web認証(apache)

概要

LDAPを使用してweb認証(BASIC認証)を行う場合のwebサーバ設定例を記述する。
webサーバのOSとして以下の2つを例として記述する。（共にapache2.0を使用）
RedHat Enterprise ES3（以下、RedHatES3）
Fedora core 3（以下、FC3）
なおFC3についてはフィルタリング認証の例のみ記述する。

認証モジュール

- RedHatES3
  RedHatES3の場合、webのldap認証モジュールはmod_authz_ldapを使用する。rpmパッケージがRedHatのサイトにあるので、それをインストールする。なお、mod_authz_ldapの場合、単独ではldapsサーバへアクセスできない。（ldapならOK）よって、webサーバ側でstunnel（sshトンネリング）を使用して、sslのラッピングを行う。なお、stunnelに関してldapサーバ側で行うべきことは特に無い。
- FC3
  FC3の場合、webのldap認証モジュールはapacheに最初から附属しているmod_auth_ldapを使用する。stunnel等はRedHatES3と同様。

グループ認証

webではグループ認証を使用する。グループ認証とは、あるグループにそのユーザが登録されていて、パスワードが一致すれば認証ＯＫとするものである。今回の認証システムの場合、LDAPのou=groupsに、認証の対象となるグループをあらかじめ定義しておく。認証ＯＫとするパターンは以下の通り。

ユーザ名：対象グループのuniquememberに登録されているDN（＝グループに所属のユーザDN）のuid属性値
パスワード：上記DNのuserpassword属性値

なお、パスワードは暗号化されているものとする。LDAPへパスワードを登録する際はldappasswdコマンドを使用する。
また、対象グループのownerにはグループの管理者DNを格納することとする。基本的にownerは、グループ情報の更新に使用し、認証の対象とはしない方向で運用する。

上記のような認証方式で、あるディレクトリ配下にアクセス制限をかける場合の記述例は以下の通り。

```
[root@web test]# cat /home/apache/html/test/.htaccess
AuthName Internal ← 認証の名前。IEなどでは認証ダイアログで表示される。任意でＯＫ。
AuthType Basic ← BASIC認証指定
AuthzLDAPEngine on ← LDAPを使用して認証する指定。
AuthzLDAPServer "localhost:8636" ← LDAPサーバ。
ここではlocalhostの8636へldapアクセスする指定としている。
後述のstunnelにより、ldap://localhost:8636がラップされて最終的に
ldaps://(ldapサーバ):636へアクセスすることになる。ldapサーバ側ではldapsだけ待ち受ければよい。
AuthzLDAPBindDN cn=web1,ou=users,o=kogaku,dc=kyoto-u,dc=ac,dc=jp ← グループの管理者DN。
AuthzLDAPBindPassword secret ← グループ管理者DNのパスワードを平文で記述。
AuthzLDAPUserKey uid ← ユーザ名として認証に使用する属性名を設定。
AuthzLDAPUserBase ou=people,o=kogaku,dc=kyoto-u,dc=ac,dc=jp ← ユーザを検索するベースDNを指定。
AuthzLDAPGroupBase ou=groups,o=kogaku,dc=kyoto-u,dc=ac,dc=jp ← 対象グループのベースDNを指定。
AuthzLDAPGroupKey cn ← グループ名として扱われる属性名を指定。
AuthzLDAPMemberKey uniqueMember ← グループメンバーとして扱われる属性を指定。
require group web1 ← グループ認証の指定。グループ名がweb1であるグループを対象として認証。
```

.htaccess以外にhttpd.confの<Directory>や<Location>などに記述しても同じ。

上記設定の結果、認証処理の流れは以下のようになる。

1.webアクセス

2.ユーザ名・パスワード要求

3.（ユーザがユーザ名・パスワードを入力）

4.webの認証モジュール(mod_authz_ldap)は、localhostの8636ポートへldapで認証を行う。

5.BindDNとBindPasswordで指定したDNとパスワードでldapサーバに対してバインドを行う。

6.GroupBaseとGroupKey及びrequire group web1の指定内容からldapのエントリ「cn=web1,ou=groups,o=kogaku,dc=kyoto-u,dc=ac,dc=jp」を探し出す。

7.MemberKey指定により、uniqueMember属性値（複数可能）に一致するDNを参照する。

8.上記DNが、「uid=（入力ユーザ名）,以下ベースがUserBase」と一致し、そのDNのuserPasswordが入力パスワードと一致すれば認証ＯＫとなる。

フィルタリング認証

グループ認証は、専用のグループを作成する必要がある。これに対し、あらかじめ決まったデータ構造をもとにしてフィルタリングを行い、その結果でアクセス可否を判定する認証もできる。今回、ou=people配下のエントリは所属する組織の属性を持つため、その値をフィルタ条件として認証ができる。ある組織に属する人々のみをアクセス可能にするような制御の記述例は以下の通り。

- RedHatES3(mod_authz_ldap)

```
[root@web test2]# cat /home/apache/html/test2/.htaccess
AuthName Internal
AuthType Basic
AuthzLDAPEngine on
AuthzLDAPServer "localhost:8636"
（bindユーザは指定せず、匿名バインドとしている）
AuthzLDAPUserKey cn
AuthzLDAPUserBase ou=people,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
require filter (eduPersonOrgUnitDN=ou=10000,ou=organization,o=kogaku,\
dc=kyoto-u,dc=ac,dc=jp)
↑検索フィルタを指定。書式はldapのフィルタと同様になる。
この場合は、属性「eduPersonOrgUnitDN」の値が右辺の値と一致する場合のみ
アクセス許可している。
require valid-user
↑ユーザ名とパスワードの一致確認。
```

  上記の内容でアクセス制御を行うと、ユーザ名・パスワードが正しいがフィルタ条件に合致しない場合はアクセス禁止（forbidden）となる。
- FC3(mod_auth_ldap)

```
AuthName Internal
AuthType Basic
AuthLDAPBindDN "cn=test,ou=users,o=kogaku,dc=kyoto-u,dc=ac,dc=jp"
↑bindユーザ
AuthLDAPBindPassword "secret"
AuthLDAPURL "ldap://localhost:8636/ou=people,o=kogaku,dc=kyoto-u,dc=ac,dc=jp?uid?sub?(objectClass=*)"
↑フィルタの記述。uidは認証に使用するユーザ名属性、subはスコープ、括弧内がフィルタとなる。
require valid-user
```

stunnelの設定

mod_authz_ldapおよびmod_auth_ldapは単独でldapsを扱えないため、ldapをldapsにラッピングする。ここでは、ldap://localhost:8636へのldapアクセスをstunnelを使用して、ldaps://localhost:636へラッピングする。なお、ラッピング前と後でホストが異なれば、ポート番号は同じでも良い。stunnelは以下のように設定する。

/etc/stunnel/stunnel.conf

```
# Comment it out on Win32
cert = /etc/stunnel/stunnel.pem
#chroot = /usr/var/run/stunnel/
# PID is created inside chroot jailpid = /var/run/stunnel.pid
#setuid = nobody
#setgid = nobody

# Workaround for Eudora bug
#options = DONT_INSERT_EMPTY_FRAGMENTS

# Authentication stuff
#verify = 2
```

```
# don't forget about c_rehash CApath
# it is located inside chroot jail:
#CApath = /certs
# or simply use CAfile instead:
#CAfile = /usr/etc/stunnel/certs.pem

# Some debugging stuff
debug = 7
#output = stunnel.log

# Use it for client mode
client = yes

#foreground = yes

# Service-level configuration

#[pop3s]
#accept = 995
#connect = 110

#[imaps]
#accept = 993
#connect = 143

#[ssmtp]
#accept = 465
#connect = 25

[ldaps]
accept = 8636
connect = localhost:636

#[s1]
#accept = 5000
#connect = mail.osw.pl:110
# delay = yes

#[s2]
#accept = 5001
#connect = mail.osw.pl:25

#[https]
#accept = 443
#connect = 80
#TIMEOUTclose = 0
```

stunnel.conf内で指定している自ホストの公開鍵ファイルstunnel.pemを作成する。

```
[root@web stunnel]# cd /usr/share/ssl/certs
[root@web certs]# make stunnel.pem
umask 77 ; \
PEM1=`/bin/mktemp /tmp/openssl.XXXXXX` ; \
PEM2=`/bin/mktemp /tmp/openssl.XXXXXX` ; \
/usr/bin/openssl req -newkey rsa:1024 -keyout $PEM1 -nodes -x509 -days 365 -out $PEM2 ; \
cat $PEM1 >  stunnel.pem ; \
echo ""    >> stunnel.pem ; \
cat $PEM2 >> stunnel.pem ; \
rm -f $PEM1 $PEM2
Generating a 1024 bit RSA private key
...................++++++
.....................................++++++
writing new private key to '/tmp/openssl.E24976'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:JP
State or Province Name (full name) [Berkshire]:Kyoto
Locality Name (eg, city) [Newbury]:Kyoto-shi
Organization Name (eg, company) [My Company Ltd]:Kyoto University
Organizational Unit Name (eg, section) []:Faculty of Engineering
Common Name (eg, your name or your server's hostname) []:web←自ホスト名
Email Address []:
```

```
[root@web certs]# ls
Makefile  ca-bundle.crt  dovecot.pem  make-dummy-cert  stunnel.pem
```

このようにしてできたstunnel.pemを/etc/stunnel/stunnel.pemとしてコピー（移動）させておく。

stunnelを有効にするには、とりあえず以下のようにしてデーモンで動作させる。

```
/usr/sbin/stunnel /etc/stunnel/stunnel.conf
```

この状態でmod_authz_ldapがうまくldapsにアクセスできるかどうかを確認する。うまくいかないときはstunnel.confを編集し、foreground = yesにした上で上記のように起動すれば、ログが端末に表示されるのでデバックに役立つ。

本格的にstunnelを使用する場合は、起動時のスクリプトを作成する。
/etc/rc.d/init.d/stuunel

```
#!/bin/bash
#
#       /etc/rc.d/init.d/stunnel
#
# Starts the stunnel
#
# chkconfig: 345 95 5
# description: The stunnel program is designed to work as SSL encryption wrapper
#    between remote clients and local (inetd-startable) or remote servers.
#    The concept is that having non-SSL aware daemons running on your system
#    you can easily set them up to communicate with clients over secure SSL
#    channels.
# processname: stunnel
STUNNEL_OPTIONS="/etc/stunnel/stunnel.conf"
# Source function library.
. /etc/init.d/functions
test -x /usr/sbin/stunnel || exit 0
RETVAL=0
#
# See how we were called.
#prog="stunnel"
start() {
# Check if stunnel is already running
        if [ ! -f /var/lock/subsys/stunnel ]; then
            echo -n $"Starting $prog: "
            daemon /usr/sbin/stunnel $STUNNEL_OPTIONS
            RETVAL=$?
            [ $RETVAL -eq 0 ] && touch /var/lock/subsys/stunnel
            echo
        else
            echo -n $"$prog: already running"
        fi        return $RETVAL
}
stop() {
        echo -n $"Stopping $prog: "
        killproc /usr/sbin/stunnel
        RETVAL=$?
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/stunnel
        echo
        return $RETVAL
}
restart() {
        stop
        start
}
reload() {
        restart
}
status_at() {
        status /usr/sbin/stunnel
}
case "$1" in
start)
        start
        ;;
stop)
        stop
        ;;
reload|restart)
        restart
        ;;
```

```
condrestart)
        if [ -f /var/lock/subsys/stunnel ]; then
             restart
        fi
        ;;
status)
        status_at
        ;;
*)
        echo $"Usage: $0 {start|stop|restart|condrestart|status}"
        exit 1
        esac
exit $?
exit $RETVAL
```

次にstunnelをchkconfigに登録して、起動レベルを以下のように設定する。

```
[root@jimsv5 init.d]# /sbin/chkconfig --add stunnel
[root@jimsv5 init.d]# /sbin/chkconfig --level 4 stunnel off
[root@jimsv5 init.d]# /sbin/chkconfig --list | grep stunnel
stunnel         0:オフ  1:オフ  2:オフ  3:オン  4:オフ  5:オン  6:オフ
```

起動時に立ち上がるように、/usr/sbin/ntsysvを使って設定する。