Configuring Security

Overview

Artifactory's security model offers protection at several levels. It allows you to do the following:

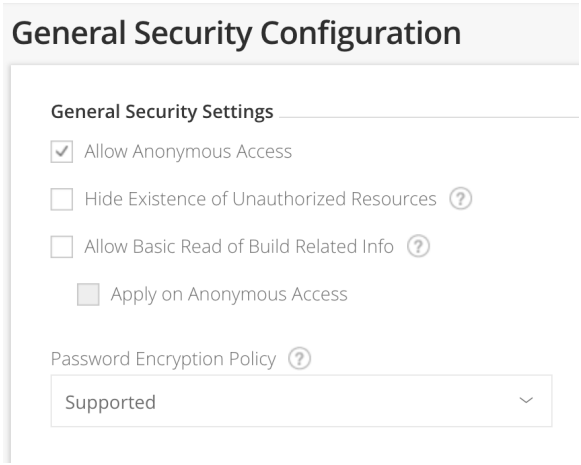
- Assign role-based or user-based permissions to areas in your repositories (called Permission Targets)
- Allow sub-administrators for Permission Targets
- Configure LDAP out-of-the-box
- Prevent clear text in Maven's `settings.xml` file
- Inspect security definitions for a single artifact or folder and more.

Artifactory's security is based on Spring Security and can be extended and customized.

This section explains the strong security aspects and controls offered by Artifactory.

General Configuration

Artifactory provides several system-wide settings to control access to different resources. These are found under **Security | General** in the **Administration** tab.



Page Contents

Read more

- Managing Users
- Managing Permissions
- Centrally Secure Passwords
- Artifactory Key Encryption
- Managing Security with LDAP
- Managing Security with Active Directory
- Managing Certificates
- Using a Self-Signed Certificate
- Access Tokens

Allow Anonymous Access

Artifactory provides a detailed and flexible permission-based system to control users' access to different features and artifacts.

However, Artifactory also supports the concept of "Anonymous Access" which controls the features and artifacts available to a user who has not logged in.

This is done through an "Anonymous User" which comes built-in to Artifactory with a default set of permissions.

Anonymous access may be switched on (default) or off using the **Allow Anonymous Access** setting under **Security General Settings** in the **Admini stration** module**.**

You can modify the set of permissions assigned to the "Anonymous User" just like you would for any other user, and this requires that **Allow Anonymous Access** is enabled.

Allow Basic Read of Build Related Info

This setting gives all users view permissions to published modules for all builds in the system. This is regardless of any specific permissions applied to a particular build. And only applies to anonymous users if the "**Apply on Anonymous Access**" is enabled.

Hide Existence of Unauthorized Resources

When a user tries to access a resource for which he is not authorized, Artifactory's default behavior is to indicate that the resource exists but is protected.

For example, an anonymous request will result in a request for authentication (401), and a request by an unauthorized authenticated user will simply be denied (403).

You can configure Artifactory to return a 404 (instead of 403) - Not Found response in these cases by setting **Hide Existence of Unauthorized Resources** under **Security | General** in the **Administration** module.

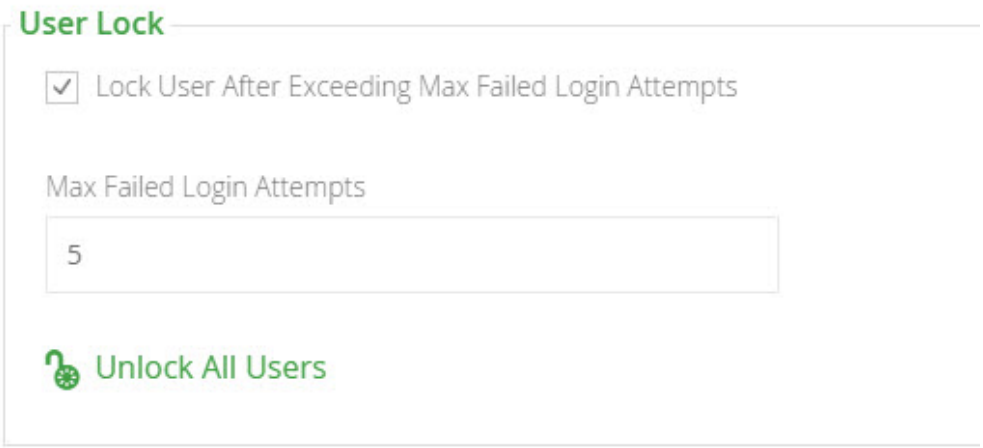
Password Encryption Policy

Artifactory provides a unique solution to support encrypted passwords through the **Password Encryption Policy** setting as follows:

Supported	Artifactory can receive requests with an encrypted password but will also accept requests with a non-encrypted password (default)
Required	Artifactory requires an encrypted password for every authenticated request
Unsupported	Artifactory will reject requests with encrypted password

For more details on why Artifactory allows you to enforce password encryption please refer to Centrally Secure Passwords.

User Lock and Login Suspension



User account locking and temporary login suspension are two mechanisms employed by Artifactory to prevent identity theft via brute force attack.

Temporary Login Suspension

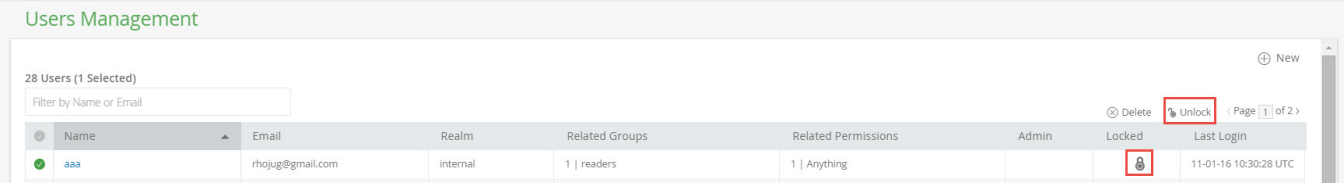
Temporary login suspension means that when a login attempt fails due to incorrect authentication credentials being used, Artifactory will temporarily suspend that user's account for a brief period of time during which Artifactory ignores additional login attempts. If login attempts fail repeatedly, Artifactory will increase the suspension period each time until it reaches a maximum of 5 seconds.

User Account Locking

In addition to temporary login suspension, you can configure Artifactory to lock a user's account after a specified number of failed login attempts. This is enabled by checking "Lock User After Exceeding Max Failed Login Attempts", and specifying the **Max Failed Login Attempts** field. Users who get locked out of their account because they have exceeded the maximum number of failed login attempts allowed (as specified in **Max Failed Login Attempts)** must have an administrator access to unlock their account.

Unlocking User Accounts

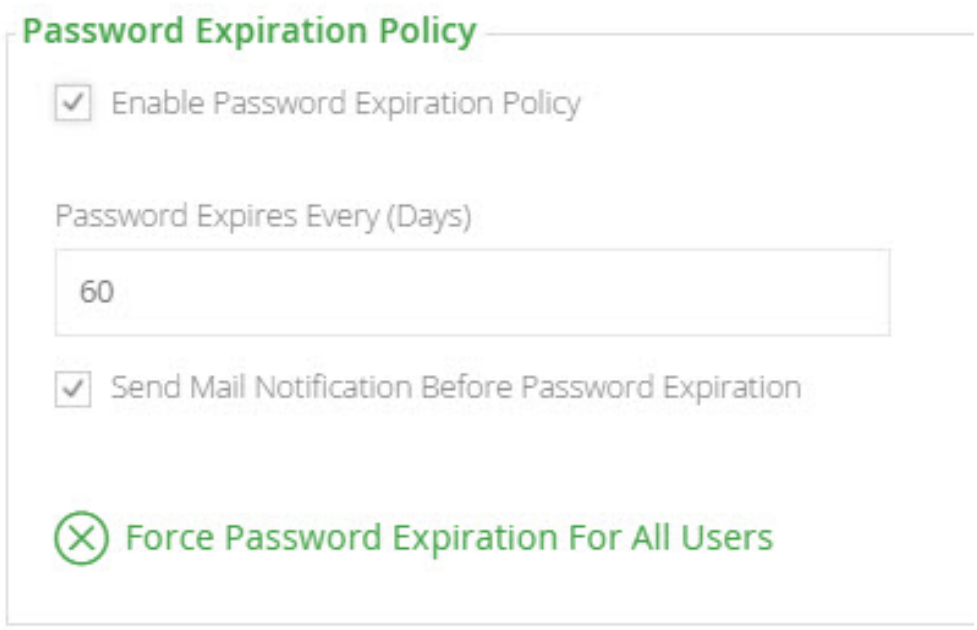
An Artifactory administrator can unlock all locked-out users using the "Unlock All Users" button under **Security General Configuration** screen where user locking is configured. An administrator can also unlock a specific user or a group of users in the **Security Module** under **User Management.**



Through the REST API, an administrator can unlock a single user, a group of users or all locked-out users at once.

Password Expiration Policy

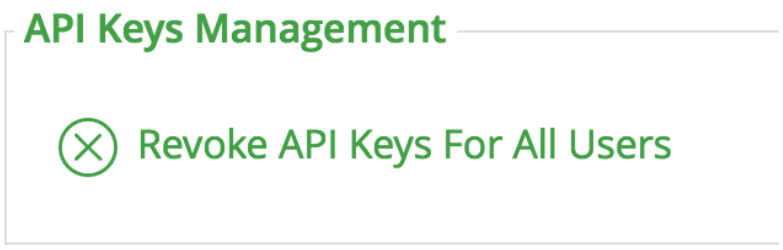
Artifactory lets an admin user enforce a password expiration policy that forces all users to change their passwords at regular intervals. When the password expiration policy is enforced, users who do not within the specified time interval will be locked out of their accounts until they change their password.



Enable Password Expiration Policy	When checked, password expiration policy is enabled.
Password Expires Every (Days)	Specifies how frequently all users must change their password.
Send Mail Notification Before Password Expiration	When checked, users receive an email notification a few days before their password expires.
Force Password Expiration For All Users	Forces all passwords to expire. All users will have to change their password at next login.

Managing API Keys

As an admin user, you can revoke all the API keys currently defined in the system under **Security | General** in the **Administration** module.



To revoke all API keys in the system, click "Remove API Keys for All Users".

To revoke a specific user's API key, navigate to **Administration** module >> **Security | Users** and select the relevant user to edit . Once in the edit screen one of the available actions is "Revoke API key"

> Once you revoke an API key, any REST API calls using that API key will no longer work. The user will have to create new API key and update any scripts that use it.

Passwords Encryption

Different configuration files in Artifactory may include password information stored in plain text.

To keep passwords secure, you may choose to encrypt them as described in Artifactory Key Encryption.

CSRF Protection

From version 5.11, Artifactory can prevent CSRF attacks by using a new custom header, `X-Requested-With,` for internal UI calls. This feature may require modification to your proxy server (if you are using one) to make sure the proxy does not filter out this header.

In version 5.11, the CSRF Protection is disabled by default. To **enable** the CSRF Protection, add the `artifactory.csrf.filter.enabled = true` flag in the *`artifactory.system.properties`* file under the *`$ARTIFACTORY_HOME/etc`* folder and restart Artifactory to apply the change.

From version 6.0 and onwards, the CSRF Protection is enabled by default and cannot be disabled.

Hardening Security for Secrets

Artifactory uses a set of encrypted parameters (secrets) used to connect to external resources such as the different databases it uses. While these secrets may be stored in the Artifactory configuration file, this poses a risk of their being exposed.

To keep secrets safe from exposure, **from version 6.6**, you may pre-load secrets from a temporary file when you startup Artifactory. Once Artifactory has read and successfully used the secrets, the file is deleted.

The snippet below shows an example of the parameters you could include in this temporary file. These are the parameters Artifactory uses to connect to a PostgreSQL database.

```
type=postgresql
driver=org.postgresql.Driver
url=jdbc:postgresql://postgresql:5432/artifactory
username=artifactory
password=JE2cyPQtEmJovMbxwEGrghre9EXcu4ANtTtPu9Lk3s15UPs73M
```

While we recommend only including sensitive information such as encrypted connection strings, this file may contain any of the database configuration parameters, and any parameters specified (including environment variables and system properties) will override the corresponding ones in the database configuration file.

To load parameters using this mechanism, place them in the following temporary file before your startup Artifactory:

`$ARTIFACTORY_HOME/etc/.secrets/.temp.db.properties`

> **Execute on every restart of Artifactory**
>
> Since the temporary file is deleted when Artifactory starts, you need to replace the temporary file each time you restart Artifactory.