Spring Security Kerberos Plugin - Reference Documentation

Burt Beckwith

Version 3.0.0

Table of Contents

Chapter 1. Introduction to the Spring Security Kerberos Plugin

The Kerberos plugin adds Kerberos single sign-on support to a Grails application that uses Spring Security. It depends on the Spring Security Core plugin.

Once you have configured a Kerberos server (typically Microsoft Active Directory or MIT Kerberos) and have configured your Grails application(s) as clients, users who are have authenticated at the Kerberos server will be automatically authenticated as a user of your application(s) without requiring a password.

In addition to this document, you should read the Spring Security Kerberos documentation.

1.1. History

- December 8, 2015
  - 3.0.0 release
- December 7, 2015
  - 1.0.0 release
- October 24, 2013
  - 1.0-RC1 release
- January 30, 2011
  - initial 0.1 release

Chapter 2. Usage

> Configuring your Kerberos server is beyond the scope of this document. There are several options and this will most likely be done by IT staff. It’s assumed here that you already have a running Kerberos server.

The plugin adds support for Kerberos and is based on the Spring Security Kerberos extension.

There isn’t much that you need to do in your application to be a Kerberos client. Just install this plugin, and configure the two required parameters and whatever optional parameters you want in `application.groovy`. These are described in detail in the Configuration section but typically you only need to set these properties:

```
grails.plugin.springsecurity.kerberos.ticketValidator.servicePrincipal =
      'HTTP/kerberos.server.name@KERBEROS.DOMAIN'

grails.plugin.springsecurity.kerberos.ticketValidator.keyTabLocation =
      'file:///path/to/your.keytab'
```

2.1. UserDetailsService

Currently the only information that is retrieved from Kerberos is the username (plus the authentication status of course) so you’ll need to have user and role data in your database corresponding to Kerberos users. Since you’ll be authenticating externally you can either remove the password field from the user class and use a custom `UserDetailsService` or just store dummy values in the password column to satisfy the not-null constraint.

Chapter 3. Configuration

There are a few configuration options for the Kerberos plugin.

> All of these property overrides must be specified in `grails-app/conf/application.groovy` (or `application.yml`) using the `grails.plugin.springsecurity` suffix, for example
>
> ```
> grails.plugin.springsecurity.kerberos.debug = true
> ```

There are two required properties:

Name	Default	Meaning
kerberos.ticketValidator.servicePrincipal	none, required	the web application service principal, e.g. `HTTP/www.example.com@EXAMPLE.COM`
kerberos.ticketValidator.keyTabLocation	none, required	the URL to the location of the keytab file containing the service principal’s credentials, e.g. `file:///etc/http-web.keytab`

and some optional properties:

Name	Default	Meaning
kerberos.active	`true`	set to `false` to disable the plugin
kerberos.client.debug	`false`	if `true` enables debug logs for the kerberos client bean
kerberos.configLocation	`null`	The location of the Kerberos config file (specify the path to the file, but omit "file://", e.g. "c:/krb5.conf"). Leave unset to use the default location (e.g. `/etc/krb5.conf`, `c:\winnt\krb5.ini`, `/etc/krb5/krb5.conf`)
kerberos.debug	`false`	if `true` enables debug logs for the kerberosConfig bean

Name	Default	Meaning
kerberos.skipIfAlreadyAuthenticated	`true`	if `true` skip SpnegoAuthenticationProcessingFilter processing if already authenticated
kerberos.spnegoEntryPointForwardUrl	`null`	if set (e.g. '/login/auth') the EntryPoint will forward there in addition to setting the `WWW-Authenticate` header
kerberos.successHandler.headerName	'WWW-Authenticate'	the name of the header to set following successful authentication
kerberos.successHandler.headerPrefix	'Negotiate '	the prefix for the encoded response token value
kerberos.ticketValidator.debug	`false`	if `true` enables debug logs for the ticketValidator bean
kerberos.ticketValidator.holdOnToGSSContext	`false`	if `true` hold on to the GSS security context, otherwise call `dispose()` immediately