**Setup**

1. Configure the LDAP plugin by editing *$SONARQUBE-HOME/conf/sonar.properties* (see table below)
2. Restart the SonarQube server and check the log file for:
3. INFO org.sonar.INFO Security realm: LDAP ...
4. INFO o.s.p.l.LdapContextFactory Test LDAP connection: OK
5. Log into SonarQube
6. On logout users will be presented a login page (*/sessions/login*), where they can choose to login as technical user or a domain user by passing appropriate credentials

From SonarScanners, we recommend using [local technical users](https://docs.sonarqube.org/7.9/instance-administration/security/) for authentication against SonarQube Server.

**General Configuration**

| **Property** | **Description** | **Default value** | **Required** | **Example** |
| --- | --- | --- | --- | --- |
| sonar.security.realm | Set this to LDAP authenticate first against the external sytem. If the external system is not reachable or if the user is not defined in the external system, authentication will be performed against SonarQube's internal database. | none | Yes | LDAP (only possible value) |
| sonar.authenticator.downcase | Set to true when connecting to a LDAP server using a case-insensitive setup. | false | No |  |
| ldap.url | URL of the LDAP server. If you are using ldaps, you should install the server certificate into the Java truststore. | none | Yes | ldap://localhost:10389 |
| ldap.bindDn | The username of an LDAP user to connect (or bind) with. Leave this blank for anonymous access to the LDAP directory. | none | No | cn=sonar,ou=users,o=mycompany |
| ldap.bindPassword | The password of the user to connect with. Leave this blank for anonymous access to the LDAP directory. | none | No | secret |
| ldap.authentication | Possible values: simple, CRAM-MD5, DIGEST-MD5, GSSAPI. See [the tutorial on authentication mechanisms](http://java.sun.com/products/jndi/tutorial/ldap/security/auth.html) | simple | No |  |
| ldap.realm | See [Digest-MD5 Authentication](http://java.sun.com/products/jndi/tutorial/ldap/security/digest.html), [CRAM-MD5 Authentication](http://java.sun.com/products/jndi/tutorial/ldap/security/crammd5.html) | none | No | example.org |
| ldap.contextFactoryClass | Context factory class. | com.sun.jndi.ldap.LdapCtxFactory | No |  |
| ldap.StartTLS | Enable use of StartTLS | false | No |  |
| ldap.followReferrals | Follow referrals or not. See [Referrals in the JNDI](http://docs.oracle.com/javase/jndi/tutorial/ldap/referral/jndi.html) | true |  |  |

**User Mapping**

| **Property** | **Description** | **Default value** | **Required** | **Example for Active Directory** |
| --- | --- | --- | --- | --- |
| ldap.user.baseDn | Distinguished Name (DN) of the root node in LDAP from which to search for users. | None | Yes | cn=users,dc=example,dc=org |
| ldap.user.request | LDAP user request. | (&(objectClass=inetOrgPerson)(uid={login})) | No | (&(objectClass=user)(sAMAccountName={login})) |
| ldap.user.realNameAttribute | Attribute in LDAP defining the user’s real name. | cn | No |  |
| ldap.user.emailAttribute | Attribute in LDAP defining the user’s email. | mail | No |  |

**Group Mapping** Only [groups](http://identitycontrol.blogspot.fr/2007/07/static-vs-dynamic-ldap-groups.html) are supported (not [roles](http://identitycontrol.blogspot.fr/2007/07/static-vs-dynamic-ldap-groups.html)). Only [static groups](http://identitycontrol.blogspot.fr/2007/07/static-vs-dynamic-ldap-groups.html) are supported (not [dynamic groups](http://identitycontrol.blogspot.fr/2007/07/static-vs-dynamic-ldap-groups.html)).

For the delegation of authorization, [groups must be first defined in SonarQube](https://docs.sonarqube.org/7.9/instance-administration/security/). Then, the following properties must be defined to allow SonarQube to automatically synchronize the relationships between users and groups.

| **Property** | **Description** | **Default value** | **Required** | **Example for Active Directory** |
| --- | --- | --- | --- | --- |
| ldap.group.baseDn | Distinguished Name (DN) of the root node in LDAP from which to search for groups. | none | No | cn=groups,dc=example,dc=org |
| ldap.group.request | LDAP group request. | (&(objectClass=groupOfUniqueNames)(uniqueMember={dn})) | No | (&(objectClass=group)(member={dn})) |
| ldap.group.idAttribute | Property used to specifiy the attribute to be used for returning the list of user groups in the compatibility mode. | cn | No | sAMAccountName |

**Sample Configuration**

# LDAP configuration

# General Configuration

sonar.security.realm=LDAP

ldap.url=ldap://myserver.mycompany.com

ldap.bindDn=my\_bind\_dn

ldap.bindPassword=my\_bind\_password

# User Configuration

ldap.user.baseDn=ou=Users,dc=mycompany,dc=com

ldap.user.request=(&(objectClass=inetOrgPerson)(uid={login}))

ldap.user.realNameAttribute=cn

ldap.user.emailAttribute=mail

# Group Configuration

ldap.group.baseDn=ou=Groups,dc=sonarsource,dc=com

ldap.group.request=(&(objectClass=posixGroup)(memberUid={uid}))

**Advanced LDAP Topics**

**Authentication Methods**

* **Anonymous** - Used when only read-only access to non-protected entries and attributes is needed when binding to the LDAP server.
* **Simple** Simple authentication is not recommended for production deployments not using the ldaps secure protocol since it sends a cleartext password over the network.
* **CRAM-MD5** - The Challenge-Response Authentication Method (CRAM) based on the HMAC-MD5 MAC algorithm ([RFC 2195](http://tools.ietf.org/html/rfc2195)).
* **DIGEST-MD5** - This is an improvement on the CRAM-MD5 authentication method ([RFC 2831](http://www.ietf.org/rfc/rfc2831.txt)).
* **GSSAPI** - GSS-API is Generic Security Service API ([RFC 2744](http://www.ietf.org/rfc/rfc2744.txt)). One of the most popular security services available for GSS-API is the Kerberos v5, used in Microsoft's Windows 2000 platform.

For a full discussion of LDAP authentication approaches, see [RFC 2829](http://www.ietf.org/rfc/rfc2829.txt) and [RFC 2251](http://www.ietf.org/rfc/rfc2251.txt).

**Multiple Servers**

To configure multiple servers:

# List the different servers

ldap.servers=server1,server2

# Configure server1

ldap.server1.url=ldap://server1:1389

ldap.server1.user.baseDn=dc=dept1,dc=com

...

# Configure server2

ldap.server2.url=ldap://server2:1389

ldap.server2.user.baseDn=dc=dept2,dc=com

...

Authentication will be tried on each server, in the order they are listed in the configurations, until one succeeds. User/Group mapping will be performed against the first server on which the user is found.

Note that all the LDAP servers must be available while (re)starting the SonarQube server.

Web.log to be checked for LDAP connection status