From EULER Project

PmWiki: Auth User

AuthUser is PmWiki's identity-based authorization system that allows access to pages to be controlled through the use of usernames and passwords. AuthUser can be used in addition to the password-based scheme that is PmWiki's default configuration.

AuthUser is a very flexible system for managing access control on pages, but flexibility can also bring complexity and increased maintenance overhead to the wiki administrator. This is why PmWiki defaults to the simpler password-based system. For some thoughts about the relative merits of the two approaches, see PmWiki:ThoughtsOnAccessControl.

See also: Cookbook:Quick Start for AuthUser.

Activating AuthUser

To activate PmWiki's identity-based system, add the following line to *local/config.php*:

```
include_once("$FarmD/scripts/authuser.php");
```

Ensure that you have set a site wide admin password, otherwise you will not be able to edit SiteAdmin.AuthUser.

Note: Older versions of PmWiki (before 2.2.0-beta58) use *Site.AuthUser*.
PmWiki caches some group and page authorization levels when a page is accessed. For this reason, it is better to include `authuser.php` quite early in config.php, notably

- after any recipe which inserts some custom writable PageStore class (MySQL, SQLite, Compressed PageStore or other)
- and after any internationalization (UTF-8 and XLPage).

(If you don't use a custom PageStore class and i18n, include `authuser.php` first thing in `config.php`.)

All other recipes should be included after these.

Creating user accounts

Most of AuthUser's configuration is performed via the SiteAdmin.AuthUser page. To change the AuthUser configuration, simply edit this page like any other wiki page (you'll typically need to use the site's admin password for this).

To create a login account, simply add lines to SiteAdmin.AuthUser that look like:

```
username: (:encrypt password:)
```

For example, to create a login account for "alice" with a password of "wonderland", enter:

```
alice: (:encrypt wonderland:)
```

When the page is saved, the "`(:encrypt wonderland:)`" part of the text will be replaced by an encrypted form of the password "wonderland". This encryption is done so that someone looking at the SiteAdmin.AuthUser page cannot easily determine the passwords stored in the page.

To change or reset an account's password, simply replace the encrypted string with another `(:encrypt:)` directive.

Controlling access to pages by login

Pages and groups can be protected based on login account by using "passwords" of the form `id:username` in the password fields of `?action=attr` (see PmWiki.Passwords). For example, to restrict a page to being edited by Alice, one would set the password to "`id:alice`".

It's possible to use multiple "id:" declarations and passwords in the `?action=attr` form, thus the following setting would allow access to Alice, Carol, and anyone who knows the password "quick":

```
quick id:alice,carol
```

To allow access to anyone who has successfully logged in, use "`id:*`".

One can also perform site-wide restrictions based on identity in the `$DefaultPasswords` array: e.g.

```
# require valid login before viewing pages
$DefaultPasswords['read'] = 'id:*';
# Alice and carol may edit
$DefaultPasswords['edit'] = 'id:alice,carol';
# All admins and Fred may edit
$DefaultPasswords['edit'] = array('@admins', 'id:Fred');
```

You can change the `$DefaultPasswords` array in local customization files such as:

- local/config.php (for entire wiki)
- farmconfig.php (for entire wikifarm)

Organizing accounts into groups

AuthUser also makes it possible to group login accounts together into authorization groups, indicated by a leading "@" sign. As with login accounts, group memberships are maintained by editing the SiteAdmin.AuthUser page. Group memberships can be specified by either listing the groups for a login account (person belongs to groups) or the login accounts for a group (group includes people). You can repeat or mix-and-match the two kinds as desired:

```
@writers: alice, bob
carol: @writers, @editors
@admins: alice, dave
```

Then, to restrict page access to a particular group, simply use "`@group`" as the "password" in `?action=attr` or the `$DefaultPasswords` array, similar to the way that "`id:username`" is used to restrict access to specific login accounts.

Excluding individuals from password groups

Group password memberships are maintained by editing the SiteAdmin.AuthUser page. To specify a password group that allows access to anyone who is authenticated, you can specify:

```
    @wholeoffice: *
```

If you need to keep "Fred" out of this password group :

```
    @wholeoffice: *,-Fred
```

To allow all users except Fred to change page attributes, for example, you can add to config.php :

```
$DefaultPasswords['attr'] = array('id:*,-Fred');
```

Getting account names and passwords from external sources

The AuthUser script has the capability of obtaining username/password pairs from places other than the SiteAdmin.AuthUser page, such as passwd-formatted files (usually called '.htpasswd' on Apache servers), LDAP servers, or even the *local/config.php* file.

Passwd-formatted files (.htpasswd/.htgroup)

Passwd-formatted files, commonly called *.htpasswd* files in Apache, are text files where each line contains a username and an encrypted password separated by a colon. A typical *.htpasswd* file might look like:

```
    alice:vK99sgDV1an6I
    carol:Q1kSeNcTfwqjs
```

To get AuthUser to obtain usernames and passwords from a *.htaccess* file, add the following line to SiteAdmin.AuthUser, replacing "/path/to/.htpasswd" with the filesystem path of the *.htpasswd* file:

```
    htpasswd: /path/to/.htpasswd
```

Creation and maintenance of the *.htpasswd* file can be performed using a text editor, or any number of other third-party tools available for maintaining *.htpasswd* files. The Apache web server typically includes an *htpasswd* command for creating accounts in .htpasswd:

```
    $ htpasswd /path/to/.htpasswd alice
    New password:
    Re-type new password:
    Adding password for user alice
    $
```

Similarly, one can use *.htgroup* formatted files to specify group memberships. Each line has the name of a group (without the "@"), followed by a colon, followed by a space separated list of usernames in the group.

```
    writers: carol
    editors: alice carol bob
    admins: alice dave
```

Note that the groups are still "@writers", "@editors", and "@admins" in PmWiki even though the file doesn't specify the @ signs. To get AuthUser to load these groups, use a line in SiteAdmin.AuthUser like:

```
htgroup: /path/to/.htgroup
```

Configuration via *local/config.php*

AuthUser configuration settings can also be made from the *local/config.php* file in addition to the SiteAdmin.AuthUser page. Such settings are placed in the $AuthUser array, and *must be set prior to including the* authuser.php *script*. Some examples:

```
# set a password for alice
$AuthUser['alice'] = crypt('wonderland');
# set a password for carol
$AuthUser['carol'] = '$1$CknC8zAs$dC8z2vu3UvnIXMfOcGDON0';
# define the @editors group
$AuthUser['@editors'] = array('alice', 'carol', 'bob');
# Use local/.htpasswd for usernames/passwords
$AuthUser['htpasswd'] = 'local/.htpasswd';
# Use local/.htgroup for group memberships
$AuthUser['htgroup'] = 'local/.htgroup';
```

Configuration via LDAP

Authentication can be performed via an external LDAP server -- simply set an entry for "ldap" in either SiteAdmin.AuthUser or the *local/config.php* file.

```
# use ldap.airius.com for authentication
$AuthUser['ldap'] = 'ldap://ldap.airius.com/ou=People,o=Airius?cn?sub';
```

Make sure to include AuthUser below the entry for the ldap server:

```
# Want to use AuthUser so we can use ldap for passwords
include_once("$FarmD/scripts/authuser.php");
```

And remember to assign the Security Variables for edit and history (or whatever):

```
#Security Variables set login for edit & history page
# to let anyone edit that has an ldap entry:
$HandleAuth['diff'] = 'edit';
$DefaultPasswords['edit'] = 'id:*';
$Author = $AuthId;
```

LDAP authentication in AuthUser closely follows the model used by Apache 2.0's mod_auth_ldap(approve sites) module; see especially the documentation for AuthLDAPUrl(approve sites) for a description of the url format.

For servers that don't allow anonymous binds, AuthUser provides `$AuthLDAPBindDN` and `$AuthLDAPBindPassword` variables to specify the binding to be used for searching.

See also Cookbook:AuthUser via Microsoft LDAP

Setting the Author Name

By default, PmWiki will use a login name in the Author field of the edit form, but allows the author to change this value prior to saving. To force the login name to always be used as the author name, use the following sequence in config.php to activate AuthUser:

```
include_once("$FarmD/scripts/authuser.php");
$Author = $AuthId; # after include_once()
```

To allow more flexibility, but still enable changes to be linked to the authorized user, one can give the author name a prefix of the `$AuthId` instead:

```
include_once("$FarmD/scripts/author.php");
include_once("$FarmD/scripts/authuser.php");
if ($Author) {
    if (strstr($Author, '-') != false) {
        $Author = "$AuthId-" . preg_replace('/^[^-]*-/', '', $Author);
    } else if ($Author != $AuthId) {
        $Author = $AuthId . '-' . $Author;
    } else {
        $Author = $AuthId;
    }
} else {
    $Author = $AuthId;
}
$AuthorLink = "[[~$Author]]";
```

The above will allow the user to put in the author name of their choice, but that will always be replaced by that name prefixed with "`$AuthId`-". The reason why $AuthorLink needs to be set is that, if it isn't, the RecentChanges page will have the wrong link in it.

Removing the "Author" edit field

To force users to edit with their AuthID instead of having a field they can place any name in. This enables administration to keep track of who is doing what better. This line also links the Author name to their Profile. Go to Site.EditForm, remove the line
`$[Author]: (:input e_author:)`
or replace it with
`$[Author]: [[Profiles/{$Author}]]`

Authorization, Sessions, and WikiFarms

PmWiki uses PHP sessions to keep track of any user authorization information. By default PHP is configured so that all interactions with the same server (as identified by the server's domain name) are treated as part of the same session.

What this means for PmWiki is that if there are multiple wikis running within the same domain name, PHP will treat a login to one wiki as being valid for all wikis in the same domain. The easiest fix is to tell each wiki to have use a different "session cookie". Near the top of a wiki's *local/config.php* file, before calling authuser or other recipes, add a line like:

```
session_name('XYZSESSID');
```

The XYZSESSID can be any unique name (letters only is safest).

See Also

- PmWiki.Passwords
- PmWiki.PasswordsAdmin
- Cookbook:AuthUser for tips and tricks
- SiteAdmin.AuthUser

I get http error 500 "Internal Server Error" when I try to log in. What's wrong?

This can happen if the encrypted passwords are not created on the web server that hosts the PmWiki. The crypt function changed during the PHP development, e.g. a password encrypted with PHP 5.2 can not be decrypted in PHP 5.1, but PHP 5.2 can decrypt passwords created by PHP 5.1. This situation normally happens if you prepare everything on your local machine with the latest PHP version and you upload the passwords to a webserver which is running an older version. The same error occurs when you add encrypted passwords to local/config.php.

Solution: Create the passwords on the system with the oldest PHP version and use them on all other systems.

Can I specify authorization group memberships from with *local/config.php*?

Yes -- put the group definition into the $AuthUser array (in config.php):

```
        $AuthUser['@editors'] = array('alice', 'carol', 'bob');
```

I'm running multiple wikis under the same domain name, and logins from one wiki are appearing on other wikis. Shouldn't they be independent?

This is caused by the way that PHP treats sessions. See PmWiki.AuthUser#sessions for more details.

Is there any way to record the time of the last login for each user when using AuthUser? I need a way to look for stale accounts.

See Cookbook:UserLastAction.

Though every settings seem correct, authentication against LDAP is not working, and there is nothing in ldap log. What's wrong ?

Be sure ldap php module is installed (on debian apt-get install php(4|5)-ldap ; apache(2)ctl graceful)

The login form asks for username and password, but only password matters.

Username can be left blank and it still signs in under the account. Is this intentional and if so, can I change it so that the username and password must both be entered? - X 1/18/07 Never mind I think this has something to do with using the admin password. I created a test account and it's working ok.
Make sure you are not entering the admin password when testing the account because, if the password is equal to the admin password, it will authenticate directly through the config.php file and skip any other system.

Do note that even with AuthUser activated you can still log in with a blank username and only entering the password. In that case any password you enter will be "accepted" but only passwords which authenticate in the given context will actually give you any authorization rights. Using this capability AuthUser comfortably coexists with the default password-based system.

If you want to require both username and password, then you need to set an admin id **before** including authuser.php:

```
## Define usernames and passwords.
$AuthUser['carol'] = '$1$CknC8zAs$dC8z2vu3UvnIXMfOcGDON0';

## Enable authentication based on username.
include_once('scripts/authuser.php');

# $DefaultPasswords['admin'] = crypt('secret');
$DefaultPasswords['admin'] = 'id:carol';
```

A username and password will then be required before login is successful.

Is there any way to hide IP addresses once someone has logged in so that registered users can keep their IP addresses invisible to everyone except administrators? - X 1/18/07

Not yet.

Is there a way that people could self-register through AuthUser?

You can see Cookbook:AuthUserSignup for a recipe about this problem.

I would like it that after I have AuthUser turned and a user is authenticated to get on my site, that if I have a password put on a particular page or group that they don't get the AuthUser form to show up (username and password), but only the typical field for password?

See this thread of the mailing list(approve sites).

Retrieved from https://www-sop.inria.fr/mascotte/EULER/wiki/pmwiki.php/PmWiki/AuthUser
Page last modified on July 07, 2009, at 10:09 PM