VITELS - HOWTO

Apache Authentication with LDAP

Thomas Jampen
University of Bern
Neubrückstrasse 10
3012 Bern
+41 31 631 86 92
jampen@iam.unibe.ch

1 Apache Authentication Based on LDAP

The Apache webserver [1] offers the possibility to restrict access to certain webpages with `.htaccess` files. These files can be placed within the directories to be protected from unauthorized access.

Usually, these files refer to a password file which looks similar to a Unix standard `passwd` or `shadow` file. The password file contains usernames and the corresponding encrypted password. Every time someone opens a browser in order to access webpages that are located within a protected directory he must enter his username and password in order to authenticate himself to the webserver. Once authenticated, the user can access all restricted directories and subdirectories without being prompted for the password again. The session holds as long as the browser is open.

If an LDAP server is running, the Apache webserver can use an LDAP module in order to access an LDAP directory. The user authentication on websites can be done with the help of the LDAP directory instead of the password file and, thus, it is possible to reduce the administrational overhead of maintaining an additional password file.

The proposed LDAP authentication module for Apache is called `auth_ldap` and was written by Dave Carrigan [3]. `auth_ldap` is designed to have excellent performance and to support Apache version 1.3.x on both Unix and Windows. The use of this module is recommended as its configuration is easy and plain, although there are several other modules available on the web [2]. Furthermore, the new version of Apache - Apache version 2.0.x - comes with integrated LDAP authentication support. The directives used there are exactly the same as the ones described below. Thus, the other modules are not recommended as they use another syntax.

1.1 Installation

If the `auth_ldap` module is intended to be installed on a Debian/GNU Linux system using a pre-compiled Apache webserver, the appropriate package `libapache-auth-ldap` can be downloaded from the Debian homepage [4]. The following command automatically installs the package:

```
debian:~# dpkg -i libapache-auth-ldap_version_system.deb
```

It is also possible to use `dselect` or `apt-get`. Dependency problems with other packages will be displayed automatically.

If the operating system is not Debian/GNU Linux, there is a `tar.gz` archive on the module's website [3] that can be downloaded and extracted. Further instructions for compilation can be found on this webpage as well.

1.2 Apache Configuration

In order to load the module, the appropriate `LoadModule` directive has to be added to the `httpd.conf` file:

```
LoadModule  auth_ldap_module  /usr/lib/apache/auth_ldap.so
```

Finally, the webserver has to be restarted using the command:

```
debian:~# /etc/init.d/apache restart
```

It has to be considered that the `.htaccess` files can only be used if the directives specified within the Apache configuration file are allowed to be overridden by the `.htaccess` files. This can be achieved by changing the parameter `AllowOverride` in the `httpd.conf` file from `None` to `AuthConfig`.

1.3 .htaccess Files

An `.htaccess` file is a special file that can be placed within any directory below the directory specified by `DocumentRoot`. This file prevents the directory and all its subdirectories from being accessed without successful authentication. A simple `.htaccess` file looks as follows:

file: **.htaccess**

```
AuthType Basic
AuthName "Restricted Area"
AuthLDAPURL ldap://webct.unibe.ch:389/ou=users,o=Universitaet Bern,c=CH?uid
require valid-user
```

Most often `Basic` can be used for the `AuthType` and any quoted string for `AuthName`. This string is displayed when the user is prompted for username and password. These two directives are mandatory. The next line contains the LDAP server and the port followed by the base dn from where the search will be started. After an interrogation mark, the attribute to be used for the authentication is specified. Finally, a `require` directive is needed in order to indicate which users should be accepted. In this example `valid-user` is used, which means that every user can access the page if he provides a valid username and password.

1.4 LDAP Directives

The most important directives are listed and described here. For further details refer to the `auth_ldap` homepage [3].

AuthLDAPBindDN and AuthLDAPBindPassword can be specified if the module has to authenticate itself before performing the search.

AuthLDAPDereferenceAliases allows to specify how aliases should be dereferenced during the LDAP operations. `never`, `always`, `searching` and `finding` are allowed. Default is `always`.

AuthLDAPEnabled allows to completely disable LDAP authentication when set to `off`. This can be useful if `auth_ldap` is used but needed to be disabled within certain subdirectories.

AuthLDAPGroupAttribute specifies the attribute used by the LDAP server to store members of a group. This directive can be specified multiple times. Default values are `member` and `uniquemember`.

AuthLDAPUrl specifies the LDAP search parameters to be used. The format is the following: `ldap://server:port/basedn?attribute?scope?filter`. The values `server` and `port` specify the LDAP server to use for the search. Default is `localhost` and `389`, respectively. In order to list multiple servers they can be separated by spaces. `basedn` specifies the base dn for all LDAP searches. `attribute` determines the attribute that should be compared with the supplied username. The default is `uid`. `scope` specifies the scope to be used for LDAP searches. Possible values are `one` or `sub`. Default is `sub`. `filter` allows to specify a valid LDAP filter. Default is `(objectClass=*)`.

Below the authentication directives, the following authorization directives can be specified:

require valid-user allows any user that is successfully authenticated.

require user specifies a list of allowed user. A user is defined by the attribute used in the `AuthLDAPUrl`.

require group allows access to any successfully authenticated user of the given group. The group members must be listed in the attribute given by `AuthLDAPGroupAttribute`. Note: The group dn must be a full dn.

require dn allows to specify distinguished names of the allowed users.

All the above described directives can be entered in a `.htaccess` file or within a `<Location /some/directory>` directive of the `httpd.conf` file.

1.5 Usage of auth_ldap for the VITELS Project

The `auth_ldap` module allows to dereference aliases and, thus, can be configured to check the `ou=Modules,o=VITELS,c=CH` directory subtree. This allows the use of Apache authentication in order to restrict access to certain webpages to the currently registered student. A sample `.htaccess` file looks as follows:

file: **.htaccess**

```
AuthType Basic
AuthName "Restricted Area"
AuthLDAPDereferenceAliases finding
AuthLDAPEnabled on
AuthLDAPURL ldap://webct.unibe.ch:389/mid=6,ou=Modules,o=VITELS,c=CH?uid
require valid-user
```

Alias dereferencing is set to `finding` in order to dereference already the base object because the aliased entry is directly specified in the base dn `mid=6,ou=Modules,o=VITELS,c=CH`. Then,

authentication is enabled. After that, the LDAP server and the port are specified, followed by the base dn - pointing to the module to be accessed - and the attribute to search for (`uid`).

This `.htaccess` file can be placed in the directory containing the webpages that load the Java SSH Applets. This prevents unregistered users from accessing the course and the Applet.

Bibliography

[1] Apache HTTP Server Project.
http://httpd.apache.org/.

[2] Apache Module Registry.
http://modules.apache.org.

[3] An Authentication Module for Apache.
http://www.rudedog.org/auth_ldap/.

[4] LDAP Authentication Module for Apache.
http://packages.debian.org/testing/interpreters/libapache-auth-ldap.h%tml.