

AuthzUserPlugin

The AuthzUser Plugin

This [SpamAssassin](#) plugin module allows you to use a standard HTGroup file to control access to certain services via the `services_authorized_for_username` plugin hook.

This is a sample plugin, it may not work at all, so buyer beware.

The groupfile for this feature looks something like:

```
bayessql: someuser foobar1 foobar2
```

Requirements

Requires Apache::Htgroup

Code

Add the following to your local.cf file:

```
loadplugin      Mail::SpamAssassin::Plugin::AuthzUser /path/to/AuthzUser.pm
authzuser_group_file /path/to/htgroupfile
```

AuthzUser.pm:

```
=head1 NAME

package Mail::SpamAssassin::Plugin::AuthzUser;

=head1 SYNOPSIS

    loadplugin Mail::SpamAssassin::Plugin::AuthzUser /path/to/AuthzUser.pm

    authzuser_group_file /path/to/htgroup

=head1 DESCRIPTION

This SpamAssassin plugin module allows you to use a standard HTGroup
file to control access to certain services via the
services_authorized_for_username plugin hook.

=head1 WARNING

This is a sample plugin, it may not work at all, so buyer beware.

The groupfile for this feature looks something like:

bayessql: user foobar1 foobar2

=cut

package Mail::SpamAssassin::Plugin::AuthzUser;

use strict;
use bytes;

use Mail::SpamAssassin::Plugin;

use Apache::Htgroup;

use vars qw(@ISA);
@ISA = qw(Mail::SpamAssassin::Plugin);

sub new {
```

```

my $class = shift;
my $mailsaobject = shift;

# some boilerplate...
$class = ref($class) || $class;
my $self = $class->SUPER::new($mailsaobject);
bless ($self, $class);

return $self;
}

sub parse_config {
    my ($self, $opts) = @_;

    my $key = $opts->{key};

    if ($key eq 'authzuser_group_file') {
        $opts->{conf}->{authzuser_group_file} = $opts->{value};
        $self->inhibit_further_callbacks();
        return 1;
    }

    return 0;
}

sub services_authorized_for_username {
    my ($self, $options) = @_;

    my $username = $options->{username};

    my $services = $options->{services};

    my $conf = $options->{conf};

    my $htgroup = Apache::Htgroup->load($conf->{authzuser_group_file});

    foreach my $servicename (keys %{$services}) {
        if ($htgroup->ismember($username, $servicename)) {
            $services->{$servicename} = 1;
        }
    }

    return;
}

1;

```

How To Use It

Add the above configuration to your local.cf file. Also, you will need to set the `bayes_sql_username_authorized` config option to active the BayesSQL checking.