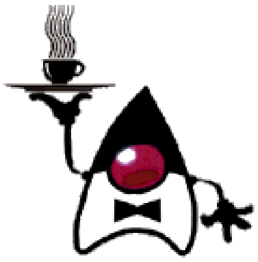

The Java Web Server Documentation Pages

Java Web Server Access Control Lists (ACLs)

Introduction

Java Web Server has an extensible ACL framework that is used for controlling client requests for files and servlets. This framework can also be used by servlets to implement their own ACL mechanism. Java Web Server allows for the configuration of different authentication schemes and different ACLs for the various files, directories and servlets that Java Web Server serves.

Authentication schemes

The following authentication schemes are supported: Basic HTTP authentication, Digest authentication, and SSL. Each file, directory tree or servlet being served can be configured to use a particular authentication scheme.

Access Control List (ACL) files

An ACL file specifies the security model desired while serving a group of files or servlets. The ACL file specifies the users that are allowed to access the object that is being guarded by the ACL. The objects that can have an ACL are:

- Any file that is served by Java Web Server can have an ACL associated with it. If a file is not explicitly protected, it inherits the ACL from its parent directory recursively. If there is no ACL for the entire tree, access is allowed by default.

- Each directory tree served by Java Web Server in the document tree can be associated with an ACL file. If a directory is not explicitly protected, it inherits the ACL from its parent directory.
- Any servlet that is served by Java Web Server can be associated with an ACL file. If there is no ACL explicitly associated with a servlet, the one associated with Servlet.* is used. If there is none, access is allowed by default.

Access Control Configuration(ACC) file

An ACC file can be associated with the server using the httpd property "acl.config". The value of this property must be a server root relative file name that describes the access configuration. Access control can be enabled or disabled using the httpd property "enable.acls". The ACC file associates protected objects with authentication and authorization configuration. The ACC file is server root relative if it is not an absolute path. If there is no file explicitly associated with the server, authentication and access control checks are completely disabled at the server. The syntax of the ACC file is that of a properties file. The basic syntax is as shown

<protectedObject>=<scheme>:<realm>:<ACL name>

protectedObject
: This is the object that is protected. In Java Web Server, this can be a file name, directory name or name of a servlet. File and directory names must be absolute or relative to the server root. Servlets are named as Servlet.<servletName> Default protection for all servlets are specified by specifying "*" for the <servletName>.

scheme
: This is the authentication scheme to be used. Currently only "Basic" is allowed. In the future, others will be supported.

realm
: This is the realm under which this object is protected. The realm defines the scope of the authentication and authorization. For each realm, there must be a directory with the same name as the name of the realm under the <serverRoot>/admin/realms/ directory. This directory with the realm name should contain a "keyfile" file that lists all the users and their passwords in this realm. For each group that is listed in the ACLs belonging to this realm should have file identified by the group name in this directory. Also all ACL files are to be founded in this directory.

ACL name
: The ACL name corresponds to a file in the realm directory that describes a list of entries. Each entry associates a principal or a group with a list of permissions. An entry can also specifically take away privileges rather than grant them. The ACL file syntax is described in more detail in the next section

If a file, directory or servlet is not explicitly protected, the default protection mode is to allow access to everyone. An example ACC file is shown below:

```
/opt/web/jeeves/public/tree1/tree2/=Basic:basicRealm:aclFile.1
```

```
/opt/web/jeeves/public/tree1/fileA=Basic:basicRealm:aclFile.2
Servlet.phoneServlet=Basic:basicRealm:aclFile.2
Servlet.*=unprotected
/=Basic:basicRealm:aclFile.root
```

Access Control List(ACL) file syntax

An ACL file is a list of entries in the Java properties format. Each entry describes the permissions allowed for a principal or a group. An entry can also describe the permissions that are explicitly disallowed for a principal or a group. So the syntax of an entry in the ACL file is:

<sign><attribute>.<name>=<permissions>

sign
: The sign must be a "+" or a "-". A "+" sign or no sign indicates additive permissions. Entries that have additive permission list the permissions allowed while entries that begin with a "-" list permissions that are explicitly disallowed. So if a group is given a set of permissions, some members of the group can be specifically disallowed access that other members of the groups have.

attribute
: The attribute must be either "User" or "Group". If attribute is "User", it indicates that the name that follows the period is the name of a user, otherwise it indicates that the name that follows the period is the name of a group.

name
: The name of the principal or group for which permissions are listed. A special value of "world" within the context of a "Group" attribute indicates that this is a group that comprises of all the principals in all the realms.

permissions
: A comma separated list of permissions. No white spaces are allowed. The current list of supported permissions (in Java Web Server) are GET and POST. A special value "all" or "*" is used to indicate all the allowable permissions.

An example ACL file is shown below:

```
+Group.world=GET
+Group.adminGroup=all
-User.satish=POST
```

Host name based access control

In addition to the above, Java Web Server allows host name based accesss. An ACL file is also allowed to have a special syntax to specify the list of hosts that are allowed to access the object being proteced. The syntax is:

```
AllowedHosts=eng.sun.com,*.transarc.com,*.edu
```

A fully specified hostname or a sub-domain can be specified on the right hand side. Request that originate from somewhere other than that specified in the AllowedHosts attribute are disallowed.