Technical White Paper - JBoss Security

Clustered SSO

1.0

Table of Contents

Target Audience

This technical white paper on ClusteredSingleSignOn is intended for System Administrators, Developers and Enterprise Architects who intend to implement SSO solution using JBoss Clustered Servers environment.

Preface

The Technical White Papers from JBoss Security are important sources of information for secure operation of JBoss Products as well as applications running on them.

1

Clustered SingleSignOn

1.1. Introduction to SingleSignOn

Single Sign-On allows an user to authenticate to one application and be recognized on all applications deployed in the same virtual host.

1.2. JBoss implementation of Clustered SingleSignOn solution

JBoss web tier offers clustered SSO solution for web clients with the `org.jboss.web.tomcat.tc5.sso.ClusteredSingleSignOn` valve, which extends the standard Tomcat SSO valve. JBossCache is used for SSO credential caching and replication.

- The SSO solution will enable SSO failover across multiple nodes.
- It allows load balancer to direct request for different web apps to different clustered members while maintaining the SSO.
- The user will not be challenged as long as he accesses only unprotected resources in any of the web applications on this virtual host.
- To access a protected resource in any web app, the user will be challenged to authenticate using the login method defined for the web app.
- Once authenticated, the roles associated with this user will be utilized for access control decisions across all of the associated web applications, without challenging the user to authenticate them to each application individually.
- If the user logs out of one web application (for example, by invalidating the corresponding session if form based login is used), the user's sessions in all web applications will be invalidated.
- A session timeout does not invalidate the SSO if other sessions are still valid.

1.2.1. Configuration of Clustered SingleSignOn for different JBoss versions

1. For 3.2.4

   To enable Clustered SingleSignOn in JBoss, in the `jbossweb-tomcat5x.sar/server.xml` file, inside the element of any virtual hosts for which you want single sign-on support, add an element:

```
<Valve className="org.jboss.web.tomcat.tc5.sso.ClusteredSingleSignOn"
         treeCacheName="jboss.cache:service=TomcatClusteringCache" debug="0"/>
```

   If using ClusteredSingleSignOn valve, make sure the standard single sign on valve is commented out.

   The Clustered SingleSignOn valve depends on the existence of a JBossCache MBean, which must be separately configured. This is done by deploying a MBean as a service. Note that the value of the treeCacheName attribute of Clustered SingleSignOn valve element in `server.xml` must match the JMX object name of the JBossCache MBean.

2. For 3.2.6/4.0

   Beginning with JBoss-3.2.6 and 4.0 releases, the Clustered SingleSignOn valve by default shares a JBossCache MBean with the clustered HTTP session replication service. So we don't need to configure treeCacheName and a MBean service explicitly because it is already available as a standard JBoss Service. We just need to configure Clustered SingleSignOn valve like this in `server.xml` file under `jbossweb-tomcat5x.sar` directory.

```
<Valve className="org.jboss.web.tomcat.tc5.sso.ClusteredSingleSignOn"
         debug="0"/>
```

   The default JBossCache MBean is described in the `tc5-cluster-service.xml` file in the deploy directory.

3. For 4.0.4

   Prior to release 4.0.4.CR2, the SSO cookie is scoped to a single hostname. It means SSO will work for same host names like this:

   - `http://www.xyz.com/app1`
   - `http://www.xyz.com/app2`

   and SSO will not work if the host names are different:

   - `http://app1.xyz.com`
   - `http://app2.xyz.com`

   To address this issue, a new attribute called cookie domain in the CSSO valve. The purpose of this attribute is to increase the scope of the SSO cookie from default '/' domain to much broader domain. For example, for the above case to support SSO, we can configure like this:

```
<Valve className="org.jboss.web.tomcat.tc5.sso.ClusteredSingleSignOn"
         cookieDomain="xyz.com"/>
```

1.3. Limitations

- Only useful within a cluster of JBoss web servers; SSO does not propagate to other resources.
- Requires use of container managed authentication (via <login-config> element in web.xml).
- Requires Cookies. SSO is maintained via a cookie and URL rewriting is not supported.
- BuddyReplication and Clustered SingleSignOn

  From Jboss 4.0.5 onwards JBossCache offers configuration option called Buddy Replication, which means the cached data will not be replicated across the entire cluster. Instead it will be replicated to one or more buddy to ensure backup copies. Buddy Replication is much more efficient, and it is perfect for cached HTTP sessions as it uses session affinity which makes the load balancer send all requests for a session to the same server. However, with Clustered SingleSignOn, the load balancer might link the user to different servers for the different webapps being accessed. As a result, multiple servers in the cluster need to monitor the cached data. For example, if the user logs out of the SSO on one server, all the cluster members in the cluster need to know that the SSO is invalidated. It is insufficient if only the buddies are informed. If users want to enable buddy replication for HTTP sessions, to over come the above limitation they would have to configure and deploy Separate JBossCache MBean for Clustered SingleSignOn.
- Custom Principals and Classloader-issues

  If you use custom principal implementations in your login modules in isolated EAR-files, you need to make sure that their classloaders use exactly the * same * library. You can guarantee this by putting the principal object (and other associated objects) into server/default/lib directory and removing them from your WAR/EAR files.

1.4. Resources

JBoss Wiki [1]

JBoss Documentation [2]

[1] http://wiki.jboss.org/wiki/Wiki.jsp?page=SingleSignOn
[2] http://docs.jboss.org/jbossas/jboss4guide/r3/html/ch9.chapt.html#d0e22429