Platform SDK Developer's Guide

Using Kerberos Authentication in Platform SDK

4/4/2025

Using Kerberos Authentication in Platform SDK

Java

Introduction

Platform SDK supports using Kerberos authentication with Configuration Server. Platform SDK can independently obtain a Kerberos ticket or use Kerberos ticket provided by user. Each case requires an individual approach.

Using Service Principal Name

Service Principal Name (SPN) is a unique identifier of service which in couples with user's credentials can uniquely identify access to requested service. To use the `ServicePrincipalName` user have to assign it using `setSPN` method of a channel `Endpoint`.

Microsoft-specific Note: SPN has to be registered in Active Directory using utility `setspn.exe`. See Microsoft technet documentation. User has to have the required access rights to execute this utility's commands.

Code example: Connect CS using SPN

```
ConfServerProtocol protocol = new ConfServerProtocol(new Endpoint(host, port).setSPN(spn));
protocol.setClientName(clientName);
protocol.setClientApplicationType(clientType);

protocol.open();
```

Usage of Independently Acquired Ticket

If user has a ticket as byte array data, Platform SDK can use it too. In this case user has to assign ticket acquirer to the protocol instance.

Code example: Connect to CS using raw data GSS Kerberos ticket

```
ConfServerProtocol protocol = new ConfServerProtocol(new Endpoint(host, port));
protocol.setClientName(clientName);
protocol.setClientApplicationType(clientType);
RawDataTicketAcquirer ticketAcquirer = new RawDataTicketAcquirer(ticketBytes);
```

```
protocol.setTicketAcquirer(ticketAcquirer);

protocol.Open();
```

The previous example applies only for tickets compatible with GSS API (RFC 2743). Configuration Server also supports pure Kerberos tickets without a GSS envelope, as obtained by using the MIT Kerberos library for instance.

In this case please use the second constructor of `RawDataTicketAcquirer`:

```
RawDataTicketAcquirer(byte[] arguments, bool isGSSTicket)
```

If `isGSSTicket` is false, then a registration message is created with another attribute specially designed for this goal.

Code example: Connect to Configuration Server Using Raw Data Pure Kerberos Ticket

```
boolean isGSSTicket = false;
ConfServerProtocol protocol = new ConfServerProtocol(new Endpoint(host, port));
protocol.setClientName(clientName);
protocol.setClientApplicationType(clientType);
RawDataTicketAcquirer ticketAcquirer = new RawDataTicketAcquirer(ticketBytes, isGSSTicket);
protocol.setTicketAcquirer(ticketAcquirer);

protocol.Open();
```

Notes for Windows

Kerberos authorization as current logined user must be enabled manually in few steps:

1. set registry key "AllowTGTSessionKey"=dword:00000001 in HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters
2. update JCE policy from oracle site
   - jre1.7/lib/security <- http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html
   - jre1.8/lib/security <- http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html
3. Time on kdc,server and client machines must be seconds synchronized
4. krb5.conf can be placed in any folder but you must specify its location using system property "java.security.krb5.conf"

.NET

Introduction

Platform SDK supports using Kerberos authentication with Configuration Server. Platform SDK can independently obtain a Kerberos ticket or use Kerberos ticket provided by user. Each case requires an

individual approach.

Using Service Principal Name

Service Principal Name (SPN) is a unique identifier of service which in couples with user's credentials can uniquely identify access to requested service. To use the SPN user have to assign field `ServicePrincipalName` of `AbstractChannel.Endpoint`.

Microsoft-specific Note: SPN has to be registered in Active Directory using utility `setspn.exe`. See Microsoft technet documentation. User has to have the required access rights to execute this utility's commands.

Code example: Connect CS using SPN

```
var protocol = new ConfServerProtocol(new Endpoint(host, port) { ServicePrincipalName = spn })
{
  ClientApplicationType = clientApp,
  ClientName = clientName
};

protocol.Open();
```

Usage of Independently Acquired Ticket

If user has a ticket as byte array data Platform SDK can use it too. In this case user has to assign ticket acquirer to the protocol instance.

Code example: Connect to CS using raw data GSS Kerberos ticket

```
var protocol = new ConfServerProtocol(new Endpoint(host, port))
{
  ClientApplicationType = clientApp,
  ClientName = clientName,
  KerberosTicketAcquirer = new RawDataTicketAcquirer(rawTicketData)
};

protocol.Open();
```

The previous example applies only for tickets compatible with GSS API (RFC 2743). Configuration Server also supports pure Kerberos tickets without a GSS envelope, as obtained by using the MIT Kerberos library for instance.

In this case please use the second constructor of `RawDataTicketAcquirer`:

```
RawDataTicketAcquirer(byte[] arguments, bool isGSSTicket)
```

If `isGSSTicket` is false, then a registration message is created with another attribute specially designed for this goal.

Code example: Connect to Configuration Server Using Raw Data Pure Kerberos Ticket

```
var isGSSTicket = false;
var protocol = new ConfServerProtocol(new Endpoint(host, port))
{
  ClientApplicationType = clientApp,
  ClientName = clientName,
  KerberosTicketAcquirer = new RawDataTicketAcquirer(rawTicketData, isGSSTicket)
};

protocol.Open();
```