WhiteListSubjectPlugin

The WhiteListSubject Plugin

This SpamAssassin plugin module provides eval tests for whitelisting and blacklisting particular strings in the Subject header. The value for whitelist_from or blacklist_from is a pseudo regular expression similar to how whitelist_from, etc works, see the SA configuration documentation for more information.

Code

Add the following to your local.cf file:

```
loadplugin Mail::SpamAssassin::Plugin::WhiteListSubject [/path/to/WhiteListSubject.pm]

header SUBJECT_IN_WHITELIST eval:check_subject_in_whitelist()
describe SUBJECT_IN_WHITELIST Subject header is in user's white-list

header SUBJECT_IN_BLACKLIST eval:check_subject_in_blacklist()
describe SUBJECT_IN_BLACKLIST Subject header is in user's black-list

score SUBJECT_IN_WHITELIST -100
score SUBJECT_IN_BLACKLIST 100
```

Then the following in either the local.cf or user_prefs file:

```
whitelist_subject [Bug *]
blacklist_subject Make Money Fast
```

WhiteListSubject.pm:

```
=head1 NAME

Mail::SpamAssassin::Plugin::WhiteListSubject

=head1 SYNOPSIS

 loadplugin Mail::SpamAssassin::Plugin::WhiteListSubject [/path/to/WhiteListSubject.pm]

 header SUBJECT_IN_WHITELIST eval:check_subject_in_whitelist()
 describe SUBJECT_IN_WHITELIST Subject header is in user's white-list

 header SUBJECT_IN_BLACKLIST eval:check_subject_in_blacklist()
 describe SUBJECT_IN_BLACKLIST Subject header is in user's black-list

 score SUBJECT_IN_WHITELIST -100
 score SUBJECT_IN_BLACKLIST 100

 whitelist_subject [Bug *]
 blacklist_subject Make Money Fast

=head1 DESCRIPTION

This SpamAssassin plugin module provides eval tests for whitelisting and blacklisting particular
strings in the Subject header.  The value for whitelist_from or blacklist_from is a pseudo
regular expression similar to how whitelist_from, etc works, see the SA configuration documentation
for more information.

=head1 AUTHOR

Michael Parker <parkerm@pobox.com>

=head1 COPYRIGHT

Copyright (c) 2005 Michael Parker. All rights reserved.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
```

```
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

=cut

package Mail::SpamAssassin::Plugin::WhiteListSubject;

use Mail::SpamAssassin::Plugin;
use strict;
use warnings;
use bytes;

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

# constructor: register the eval rule
sub new {
  my $class = shift;
  my $mailsaobject = shift;

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

  $self->register_eval_rule ("check_subject_in_whitelist");
  $self->register_eval_rule ("check_subject_in_blacklist");

  $self->set_config($mailsaobject->{conf});

  return $self;
}
sub set_config {
  my ($self, $conf) = @_;

  my @cmds = ();

  push(@cmds, {
               setting => 'whitelist_subject',
               code => sub {
                 my ($self, $key, $value, $line) = @_;

                 $value = lc $value;
                 my $re = $value;
                 $re =~ s/[\000\\\(]/_/gs;                  # paranoia
                 $re =~ s/([^\*\?_a-zA-Z0-9])/\\$1/g;       # escape any possible metachars
                 $re =~ tr/?/./;                            # "?" -> "."
                 $re =~ s/\*/\.\*/g;                        # "*" -> "any string"
                 $conf->{$key}->{$value} = ${re};
               }});

  push(@cmds, {
               setting => 'blacklist_subject',
               code => sub {
                 my ($self, $key, $value, $line) = @_;

                 $value = lc $value;
                 my $re = $value;
                 $re =~ s/[\000\\\(]/_/gs;                  # paranoia
                 $re =~ s/([^\*\?_a-zA-Z0-9])/\\$1/g;       # escape any possible metachars
                 $re =~ tr/?/./;                            # "?" -> "."
                 $re =~ s/\*/\.\*/g;                        # "*" -> "any string"
                 $conf->{$key}->{$value} = ${re};
               }});
```

```
  $conf->{parser}->register_commands(\@cmds);
}

sub check_subject_in_whitelist {
  my ($self, $permsgstatus) = @_;

  my $subject = $permsgstatus->get('Subject');

  return 0 unless $subject;

  return $self->_check_subject($permsgstatus->{conf}->{whitelist_subject}, $subject);
}

sub check_subject_in_blacklist {
  my ($self, $permsgstatus) = @_;

  my $subject = $permsgstatus->get('Subject');

  return 0 unless $subject;

  return $self->_check_subject($permsgstatus->{conf}->{blacklist_subject}, $subject);
}

sub _check_subject {
  my ($self, $list, $subject) = @_;

  $subject = lc $subject;

  return 1 if defined($list->{$subject});

  study $subject;
  foreach my $regexp (values %{$list}) {
    if ($subject =~ qr/$regexp/i) {
      return 1;
    }
  }

  return 0;
}

1;
```

How To Use It

Add the above configuration to your local.cf file.