Platform SDK Developer's Guide

Using Kerberos Authentication in Platform SDK

12/14/2025

Contents

Using Kerberos Authentication in Platform SDK

Introduction

Beginning with release 8.1.401.04, Platform SDK for .NET supports Kerberos-based single sign-on authentication with Configuration Server. There are two scenarios available for implementation, described in more detail below:

- Using Service Principal Name (SPN) - Platform SDK can independently obtain a Kerberos ticket.
- Using Independently Acquired Ticket - Platform SDK can use a Kerberos ticket that is provided by the user.

Using Service Principal Name (SPN)

SPN is an identifier which, when combined with user credentials, can uniquely identify access to a requested service. To use SPN, your application must set the `ServicePrincipalName` field that is part of `AbstractChannel.Endpoint`.

Code Example: Connect to Configuration Server Using SPN

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

protocol.Open();
```

Microsoft Specific Note: SPN has to be registered in Active Directory using the `setspn.exe` utility. See the Microsoft teachnet documentation for details. To execute commands with this utility, a user must have the required access rights.

> **Tip**
> Platform SDK for .Net can only acquire tickets that are compatible with GSS API (RFC 2743).

Using Independently Acquired Tickets

Platform SDK can also use independently acquired tickets that are in byte array data form. In this case, the application has to assign a ticket acquirer to the protocol instance, as shown in the following code example.

Code Example: Connect to Configuration Server 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 is only applicable for tickets that are compatible with GSS API (RFC 2743).

Configuration Server also supports pure Kerberos tickets without a GSS envelope, such as those obtained using the MIT Kerberos library. In this case, your application should use the second constructor for RawDataTicketAcquirer:

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

If `isGSSTicket` is set to false, then a registration message is created with another attribute designed for pure Kerberos tickets.

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

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

protocol.Open();
```