OpenAM/OpenSSO & JBoss EAP 5.1

(Lunch n' Learn)

JBoss by Red Hat

Peter Škopek

Feb 10, 2011

Abstract

This presentation will introduce you to the OpenAM project which is an incarnation of the OpenSSO project.
Examples will show usage with JBoss EAP 5.1.

Section 1

Welcome

redhat.

Agenda

1. **Welcome**
2. **A Piece of History**
3. **OpenAM Introduction**
   - How to deploy OpenAM on EAP 5.1
4. **How to Secure your Application**
   - Overview
   - Policy Agent
   - Setup
   - Conclusion
5. **Securing Seam Application**

Section 2

A Piece of History

redhat.

Neglected by Oracle After The Takeover

- Sun created OpenSSO in 2005 as an open source version of the Sun Java System Access Manager, licensing the software under the Common Development and Distribution License (CDDL). The software was designed for large transactional Web sites that require users to log in and keep accounts. [Neglected]
- Sun Microsystems has been taken over by Oracle in January 2010.
- No public downloads available of original OpenSSO project.
- Wiki not accessible anymore.
- Looks like Oracle neglected OpenSSO project.

redhat.

The King is dead. Long live the King.

- So an OpenAM was born. [OpenAM]
- Took over by Norwegian company, ForgeRock, which already released OpenAM Release9. [OpenAM-Download]

Section 3

OpenAM Introduction

redhat.

What is OpenAM?

- OpenAM provides core identity services to simplify the implementation of transparent single sign-on (SSO) as a security component in a network infrastructure.
- OpenAM provides the foundation for integrating diverse web applications that might typically operate against a disparate set of identity repositories and are hosted on a variety of platforms such as web and application servers.

redhat.

Installation

- Download EAP 5.1 on
  `http://download.devel.redhat.com/released/JBEAP-5/5.1.0/zip/jboss-eap-5.1.0.zip`
  Yes, this time I used regular EAP as shipped to customer.
- Install EAP (unzip it).
- We are going to install OpenAM into clone of the "default" profile called "openam".
- Get OpenAM and unzip it. [OpenAM-Download]
- Get opensso/deployable-war/opensso.war archive and install it to $JBOSS_HOME/server/openam/deploy as exploded. This will help to overcome one issue with JBoss EAP VFS.

redhat.

Setup

- copy $JBOSS_HOME/bin/run.conf to $JBOSS_HOME/server/openam
- Include following properties to $JBOSS_HOME/server/openam/run.conf.

```
JBOSS_HOME=$JAVA_OPTS -Dcom.iplanet.am.cookie.encode=true
JBOSS_HOME=$JAVA_OPTS -Djboss.service.binding.set=ports-01
```

- Create jboss-web.xml file in openam.war/WEB-INF directory of OpenAM deployment with following content:

```
<jboss-web> <class-loading java2ClassLoadingCompliance='true'>
<loader-repository> jbia.loader:loader=opensso
<loader-repository-config> java2ParentDelegaton=true </loader-repository-config>
</loader-repository>
</class-loading>
</jboss-web>
```

- Change property "configuration.dir" in file $JBOSS_HOME/server/default/deploy/opensso.war/WEB-INF/classes/bootstrap.properties

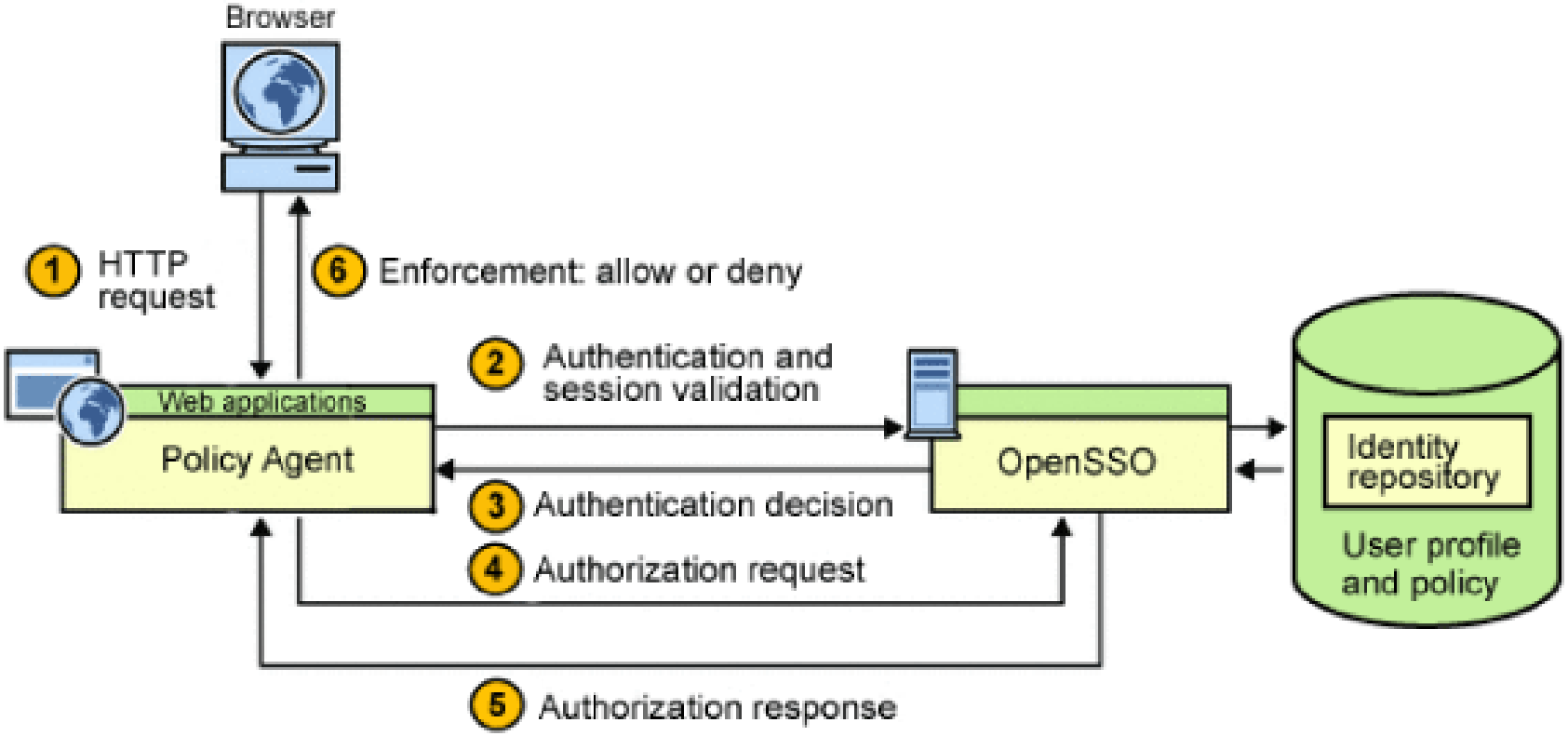
```
configuration.dir=/home/pskopek/projects/openam/jboss-eap-5.1/jboss-as/server/openam/conf/opensso
```

Section 4

How to Secure your Application

redhat.

Overview[PolicyAgents]



Policy Agent can intercept requests to the application and conduct security checks. When a user attempts to access a protected resource or page in the application, the OpenSSO login page is displayed.
After authentication, the user is granted access.

redhat.

The Process part 1 [PolicyAgents]

1. The browser sends a request for the protected resource to the deployment container (such as a Web server or an application server) protected by the Policy Agent.
2. The Policy Agent intercepts the request and checks whether a session token is embedded in a cookie. If the answer is yes, the Policy Agent validates the token for SSO. Otherwise, the Policy Agent directs the user to the OpenSSO login page to log in with the credentials, such as a user name and password, which the Policy Agents then verifies against the data in the identity repository.

redhat.

The Process part 2 [PolicyAgents]

3. If authentication succeeds, the Policy Agent establishes a session token and proceeds to step 4. In addition, the Policy Agent might create a cookie for the user's browser.
   If authentication fails, the Policy Agent denies the user access.
4. The Policy Agent sends an authorization request to the OpenSSO policy service for user access to the protected resource.
5. OpenSSO responds with the policy decision.
6. The Policy Agent interprets the decision and allows or denies access to the requested resource.

redhat.

Installation

- For installing agent and running application we will use different instance of JBoss EAP (using “default“ profile without port shifting)
- Download policy agent for JBoss v 4.2 & v 5.x [OpenAM-Download]
- Unzip and change to j2ee_agents/jboss_v42_agent directory
- Set JAVA_HOME environment variable and

  `run ./bin/agentadmin --install`
- Follow installation instructions (very clean)

redhat.

Installation Notes

1. Use following values for properties when asked for:

   ```
   OpenSSO server URL: http://localhost:8180/opensso
   Agent URL: http://localhost:8080/agentapp
   ```

2. Be careful when asked for password file. File contains password for Agent Profile of your choice which has to contain password characters only.
3. Copy content of locale directory of Agent installation to Agent_001/config directory
4. set JBOSS_CLASSPATH environment variable before starting JBoss EAP which serves the application

redhat.

Policy Agent OpenSSO setup

- Go to `http://localhost:8180/opensso` and click on "Access Control" tab and then on top level realm (in list of realms)
- Select "Agents" tab and "J2EE" sub tab
- Create J2EE Agent named "myfirstjbossagent"
  Use the same password as in Agent password file earlier.
- Use the exactly same agent application URL as in agent installation (`http://localhost:8080/agentapp`)
  Note: Agent wizard doesn't allow you to use host name localhost, but can me changed later.

redhat.

Access Policy Setup

- Go to OpenSSO main page and select the tab "Policies"
- Use new policy to create policies for agentsample application as described in j2ee_agents/jboss_v42_agent/sampleapp/readme.txt file.
  Note: The file also contains instructions how to create subjects (users) and groups for the agentsample application.

redhat.

Conclusion

- Default Agent realm is "AMRealm"
  See: META-INF/jboss.xml in jar file,
  WEB-INF/jboss-web.xml in war file of agentsample.ear.
- ejb-jar.xml and web.xml deployment descriptors are using this role-name element

  `<role-name>id=employee,ou=group,dc=opensso,dc=java,dc=n`
- Agent calls are implemented through filter in web.xml descriptor. Check

  `<filter>` and `<filter-mapping>`

  elements.

Section 5

Securing Seam Application

redhat.

Introduction

PicketLink has a Seam module that enables developers to connect their Seam applications to external identity providers. SAMLv2 as well as OpenID based providers are supported.
There is a sample application called seam-sp, which can be used to play around with a very simple Seam application that enables users to login at an OpenID or SAML identity provider.
[SeamOpenSSO]

redhat.

Setup

- We are going to use seam-sp application from PicketLink. `https://svn.jboss.org/repos/picketlink/federation/branches/Branch_1_x/picketlink-webapps/seam-sp`
- Now proceed OpenSSO admin console and choose "Create hosted entity provider". Choose "test" as the signing key, and enter the name "mycircle" for the new circle of trust and accept all other settings without a change. Press the "configure" button and your identity provider has been configured.
- Restart the OpenAM server.
- Build the application using maven and install it as exploded to $JBOSS_HOME/server/default/deploy

redhat.

Setup part 2

- Replace content of EntityDescriptor with entityID="http://localhost:8888/opensso" in seam-sp.war/WEB-INF/classes/saml-entities.xml file with `http://localhost:8180/opensso/saml2/jsp/exportmetadata.jsp`
- Locate

  ```
  <SamlIdentityProvider
  entityId="http://localhost:8888/opensso" />
  ```

  in seam-sp.war/WEB-INF/classes/external-authentication-config.xml file and change port part of URL to 8180.

redhat.

Configure seam-sp as a service provider in OpenAM

- In OpenSSO admin console choose "Register Remote Service Provider". It will prompt you for a URL where the meta data of the service provider is located. Fill in the following URL: `http://localhost:8080/seam-sp/MetaDataService.seam`
- Press button "Configure"
  Note: Server wich serves the seam-sp application has to be up and running.

redhat

Bibliography

Sean Brydon and Aravindan Ranganathan

Protecting Java EE Applications With OpenSSO Policy Agents
http://www.oracle.com/technetwork/java/policyagents-139399.html

Joab Jackson, IDG News:

OpenSSO, Neglected by Oracle, Gets Second Life
http://www.pcworld.com/businesscenter/article/201889/opensso_neglected_by_oracle_gets_second_life.html

OpenAM download site.

http://forgerock.com/downloads.html

OpenAM site.

http://forgerock.com/openam.html

OpenAM wiki.

https://wikis.forgerock.org/openam/

Marcel Kolsteren

External authentication example using OpenSSO
http://community.jboss.org/wiki/ExternalauthenticationexampleusingOpenSSO

The end.

Thanks for listening.