SubVersion Server with LDAP Authentication via Fedora Directory Server  
(Last Revision 07052007 by Ashley Chew)

Background Information

This document is about the setup of a Web Based SubVersion Server on Fedora Core Distribution using LDAP authentication via Fedora Directory Server.

Requirements

Now I’m assuming you have a functional Fedora Directory Server which is essentially an LDAP server (jhett.csse.uwa.edu.au as with my other documentation) and a Fedora Core 6 Web box (I’ve called my web box svn.csse.uwa.edu.au which is running FC6 with Apache).

Before Proceeding make sure your FC6 Web box is functional such that can do HTTP authentication via LDAP (LDAP htaccess & Apache), if not read “Using Fedora Directory Server and HTTP authentication via LDAP” guide first and get that functioning before proceeding.

Note this is based on the instructions by Martin Tomes based here http://www.subversionary.org/howto/setting-up-a-server-on-fedora-core-4 and http://www.ferdychristant.com/blog/articles/DOMM-6NFJ6J but I have adapted it to LDAP authentication and specifically for Fedora Core 6.

Instructions

Now before proceeding you need to have subversion and the subversion module package for apache installed. Ie on my system.

```
[ashley@svn] % rpm -qa |grep subver  
subversion-1.4.3-2.fc6  
[ashley@svn] % rpm -qa |grep mod_dav_svn  
mod_dav_svn-1.4.3-2.fc6
```

If you don’t have these packages download them then install them ie

```
 rpm –ivh subversion-1.4.3-2.fc6.i386.rpm mod_dav_svn-1.4.3-2.fc6.i386.rpm
```
or yum install them ie

yum install subversion
yum install mod_dav_svn

If there are any dependencies you have to download those packages as well, once installed just re-query your system to make sure it is installed.

Now subversion needs an area to store its information ie the repository information for users and groups and the permissions for each repository created which are managed by the svn server.

Lets say the repository will be situated in /svn for example.

mkdir /svn
mkdir /svn/repos
mkdir /svn/permissions

The next thing we would change the ownership of the files to user running Apache. Usually the username and group ownership are apache ie you can verify this

```
[root@svn] % /etc/init.d/httpd restart
[root@svn] % ps -aux |grep -i httpd
```

Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.7/FAQ

As you can see the httpd is run by the user apache. As Subversion will be run by the httpd process we have to change access to /svn and all subdirectories to apache as everything will be managed by SubVersion Server hence apache which needs ownership of the area.

```
chown –R apache.apache /svn
```

Now you have created the main holding area for SubVersion Server we have to enable it on the apache with the svn module. In Fedora Core 6 it’s located in

/etc/httpd/conf.d/subversion.conf. Apache in Fedora will source Source configurations files from /etc/httpd/conf.d

```
[root@svn conf.d]# pwd
```
LoadModule dav_svn_module modules/mod_dav_svn.so
LoadModule authz_svn_module modules/mod_authz_svn.so

# Example configuration to enable HTTP access for a directory containing Subversion repositories, "/var/www/svn". Each repository must be readable and writable by the 'apache' user. Note that if SELinux is enabled, the repositories must be labelled with a context which httpd can write to; this will happen by default for directories created in /var/www. Use "restorecon -R /var/www/svn" to label the repositories if upgrading from a previous release.
#

# To create a new repository "http://localhost/repos/stuff" using this configuration, run as root:
#
#   # cd /var/www/svn
#   # svnadmin create stuff
#   # chown -R apache.apache stuff
#

<Location /repos>
   DAV svn
   SVNParentPath /var/www/svn
   
   # Limit write permission to list of valid users.
   <LimitExcept GET PROPFIND OPTIONS REPORT>
      # Require SSL connection for password protection.
      # SSLRequireSSL
      
      AuthType Basic
      AuthName "Authorization Realm"
      AuthUserFile /path/to/passwdfile
      Require valid-user
   </LimitExcept>
</Location>

<Location /svn/repos>
   DAV svn
   SVNParentPath /svn/repos
   AuthType Basic

Written by: Ashley Chew
Email: ashley@csse.uwa.edu.au
Last Revision: 07062007
Allow from all
AuthBasicProvider ldap
AuthName "Subversion Repository"
AuthLDAPUrl
"ldaps://jhett.csse.uwa.edu.au:636/ou=People,dc=csse,dc=uwa,dc=edu,dc=au?uid"
AuthzLDAPAuthoritative Off
require ldap-attribute gidNumber=500
require ldap-group cn=svn,ou=Groups,dc=csse,dc=uwa,dc=edu,dc=au
#AuthzSVNAccessFile /svn/permissions/svnauthz.conf
</Location>

If you don’t have a subversion.conf files, don’t worry you can place it straight in the /etc/httpd/conf/httpd.conf, just copy the sections above that are in bold into it.

The LoadModule Section are important, as you need these modules loaded in apache for Authentication and Management for SVN.

LoadModule dav_svn_module modules/mod_dav_svn.so
LoadModule authz_svn_module modules/mod_authz_svn.so

You can verify this if the modules are loaded in apache by issuing
[root@svn conf]/etc/init.d/httpd restart (Just to restart apache with new configuration if any ie loaded module)
[root@svn conf]# pwd
/etc/httpd/conf
[root@svn conf]# /usr/sbin/apachectl -M
Loaded Modules:
core_module (static)
mpm_prefork_module (static)
http_module (static)
so_module (static)
auth_basic_module (shared)
auth_digest_module (shared)
authn_file_module (shared)
authn_alias_module (shared)
authn_anon_module (shared)
authn_dbm_module (shared)
authn_default_module (shared)
authz_host_module (shared)
authz_user_module (shared)
authz_owner_module (shared)
authz_groupfile_module (shared)
authz_dbm_module (shared)
authz_default_module (shared)
ldap_module (shared)
authnz_ldap_module (shared)
include_module (shared)
log_config_module (shared)
logio_module (shared)
env_module (shared)
ext_filter_module (shared)
mime_magic_module (shared)
extpires_module (shared)
deflate_module (shared)
headers_module (shared)
usertrack_module (shared)
setenvif_module (shared)
mime_module (shared)
dav_module (shared)
status_module (shared)
autoindex_module (shared)
info_module (shared)
dav_fs_module (shared)
vhost_alias_module (shared)
negotiation_module (shared)
dir_module (shared)
actions_module (shared)
speling_module (shared)
userdir_module (shared)
alias_module (shared)
rewrite_module (shared)
proxy_module (shared)
proxy_balancer_module (shared)
proxy_ftp_module (shared)
proxy_http_module (shared)
proxy_connect_module (shared)
cache_module (shared)
suexec_module (shared)
disk_cache_module (shared)
file_cache_module (shared)
mem_cache_module (shared)
cgi_module (shared)
auth_kerb_module (shared)
mysql_auth_module (shared)
auth_pgsql_module (shared)
authz_ldap_module (shared)
perl_module (shared)
php5_module (shared)
proxy_ajp_module (shared)
python_module (shared)
ssl_module (shared)
Now that you have confirmed that the modules are loaded, let's take a look at the configuration file `subversion.conf`. As you see the configuration uses LDAP authentication specifically. Basically it uses `ldap attribute` directive where the Linux group ID of the user has to be 500, or any user belonging to ldap group `cn=svn,ou=Groups,dc=csse,dc=uwa,dc=edu,dc=au` ie these two lines will be allowed access.

```
require ldap-attribute gidNumber=500
require ldap-group cn=svn,ou=Groups,dc=csse,dc=uwa,dc=edu,dc=au
```

Now if I just wanted only certain individual users to be able to use it I would replace these two lines.

```
Require ldap-user test01
Require ldap-user test02
Require ldap-user test03
```

I would advise you to look at the LDAP Directives for authentication which you can find here [http://httpd.apache.org/docs/2.2/mod/mod_authnz_ldap.html](http://httpd.apache.org/docs/2.2/mod/mod_authnz_ldap.html)

But typically the attributes you will be playing with are these Directives.

```
require valid-user
require ldap-user
require ldap-group
require ldap-dn
require ldap-attribute
require ldap-filter
```

Note this section up to now only defines who are valid users to have physical web access, we have yet to create specific repositories and give the repository fine grain access which will be detailed now. We will be using these validated credentials to see what repository they are to access.

Now let create a test repository, the root area for SVN repository is `/svn/repos`, lets create a repository called test. To do that the command would be

```
svnadmin create /svn/repos/test
chown –R apache.apache /svn/repos/test
svnadmin create /svn/repos/ashley
```
You can do a quick test to verify the installation so far, by placing the repository in your web browser. In my case the machine is svn.csse.uwa.edu.au, the SVN Parent Path was set /svn/repos and mapped in apach. And I’ll test it with repository ashley so I would place this in my browser http://svn.csse.uwa.edu.au/svn/repos/ashley. Because of the LDAP directives you should be prompted for username / password, once verified you should see something like this.

Revision 0: /

Powered by Subversion version 1.4.3 (r23084).

(Not much to see really as you hawn’t check anything into the repository plus at the moment, any authenticated user can write/read to any of the repositories as we have not secured any repositories)

Now its time to give the repositories fine grain level access, this is done by using AuthzSVNAccessFile directive, previously it was commented out of the. Lets unhash the line so your /etc/httpd/conf.d/subversion.conf. should be something like this.

LoadModule dav_svn_module modules/mod_dav_svn.so
LoadModule authz_svn_module modules/mod_authz_svn.so

<Location /svn/repos>
DAV svn
SVNParenPath /svn/repos
AuthType Basic
Allow from all
AuthBasicProvider ldap
AuthName "Subversion Repository"
AuthLDAPUrl "ldaps://jhett.csse.uwa.edu.au:636/ou=People,dc=csse,dc=uwa,dc=edu,dc=au?uid"
AuthzLDAPAuthoritative Off
require ldap-attribute gidNumber=500
require ldap-group cn=svn,ou=Groups,dc=csse,dc=uwa,dc=edu,dc=au
AuthzSVNAccessFile /svn/permissions/svnauthz.conf
</Location>

Now we telling for that /svn/repos the repository access is dictated by /svn/permissions/svnauthz.conf. Lets create this file
[root@svn]# touch /svn/permissions/svnauthz.conf
[root@svn]# chown apache.apache /svn/permissions/svnauthz.conf

Now we created the svnauthz.conf file, we need to populate it with who has access and to which repositories once they are already authenticated.
The syntax is pretty straightforward. Let's look at the ashley repository statement. Because we defined **SVN Parent Path** as `/svn/repos`, everything below is basically a repository of sorts.

Keep in mind the General syntax is something like this:

```
[RepositoryDirectory: SubDirectory of Repository]
UnixName: r,w or rw
@UnixGroup: r,w or rw
```

In the case for this line `[ashley:/]` means it's referring to the repository ashley (/svn/repos/RepositoryDirectory) and `/` means the root. So any permissions mentioned below will refer what privileges they have in `/svn/repos/ashley`. In this case the group **svn** (Groups are indicated by `@UnixGroup`) have read permissions only, ashley has read and write privileges and svnuser1 and svnuser2 only has read permissions.

You fine tune it further by mentioning subdirectories of the repository if need be. If there was a subdirectory called `/svn/repos/ashley/Restrict` and I just want ashley to be able to read write to it. I would add this:

```
[ashley:/restrict]
ashley = rw
```

Once you have set the permissions on **svnauthz.conf** that's basically it.

Now let's check something into it via SVN via `httpd`. 

Written by: Ashley Chew
Email: ashley@csse.uwa.edu.au
Last Revision: 07062007

This will import everything from the current directory.

Now see if its imported into by putting http://svn.csse.uwa.edu.au/svn/repos/ashley into your browser

That’s about it now.