Configuring an LDAP-based security domain

Goals

After completing this section, students should be able to do the following:

- **Configure a security domain based on the LDAP login module.**

Defining a security domain based on LDAP

A security domain backed by an LDAP server, which stores users and role assignments for an application, can be defined using the Ldap login module. It integrates with an LDAP server and authenticates users against data stored in the LDAP Directory Information Tree (DIT). Users and roles are stored in Organizational Units (OUs) in the LDAP tree with the username used as the Distinguished Name (DN) to uniquely identify a user.

An LDAP-based security domain can be created with the EAP CLI, using the following command:

```
[standalone@localhost:9990] /subsystem=security/security-domain= \
    ldap-domain:add(cache-type=default)
```

```
[standalone@localhost:9990] /subsystem=security/security-domain=ldap-domain \
   /authentication=classic:add \
   (login-modules=[{"code"=>"Ldap","flag"=>"required", \ "module-
   options"=>[("java.naming.factory.initial"=> \
   "com.sun.jndi.ldap.LdapCtxFactory"), \
   ("java.naming.provider.url"=>"ldap://instructor:389"), \
   ("java.naming.security.authentication"=>"simple"), \
   ("principalDNPrefix"=>"uid="),("principalDNSuffix"=>", \ ou=people,
   dc=redhat, dc=com"),("rolesCtxDN"=>"ou=Roles,dc=redhat,dc=com"), \ ("uidAttributeID"=>"member"),
   ("matchOnUserDN"=>"true"), \ ("roleAttributeID"=>"cn"), \
   ("roleAttributeIsDN"=>"false")]}])
```

The corresponding LDAP-based security domain XML definition is as follows:

```
<security-domain name="ldap-domain">
    <authentication>
        <login-module code="Ldap" flag="required">
            <module-option name="java.naming.factory.initial"
  value="com.sun.jndi.ldap.LdapCtxFactory"/>
            <module-option name="java.naming.provider.url" value="ldap://instructor:389"/> <module-option
            name="java.naming.security.authentication" value="simple"/> <module-option name="principalDNPrefix"
            value="uid="/> <module-option name="principalDNSuffix" value=",
            ou=people, dc=redhat, dc=com"/>
            <module-option name="rolesCtxDN" value="ou=Roles,dc=redhat,dc=com"/> <module-option
            name="uidAttributeID" value="member"/> <module-option
            name="matchOnUserDN" value="true"/> <module-option
            name="roleAttributeID" value="cn"/> <module-option
            name="roleAttributeIsDN" value="false"/> </login-module> </authentication>


  </security-domain>
```

Guided Exercise: Configuring the LDAP Login Module

In this lab work, you will use an LDAP security scheme to secure an application.

Resources	
Files:	/home/student/JB248/labs/security-ldap
App URL:	http://127.0.0.1:8080/guessLDAP

Results

You must be able to enable authentication in an application using an LDAP login module.

before you start

Before beginning the guided exercise, run the following command to verify that EAP was installed to /opt/jboss-eap-7.0 and that no EAP instances are running, and to download the lab files:

```
[student@workstation ~]$ lab securing-ldap setup
```

1. **Start the standalone instance of EAP by running the following command with the base directory located at /home/student/JB248/labs/standalone:**

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ ./standalone.sh \
-Djboss.server.base.dir=/home/student/JB248/labs/standalone/
```

2. **The guessLDAP application**

The guessLDAP application is already configured to require authentication. See the following configuration files to see how it requires authentication.

2.1. **In the home/student/JB248/labs/security-ldap folder, there is a web application named guessLDAP.war. Extract the application with the following command:**

```
[student@workstation bin]$ cd /home/student/JB248/labs/security-ldap [student@workstation
security-ldap]$ jar -xvf guessLDAP.war
```

2.2. **Browse the jboss-web.xml file located at /home/student/JB248/labs/security-ldap/WEB-INF/.**

```
 ...
<jboss-web>
              <security-domain>jb248_ldap</security-domain> </jboss-
web>
```

security-domain points to a security domain used by the application, which will be named jb248_ldap.

3. Configure the LDAP-based security domain.

3.1. An LDAP server is running on the workstation virtual machine on port 389. Run the following command to run an ldapsearch on the workstation virtual machine in a new terminal window:

```
[student@workstation ~]$ ldapsearch -x -D "cn=Manager,dc=redhat,dc=com" \ -w 43etq5 -b
"dc=redhat,dc=com"
```

You should see the output for LDAP users and groups with the following at the end of the output:

```
...OUTPUT OMITTED
#numResponses: 13
#numEntries: 12
```

3.2. Start the EAP CLI with the following commands in a new terminal window. Enter jbossadm as the username and JBoss@RedHat123 as the password.

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin [student@workstation
bin]$ ./jboss-cli.sh --connect
```

3.3. Use the following commands to create the security domain named jb248_ldap:

```
[standalone@localhost:9990] /subsystem=security/security-domain\ =jb248_ldap:add(cache-
type=default) [standalone@localhost:9990] /
subsystem=security/security-domain\ =jb248_ldap/authentication=classic:add
```

3.4. Create an LDAP login module within the new security domain with the following characteristics: • Name: ldap_login

• Code: Ldap

Pay attention to enter the exact code as it is case sensitive.

• Module options:

```
java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory java.naming.provider.url=ldap://
localhost:389 java.naming.security.authentication=simple
principalDNPrefix=uid= principalDNSuffix=,ou=people,
dc=redhat, dc=com
rolesCtxDN=ou=Roles, dc=redhat, dc=com uidAttributeID=member
matchOnUserDN=true roleAttributeID=cn
roleAttributeIsDN=false
```

Use the following command to create the security domain:

```
[standalone@localhost:9990] /subsystem=security/security-domain=jb248_ldap/\
authentication=\classic/login-module=ldap_login:add( \ code=Ldap, \
  flag=required,
  \ module-options=[ \

  ("java.naming.factory.initial"=>"com.sun.jndi.ldap.LdapCtxFactory"), \
   ("java.naming.provider.url"=>"ldap://localhost:389"), \
   ("java.naming.security.authentication"=>"simple"), \
   ("principalDNPrefix"=>"uid="), \
   ("principalDNSuffix"=>",ou=people, dc=redhat, dc=com"), \
   ("rolesCtxDN"=>", ou=Roles, dc=redhat, dc=com"), \
   ("uidAttributeID"=>"member"), \
   ("matchOnUserDN"=>"true"), \
   ("roleAttributeID"=>"cn"), \
   ("roleAttributeIsDN"=>"false") \ ])
```

use

This command can be copied and pasted from /home/student/JB248/labs/ security-ldap/ldap-login-module.

3.5. Enter the following command in the CLI to verify the domain values of jb248_ldap security:

```
[standalone@localhost:9990] /subsystem=security/security-domain=\ jb248_ldap:read-
resource(recursive=true)
```

You should see the following output:

```
{
        "outcome" => "success",
        "result" => {
                ...
                "authentication" => {"classic" => { "login-modules"
                        => [{ "name" => "ldap_login",
                                "code" => "Ldap", "flag" =>
                                "required", "module"
                                => undefined, "module-
                                options" =>
                                [ ("java.naming.factory.initial"
                                        => "com.sun.jndi.ldap.LdapCtxFactory"),

                                        ("java.naming.provider.url" => "ldap://localhost:389"),
                                        ("java.naming.security.authentication" => "simple"), ("principalDNPrefix"
                                        => "uid="), ("principalDNSuffix" =>
                                        ",ou=people, dc=redhat, dc=com"), ("rolesCtxDN" =>
                                        "ou=Roles,dc=redhat,dc=com"), ("uidAttributeID" => "member"),
                                        ("matchOnUserDN" => "true"),
                                        ("roleAttributeID" => "cn"),
                                        ("roleAttributeIsDN" => "false")


                                ]
                        }],
               ...
        }
}
```

3.6. Reload the server to allow the changes to take effect.

```
[standalone@localhost:9990] :reload
```

4. Test the LDAP-based security domain.

4.1. Using the administrative console or the CLI, deploy the guessLDAP.war file located at /home/student/JB248/labs/security-ldap/.

```
[standalone@localhost:9990 /] deploy \ /home/
student/JB248/labs/security-ldap/guessLDAP.war
```

4.2. On workstation, go to http://localhost:8080/guessLDAP. you will should prompt you to login if login module was configured correctly.

4.3. Enter "bt1" as username and "ldap1" as password; You should log in with no problem.

If the authentication is successful, you should see the Guess app.

4.4. Uninstall the guessLDAP application from the standalone server using the CLI:

```
[standalone@localhost:9990] undeploy guessLDAP.war
```

4.5. Exit the CLI and stop the running EAP instance.

This concludes the guided exercise.