# Securing Jenkins

This page has been superseded by the "Securing Jenkins" section of the Jenkins User Handbook.

In the default configuration of Jenkins 1.x, Jenkins does not perform any security checks. This means the ability of Jenkins 1.x to launch processes and access local files is available to anyone with access to Jenkins.

Securing Jenkins has several aspects to it.

## Access Control

You should lock down the access to Jenkins UI so that users are authenticated and appropriate set of permissions are given to them. This setting is controlled mainly by two axes:

- **Security Realm**, which determines users and their passwords, as well as what groups the users belong to.
- **Authorization Strategy**, which determines who has access to what.

These two axes are orthogonal, and need to be individually configured. For example, you might choose to use external LDAP or Active Directory as the security realm, and you might choose "everyone full access once logged in" mode for authorization strategy. Or you might choose to let Jenkins run its own user database, and perform access control based on the permission/user matrix.

The following pages discuss various aspects of this feature in details

- Quick and Simple Security — if you are running Jenkins like "`java -jar jenkins.war`" and only need a very simple set up
- Standard Security Setup — discusses the most common set up of letting Jenkins run its own user database and do finer-grained access control
- Apache frontend for security — run Jenkins behind Apache and perform access control in Apache instead of Jenkins
- Authenticating scripted clients — if you need to programatically access security-enabled Jenkins web UI, use BASIC auth
- Help! I locked myself out! — if something goes really wrong and you can't get full access anymore
- Matrix-based security — Granting and denying finer-grained permissions

## Protect users of Jenkins from other threats

There are additional security subsystems in Jenkins that protect Jenkins and users of Jenkins from indirect attacks.

The following topics discuss features that are **off by default**. We recommend you read them first and act on them.

- CSRF Protection — prevent a remote attack against Jenkins running inside your firewall
- Security implication of building on master — protect Jenkins master from malicious builds
- Slave To Master Access Control — protect Jenkins master from malicious build agents
- Securing JENKINS_HOME — protect Jenkins from users with local access

The following topics discuss other security features that are on by default. You'll only need to look at them when they are causing problems.

- Configuring Content Security Policy — protect users of Jenkins from malicious builds
- Markup formatting — protect users of Jenkins from malicious users of Jenkins