist als User im Active Directory registriert.

# 2 Umsetzung

## 2.1 Voraussetzungen

### 2.1.1 Zeitabgleich

Die Uhren vom ADS, MS-Windows-Client und Linux-Server müssen synchronisiert sein.

### 2.1.2 Sicherung anlegen

Es wird empfohlen vor den Änderungen eine Sicherung der Systeme anzulegen.

Während der Arbeiten wird empfohlen immer eine Sicherung der betroffenen config-Dateien zu machen.

- /u01/app/oracle/Middleware/user_projects/domains/prod/config/config.xml
- /u01/app/oracle/Middleware/user_projects/domains/prod/config/fmwconfig/jps-config-jse.xml
- /u01/app/oracle/Middleware/user_projects/domains/prod/config/fmwconfig/jps-config.xml

### 2.1.3 Windows WCF .Net Applikation (kVASy Outlook Add In)

In einer .Net Entwicklungsumgebung wird ein simples WCF C# Konsolenprogramm erzeugt. Die für uns wichtigen Bestandteile sind die Datei Program.cs und App.config. Program.cs enthält den auszuführenden Programm-Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace KerberosConsoleClient
{
   class Program
   {
      static void Main(string[] args)
      {

         try
         {
            CustomerTaskServiceV1Client client = new CustomerTaskServiceV1Client();
            var titles = client.GetAssignmentTitle(new GetAssignmentTitle());
            Console.WriteLine("Result:");
            titles.ToList().ForEach(x => Console.WriteLine(x));

         }
         catch (Exception ex)
         {
```

```
            Console.WriteLine("{0}: {1}\r\n{2}", ex.GetType().Name, ex.Message, ex.StackTrace);
        }
        Console.WriteLine("press <ENTER> to exit");
        Console.ReadLine();
    }
  }
}
```

Erklärung des Programmablaufs:

Es wird ein Webservice Client initialisiert. Der ruft die Webservice Operation GetAssignmentTitle auf. Die Titel werden dann in einer Liste ausgegeben.

Die Datei App.config enthält die Konfiguration des Programms:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
    </startup>

  <system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="CustomerTaskServiceV1SOAPtest" />
      </basicHttpBinding>
      <customBinding>
        <binding name="CustomerTaskServiceV1SOAP">
          <!--Added by User: Begin-->
          <security defaultAlgorithmSuite="Basic128"
           authenticationMode="Kerberos"
           requireDerivedKeys="false" securityHeaderLayout="Lax"
           includeTimestamp="true"
           keyEntropyMode="CombinedEntropy"
           messageProtectionOrder="SignBeforeEncrypt"
```

```
messageSecurityVersion="WSSecurity11WSTrustFebruary2005WSSecureConversationFebruary20
05WSSecurityPolicy11BasicSecurityProfile10"
        requireSignatureConfirmation="false">
         <localClientSettings cacheCookies="true" detectReplays="true"
            replayCacheSize="900000" maxClockSkew="00:05:00"
            maxCookieCachingTime="Infinite"
            replayWindow="00:05:00"
            sessionKeyRenewalInterval="10:00:00"
            sessionKeyRolloverInterval="00:05:00"
            reconnectTransportOnFailure="true"
            timestampValidityDuration="00:05:00"
            cookieRenewalThresholdPercentage="60" />
         <localServiceSettings detectReplays="true"
            issuedCookieLifetime="10:00:00"
            maxStatefulNegotiations="128" replayCacheSize="900000"
            maxClockSkew="00:05:00"
            negotiationTimeout="00:01:00" replayWindow="00:05:00"
            inactivityTimeout="00:02:00"
            sessionKeyRenewalInterval="15:00:00"
            sessionKeyRolloverInterval="00:05:00"
            reconnectTransportOnFailure="true"
            maxPendingSessions="128"
            maxCachedCookies="1000"
            timestampValidityDuration="00:05:00" />
         <secureConversationBootstrap />
        </security>
        <!--Added by User: End-->
        <textMessageEncoding maxReadPoolSize="64"
          maxWritePoolSize="16"
           messageVersion="Soap11" writeEncoding="utf-8">
         <readerQuotas maxDepth="32" maxStringContentLength="8192"
          maxArrayLength="16384"
            maxBytesPerRead="4096" maxNameTableCharCount="16384"
/>
```

```
        </textMessageEncoding>
        <!--Added by User: Begin-->
        <httpTransport manualAddressing="false"
            maxBufferPoolSize="524288"
             maxReceivedMessageSize="65536" allowCookies="false"
             authenticationScheme="Anonymous"
             bypassProxyOnLocal="false"
             hostNameComparisonMode="StrongWildcard"
             keepAliveEnabled="true" maxBufferSize="65536"
             proxyAuthenticationScheme="Anonymous"
             realm="" transferMode="Buffered"
             unsafeConnectionNtlmAuthentication="false"
             useDefaultWebProxy="true" />
        <!--Added by User: End-->
      </binding>
    </customBinding>
  </bindings>
  <client>
    <endpoint address="http:///[host name].siv.de:8011/[path]/customerTaskService/V1"
       binding="customBinding" bindingConfiguration="CustomerTaskServiceV1SOAP"
       contract="CustomerTaskServiceV1" name="CustomerTaskServiceV1SOAPQSPort" >
      <identity>
        <servicePrincipalName value ="HTTP/[host name].siv.de@SIV.DE"/>
      </identity>
    </endpoint>
  </client>
 </system.serviceModel>
</configuration>
```

Die Konfiguration eines .Net Clients (Kerberos mit Message Protection) wird von Oracle in [6] ausführlich beschrieben. Die roten Zeilen in der App.config zeigen, wo wir Anpassungen für unsere Umgebung vorgenommen haben.

Als Betriebssystem wird Windows 7 Professional Service Pack 1 64 Bit verwendet.

### 2.1.4 Oracle Fusion Middleware (FMW), SOA Suite und Oracle Service Bus (OSB)

Oracle Linux Server release 6.

OSB Server 11.1.1.7

SOA Suite 11.1.1.7

**Java-Version der Weblogic-Server:**

java version "1.7.0_17"

**Active Directory Authentication provider**

Dieser muss in der Oracle WebLogic Server Administration Console konfiguriert sein.

1. Console aufrufen

*Beispiel-URL*: http://[host]:7001/console

2. Zur Seite Authentication Providers navigieren

- Auf der linken Seite in der Box Domain Structure den Menüpunkt Security Realms wählen

- unter myrealm auswählen

- den Tab Providers wählen

Der Provider muss an erster Stelle in der Liste stehen und das Control Flag des Providers muss auf SUFFICIENT stehen.

## 2.2 Anlegen des Active Directory Host Account User

Für die Kerberos-Konfiguration auf dem Linux-Server muss im AD ein Host Account User angelegt, konfiguriert und eine keytab-Datei erstellt werden. Folgende Schritte sind dazu notwendig:

### 2.2.1 Anlegen des Host Account Users

Siehe auch [1].

1. Der Name des Users kann frei gewählt werden.

2. Überprüfen, ob für diesen Account schon ein SPN existiert.

setspn –L <account>

Wenn ein SPN existiert, diesen löschen:

setspn –D <SPN> <account>

3. Wenn AES Encryption benutzt wird, z.B. wenn Encryption Typ auf All gesetzt wird:

Eigenschaften des Accounts aufrufen und im Account Reiter folgende Checkboxen markieren:

- The account supports Kerberos AES 128 bit encryption
- The account supports Kerberos AES 256 bit encryption

Die “Do not require Kerberos preauthentication box” nicht markieren.

Den OK Button betätigen.

### 2.2.2 Generieren der keytab-Datei

Im Folgenden werden die Befehle mit Beispieldaten aufgelistet. Siehe auch [1].

1. keytab-Datei generieren

Der Befehl muss auf dem AD Server ausgeführt werden.

*Befehl:*

ktpass -princ HTTP/<host name.domain>@<AD domain> -pass <password> -mapuser <account name> -out <keytab name> -ptype KRB5_NT_PRINCIPAL -kvno 0 -crypto <ALL or RC4-HMAC-NT>

Hier ist sehr wichtig, dass die AD domain immer groß geschrieben wird.

*Beispiel (mit Konsolenausgabe):*

c:\>ktpass -princ HTTP/[host].siv.de@SIV.DE -pass welcome -mapuser [user]@SIV.DE -out test.keytab -ptype KRB5_NT_PRINCIPAL -crypto ALL

Targeting domain controller: ADS01.siv.de

Successfully mapped HTTP/[host].siv.de to soatestuser.

Password succesfully set!

Key created.

Key created.

Key created.

Key created.

Key created.

Output keytab to test.keytab:

Keytab version: 0x502

keysize 64 HTTP/[host].siv.de@SIV.DE ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x1 (DES-CBC-CRC) keylength 8 (0xfbcd0ba2f7867068)

keysize 64 HTTP/[host].siv.de@SIV.DE ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x3 (DES-CBC-MD5) keylength 8 (0xfbcd0ba2f7867068)

keysize 72 HTTP/[host].siv.de@SIV.DE ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x17 (RC4-HMAC) keylength 16 (0x42ef711ec1d99ef24107549b5893b9af)

keysize 88 HTTP/[host].siv.de@SIV.DE ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x12 (AES256-SHA1) keylength 32 (0x440fe6feef24f3eaa565e7b9adf667e9b6cedcfe42d2898c4c1d4956b9d0a94a)

keysize 72 HTTP/[host].siv.de@SIV.DE ptype 1 (KRB5_NT_PRINCIPAL) vno 3 etype 0x11 (AES128-SHA1) keylength 16 (0x816fbefc76bcd97ef3ec7dc2971b09f5)

### 2.2.3 Überprüfen des SPN

*Befehl:*

setspn -L <account>

*Beispiel mit Ausgabe:*

c:\>setspn -L [user]

Registrierte Dienstprinzipalnamen (SPN) für CN=[user],CN=Users,DC=siv,DC=de:

HTTP/[host].siv.de

### 2.2.4 Weitere SPN hinzufügen

Bitte bei allen Konfigurationen immer auf die genaue Schreibweise achten. Kleine Fehler führen hier dazu, dass die Kerberos-Konfiguration nicht funktioniert. Es ist besonders auch auf die Groß- und Kleinschreibung zu achten. Die Beispiele zeigen, wie es richtig gemacht werden muss.

*Befehl:*

setspn –a <SPN> <account>

*Beispiele mit Ausgaben:*

c:\>setspn -a HTTP/[host].siv.de@SIV.DE soatestuser

Dienstprinzipalnamen (SPN) für CN=[user],CN=Users,DC=siv,DC=de werden registriert.

HTTP/[host].siv.de@SIV.DE

Aktualisiertes Objekt

c:\>setspn -L [uer]

Registrierte Dienstprinzipalnamen (SPN) für CN=[user],CN=Users,DC=siv,DC=de:

HTTP/[host].siv.de@SIV.DE

HTTP/[host].siv.de

c:\>setspn -a HTTP/[host]@SIV.DE soatestuser

Dienstprinzipalnamen (SPN) für CN=[user],CN=Users,DC=siv,DC=de werden registriert.

HTTP/[host]@SIV.DE

Aktualisiertes Objekt

c:\>setspn -L [user]

Registrierte Dienstprinzipalnamen (SPN) für CN=[user],CN=Users,DC=siv,DC=de:

HTTP/[host]@SIV.DE

HTTP/[host].siv.de@SIV.DE

HTTP/[host].siv.de

## 2.3 Kerberos-Konfiguration auf dem Linux-Server

Verwendete Dokumente für dieses Kapitel: [1]

### 2.3.1 Update der JDK Security Policy Files

Wenn AES-256 Encryption verwendet werden soll, werden die Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files benötigt.

Diese können von [9] herunter geladen werden.

Die Jurisdiction Policy Files entpacken.

Entsprechend der Anleitung in der README.txt installieren. Darauf achten, dass das richtige JDK/JRE verwendet wird.

### 2.3.2 Kopieren der keytab-Datei auf dem Linux-Server

Kopieren der keytab-Datei nach <Domain-Home>.

Beispiel:

/[domain home path]/test.keytab

### 2.3.3 Editieren der krb5.conf-Datei auf dem Linux-Server

Bitte bei allen Konfigurationen immer auf die genaue Schreibweise achten. Kleine Fehler führen hier dazu, dass die Kerberos-Konfiguration nicht funktioniert. Es ist besonders auch auf die Groß- und Kleinschreibung zu achten. Die Beispiele zeigen, wie es richtig gemacht werden muss.

Die Datei befindet sich unter /etc/krb5.conf und sollte folgenden Inhalt haben:

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 default_realm = SIV.DE
 dns_lookup_realm = false
 dns_lookup_kdc = false
 ticket_lifetime = 600
 forwardable = true

[realms]
 SIV.DE = {
  kdc = ads.siv.de
  admin_server = ads.siv.de
  default_domain = SIV.DE
```

```
}

[domain_realm]
 .siv.de = SIV.DE

[appdefaults]
autologin = true
forward = true
forwardable = true
encrypt = true
```

Wichtig ist hier auf die Großschreibung von SIV.DE an den entsprechenden Stellen zu achten.

### 2.3.4 Testen des SPN und der keytab-Datei (notwendig)

In das Domain-Home wechseln.

Folgendes Kommando ausführen: .bin/setDomainEnv.sh

Unbedingt den Punkt vor dem bin setzen!

Mit diesem Befehl wird das Domain-Environment konfiguriert.

In das Domain-Home wechseln.

*Befehl ausführen:*

kinit -V -k -t <keytab file> HTTP/<host name.domain>@<domain>

*Beispiel:*

kinit -V -k -t /[domain home path]/test.keytab HTTP/[host].siv.de@SIV.DE

Ausgabe bei Erfolg:

Using default cache: /tmp/krb5cc_54321

Using principal: HTTP/[host].siv.de@SIV.DE

Using keytab: /[domain home path]/test.keytab

Authenticated to Kerberos v5

Nur wenn der rot markierte Text erscheint, waren wir erfolgreich und können weiter gehen. Wenn nicht zurück zum Start und das Problem finden.Hiermit hat sich der Linux-Server gegenüber dem KDC (ADS) authentisiert.

### 2.3.5 Erstellen der krb5Login.conf

Die Datei im Domain-Verzeichnis erstellen.

*Beispiel:*

/[domain home path]/krb5Login.conf

*Inhalt:*

```
com.sun.security.jgss.krb5.initiate {

    com.sun.security.auth.module.Krb5LoginModule required
    principal="HTTP/[host].siv.de@SIV.DE" useKeyTab="true"
    keyTab="test.keytab" storeKey="true";
};

com.sun.security.jgss.krb5.accept {

    com.sun.security.auth.module.Krb5LoginModule required
    principal="HTTP/[host].siv.de@SIV.DE" useKeyTab="true"
    keyTab="test.keytab" storeKey="true";

};
```

Nicht den kompletten Pfad zur keytab-Datei angeben!

### 2.3.6 Editieren der setDomainEnv.sh

Aufruf der setDomainEnv.sh mit einem Editor.

Beispiel:

vi /[domain home path]/bin/setDomainEnv.sh

Folgende Zeilen an oberster Stelle eintragen, wo die “EXTRA_JAVA_PROPERTES” gesetzt werden:

EXTRA_JAVA_PROPERTIES="${EXTRA_JAVA_PROPERTIES} -Djava.security.auth.login.config=/[domain home path]/krb5Login.conf"

EXTRA_JAVA_PROPERTIES="${EXTRA_JAVA_PROPERTIES} -Djavax.security.auth.useSubjectCredsOnly=false"

EXTRA_JAVA_PROPERTIES="${EXTRA_JAVA_PROPERTIES} -Dweblogic.security.enableNegotiate=true"

### 2.3.7 Testen der Kerberos-Konfiguration mit Browser SSO (optional)

Dieser Test zeigt auf, ob die Kerberos-Konfiguration außerhalb der Nutzung des OWSM funktioniert. Sie beweist nicht, ob der OWSM richtig konfiguriert ist (bis an dieser Stelle in dieser Dokumentation ist der OWSM auch noch nicht konfiguriert worden).

Es wird empfohlen diesen Test auszuführen, um sicherzustellen, dass die grundsätzliche Kerberos-Konfiguration richtig ist.

Es wird empfohlen folgende Dateien vor den Änderungen zu sichern:

/[domain home path]/config/fmwconfig/jps-config.xml

/[domain home path]/config/config.xml

**Einrichtung des Identity Asserter in der Domain**

In der WLS Administration Console die Seite aufrufen, wo die Authentication Providers aufgelistet werden. Pfad: Home >Summary of Security Realms >myrealm >Providers

Links oben auf „Lock & Edit“ klicken.

„New“ klicken.

Namen eintragen (z.B. SPNEGO_NIA)

„NegotiateIdentityAsserter“ auswählen.

Laut [1] soll der Asserter an erste Stelle der Provider gesetzt werden. In dem Test-Environment war dies nicht notwendig. Wenn doch dann noch folgenden Schritt ausführen:

Mit „Reorder“ Identity Asserter an erste Stelle der Provider setzen.

„OK“ klicken.

„Activate Changes“ links oben klicken.

Grüne Nachricht erscheint bei Erfolg: „All changes have been activated. However 4 items must be restarted for the changes to take effect.”

Es kann sein, dass man sich nach dem Neustart nicht mehr normal in die WLS Console einloggen kann. Dies ist ein Bug von Oracle. Workaround: Sich über die OSB Console in die WLS Console einloggen (oberes Menü). Sollte das auch nicht mehr gehen, dann hilft nur noch die jps-config.xml und die config.xml zurückzuspielen, die hoffentlich vor den Änderungen gesichert wurden.

**Browser SSO testen**

Vorraussetzung: Arbeitsplatzrechner ist in der Windows-Domain und User ist im AD.

Test in der WLS Console:

Pfad: Home >Summary of Servers >Summary of Security Realms >myrealm >Users and Groups

Nutzer muss in der Liste zu finden sein. Wenn nicht muss die Konfiguration überprüft werden.

Prüfen ob alle Server wieder laufen. Dazu in der WLS Administration Console (Pfad: Home >Summary of Security Realms >myrealm >Providers >Summary of Environment >Summary of Servers) nutzen.

Mit dem Google Chrome Browser testen (benötig keine weitere Konfiguration) Siehe auch [1] Abschnitt „I. Configure the web browser“ für Internet Explorer und Firefox.

*Beispiel:*

http://[host].siv.de:8001/bpm/composer/ssologin oder

http://[host].siv.de:8001/integration/worklistapp/ssologin

Nach dem erfolgreichen Test, den Identity Asserter wieder entfernen und Server neu starten.

### 2.3.8 Wenn etwas schief geht

Wenn etwas nicht funktioniert ist es wichtig, den Debug-Modus zu aktivieren, um so detaillierte Informationen wie nur möglich zu bekommen.

Für das .out Logs:

In die setDomainEnv.sh zu der EXTRA_JAVA_PROPERTIES Sektion folgenden Eintrag hinzufügen (siehe auch Abschnitt weiter oben):

EXTRA_JAVA_PROPERTIES="${EXTRA_JAVA_PROPERTIES} -Dsun.security.krb5.debug=true"

Für die .log Logs

1. WLS Administration Console öffnen
2. Environment klicken und dann Servers
3. OSB Server auswählen
4. den Debug-Reiter klicken
5. weblogic aufklappen und dann security
6. atn anhaken
7. Unten oder oben den Enable-Button klicken

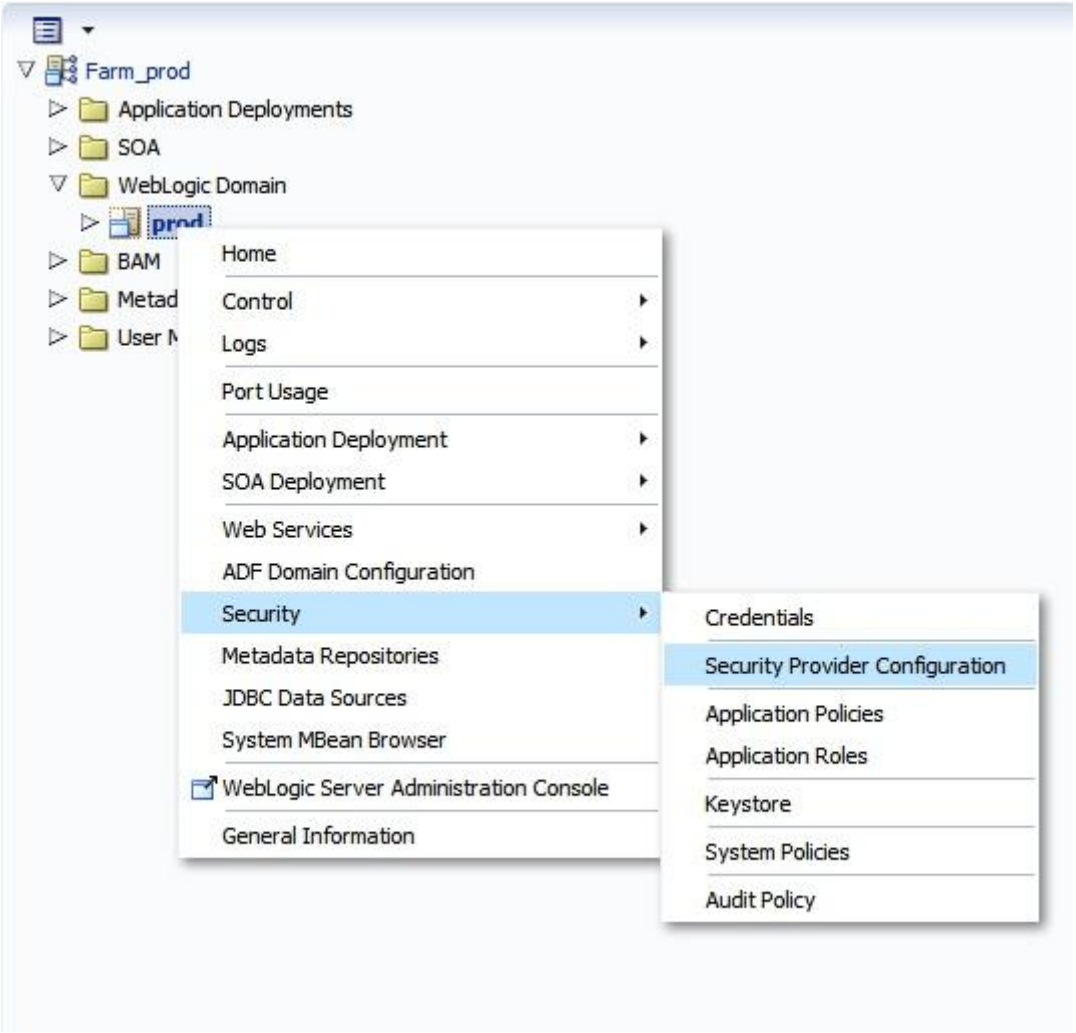
Siehe auch References in [1] und [11]

Der Debug-Modus sollte wieder zurückgesetzt werden, wenn die Kerberos-Konfiguration funktioniert.

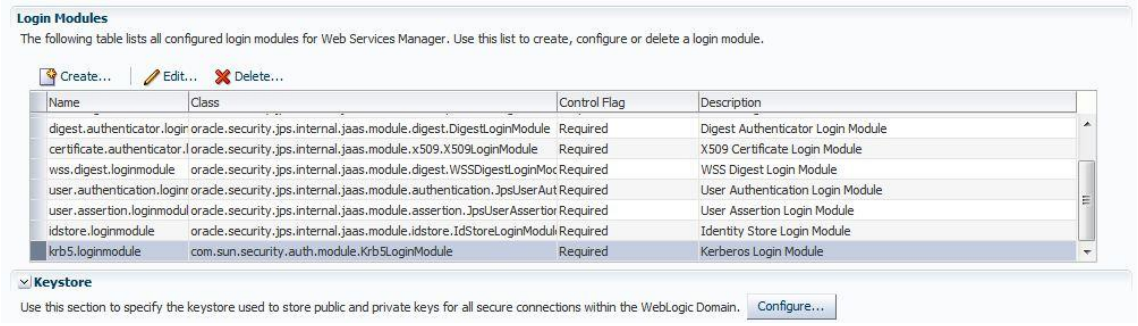
### 2.3.9 OWSM konfigurieren

**Kerberos Login Modul konfigurieren**

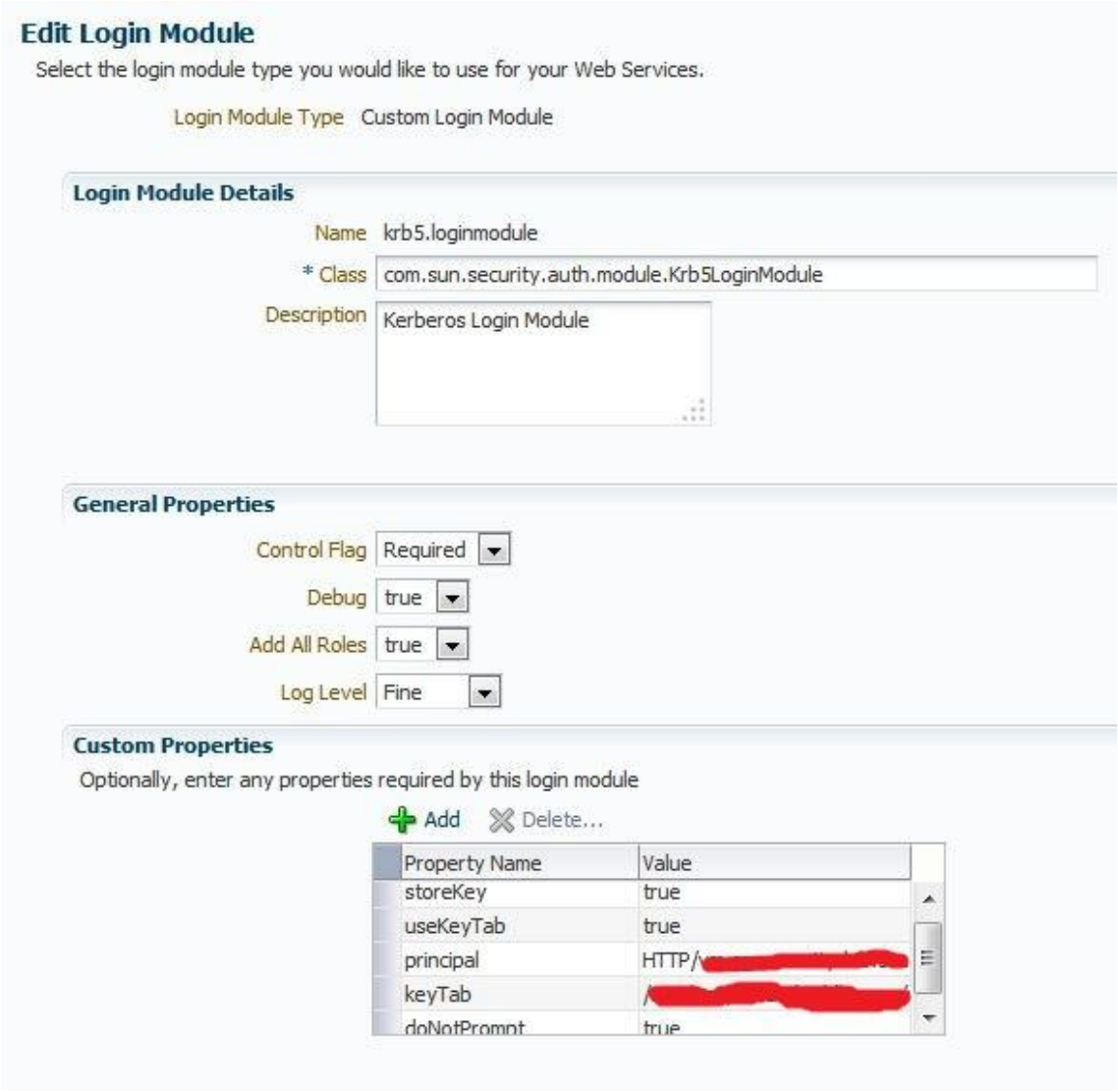
Im Enterprise Manager von der WebLogic Doamin die Security Provider Configuration aufrufen.




Das krb5.loginmodule auswählen und den Edit-Button klicken.




Hier wird Principal Name http/[host].siv.de@SIV.DE und seine keyTab Datei eingetragen. Hier ist besonders darauf zu achten, dass man die key-Namen richtig schreibt und auf Groß-/Kleinschreibung achtet (Camel Case). Das kann viel Arbeit ersparen.




Siehe auch [2]

**Key store einrichten**

Den Befehl keytool aus der Java-Umgebung starten, mit der der OSB gestartet wird.

*Erstellen eines Zertifikats für die SOA:*

[oracle@server bin]$ keytool -genkey -alias serverKey -keyalg "RSA" -sigalg "SHA1withRSA" -dname "CN=server, C=DE" -keypass welcome -keystore /tmp/server.jks -storepass welcome -validity 3600

*Kopieren der server.jks zum config/fmwconfig-Verzeichnis der SOA Suite:*

[oracle@server bin]$ cp /tmp/server.jks /[domain home path]/config/fmwconfig/

*Testen der server.jks:*

[oracle@oraclevm90 bin]$ keytool -list -keystore /tmp/server.jks

Enter keystore password:

Keystore type: JKS

Keystore provider: SUN

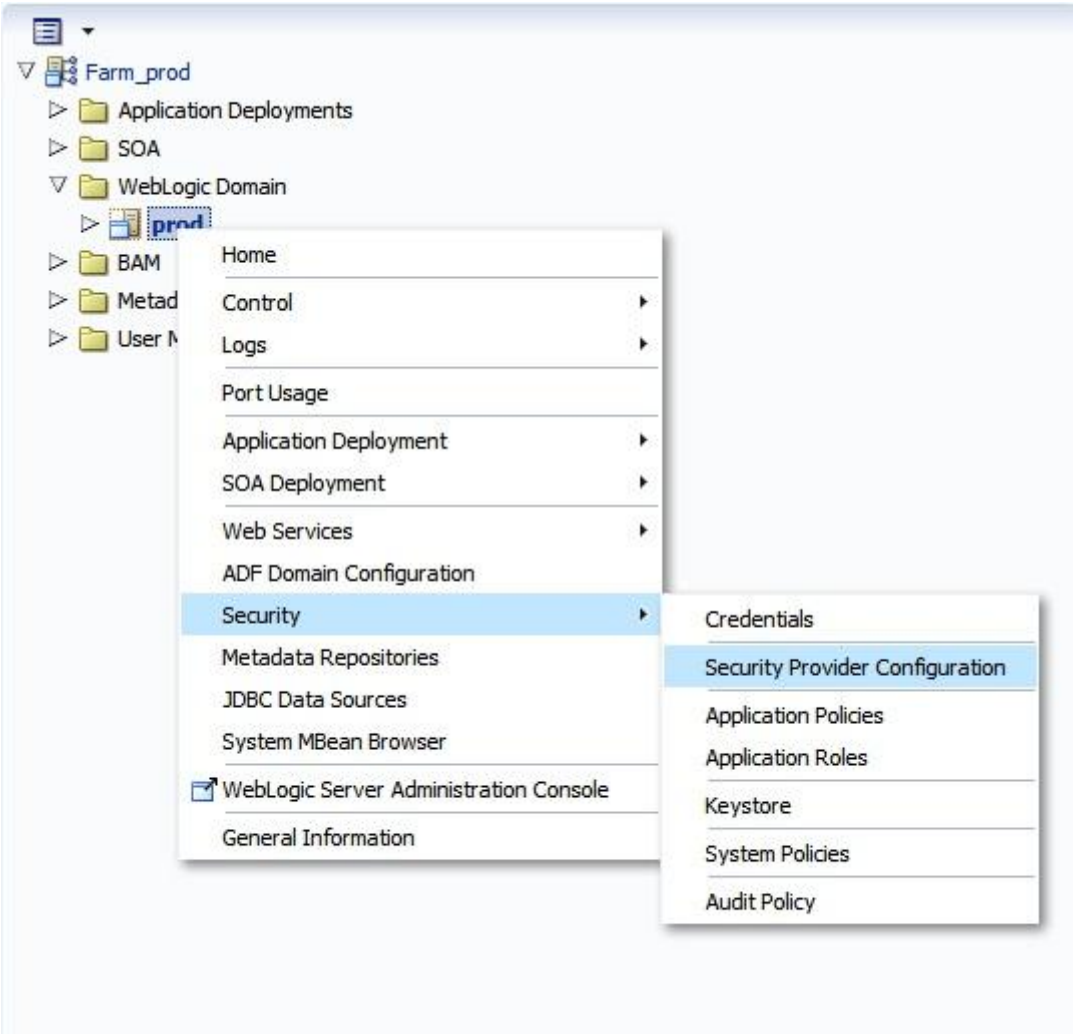
Your keystore contains 1 entry

serverkey, Apr 15, 2014, PrivateKeyEntry,

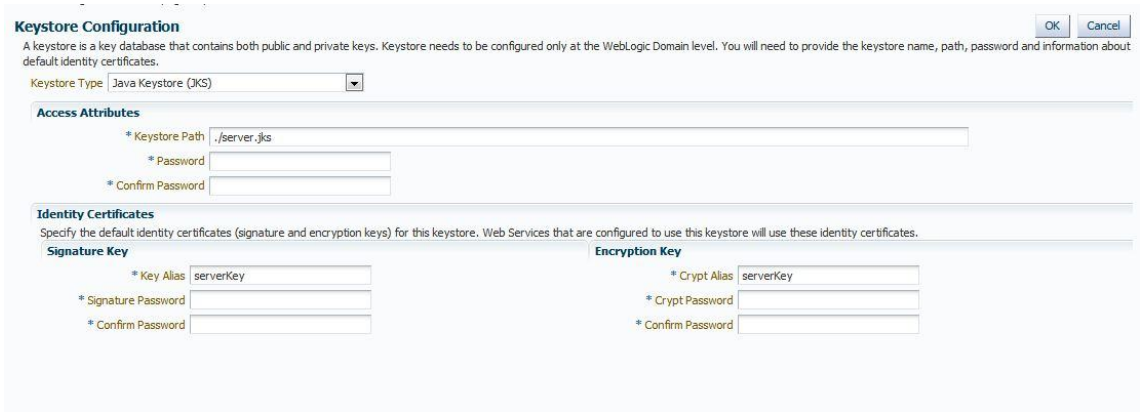
Certificate fingerprint (SHA1): 16:A9:59:62:CB:AC:86:37:57:DD:A1:CC:B0:AE:2F:DF:72:97:E8:C4

**Import des Java Keystore in den Enterprise Manager**

Im Enterprise Manager die SOA-Domäne auswählen und mit rechter Maustaste den Menüpunkt Security Provider Configuration öffnen.




Auf den Configure-Button in der keystore-Sektion klicken.

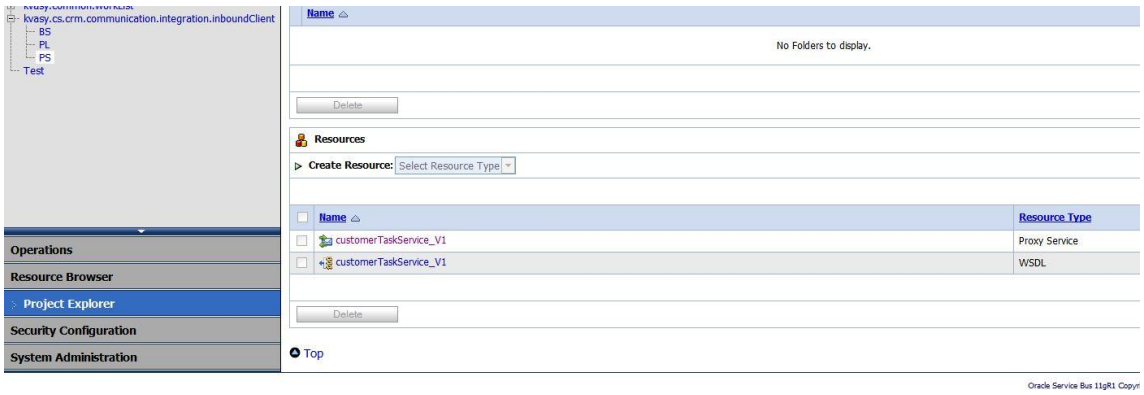

Eintragen von ./server.jks in das Keystore Path-Feld. Eintragen von serverKey in das Key Alias- und Crypt Alias-Feld. Eintragen von welcome in allen Passwort-Feldern.

Restart aller Server der Domain.

Siehe auch [3]

## 2.3.10 OSB Web Service mit Policy sichern

Das Projekt wählen und in das Verzeichni wechseln in dem der Proxy Server liegt.




customerTaskService_V1 anklicken und auf den Policies-Reiter klicken.


Die oracle/wss11_kerberos_token_with_message_protection_basic128_service_policy hinzufügen und Konfiguration speichern.

Siehe auch [1].

## 2.4 Abschließender Test des WCF Client

Den WCF Client ausführen. Er sollte nun den entsprechenden Service aufrufen können und die Ausgaben ausgeben. In unserem Fall tat er das und gab eine Liste von Titeln aus.

# 3 Glossar/Abkürzungsverzeichnis

| Begriff/Abkürzung | Erklärung |
|---|---|
| ADS | Active Directory Server |
| AD | Active Directory |
| KDC | Key Distribution Center |
| OSB | Oracle Service Bus |
| OWSM | Oracle Web Service Manager |

# 4 Quellen


| | |
|---|---|
| [1] | How To Configure Kerberos SSO Authentication for Linux or Unix Based Webcenter Content (Doc ID 1543209.1) |
| [2] | http://biemond.blogspot.de/2011/09/using-owsm-kerberos-policies.html |
| [3] | http://biemond.blogspot.de/2011/08/do-saml-with-owsm.html |
| [4] | http://docs.oracle.com/javase/7/docs/jre/api/security/jaas/spec/com/sun/security/auth/module/Krb5LoginModule.html |
| [5] | Oracle Service Bus 11g Development Cookbook |
| [6] | http://docs.oracle.com/cd/E28280_01/web.1111/e16098/interop_net.htm#BIIHEBGC |
| [7] | http://technet.microsoft.com/en-us/library/bb742431.aspx |

| | |
|---|---|
| [9] | http://www.oracle.com/technetwork/java/javase/downloads/index.html |
| [10] | https://blogs.oracle.com/blogbypuneeth/entry/steps_to_configure_weblogic_server |
| [11] | http://docs.oracle.com/javase/1.5.0/docs/guide/security/jgss/tutorials/Troubleshooting.html |
| [12] | http://fusionsecurity.blogspot.de/2011/07/5-minutes-or-less-kerberos.html |
| [13] | http://www.roguelynn.com/words/explain-like-im-5-kerberos/ |
| [15] | http://www.youtube.com/watch?v=KD2Q-2ToloE |