NAME

Authen::SASL − SASL Authentication framework

SYNOPSIS

```
use Authen::SASL;

$sasl = Authen::SASL->new(
  mechanism => 'CRAM-MD5 PLAIN ANONYMOUS',
  callback => {
    pass => \&fetch_password,
    user => $user,
  }
);
```

DESCRIPTION

SASL is a generic mechanism for authentication used by several network protocols. **Authen::SASL** provides an implementation framework that all protocols should be able to share.

The framework allows different implementations of the connection class to be plugged in. At the time of writing there were two such plugins.

Authen::SASL::Perl
: This module implements several mechanisms and is implemented entirely in Perl.

Authen::SASL::XS
: This module uses the Cyrus SASL C−library (both version 1 and 2 are supported).

Authen::SASL::Cyrus
: This module is the predecessor to Authen::SASL::XS. It is reccomended to use Authen::SASL::XS

By default the order in which these plugins are selected is Authen::SASL::XS, Authen::SASL::Cyrus and then Authen::SASL::Perl.

If you want to change it or want to specifically use one implementation only simply do

```
use Authen::SASL qw(Perl);
```

or if you have another plugin module that supports the Authen::SASL API

```
use Authen::SASL qw(My::SASL::Plugin);
```

CONTRUCTOR

new (OPTIONS)
: The constructor may be called with or without arguments. Passing arguments is just a short cut to calling the `mechanism` and `callback` methods.

  callback => { NAME => VALUE, NAME => VALUE, ... }
  : Set the callbacks. See the callback method for details.

  mechanism => NAMES
  mech => NAMES
  : Set the list of mechanisms to choose from. See the mechanism method for details.

  debug => VALUE
  : Set the debug level bit-value to `VALUE`

    Debug output will be sent to `STDERR`. The bits of this value are:

```
1   Show debug messages in the Perl modules for the mechanisms.
    (Currently only used in GSSAPI)
4   With security layers in place show information on packages read.
8   With security layers in place show information on packages written.
```

    The default value is 0.

METHODS

mechanism ( )
: Returns the current list of mechanisms

mechanism (NAMES)
: Set the list of mechanisms to choose from. NAMES should be a space separated string of the names.

callback (NAME)
: Returns the current callback associated with NAME.

callback (NAME => VALUE, NAME => VALUE, ...)
: Sets the given callbacks to the given values

client_new (SERVICE, HOST, SECURITY)
: Creates and returns a new connection object for a client-side connection.

server_new (SERVICE, HOST, OPTIONS)
: Creates and returns a new connection object for a server-side connection.

error ( )
: Returns any error from the last connection

The Connection Class

server_start (CHALLENGE)
: server_start begins the authentication using the chosen mechanism. If the mechanism is not supported by the installed SASL it fails. Because for some mechanisms the client has to start the negotiation, you can give the client challenge as a parameter.

server_step (CHALLENGE)
: server_step performs the next step in the negotiation process. The first parameter you give is the clients challenge/response.

client_start ( )
: The initial step to be performed. Returns the initial value to pass to the server or an empty list on error.

client_step (CHALLENGE)
: This method is called when a response from the server requires it. CHALLENGE is the value from the server. Returns the next value to pass to the server or an empty list on error.

need_step ( )
: Returns true if the selected mechanism requires another step before completion (error or success).

answer (NAME)
: The method will return the value returned from the last call to the callback NAME

property (NAME)
: Returns the property value associated with NAME.

property (NAME => VALUE, NAME => VALUE, ...)
: Sets the named properties to their associated values.

service ( )
: Returns the service argument that was passed to *_new−methods.

host ( )
: Returns the host argument that was passed to *_new−methods.

mechanism ( )
: Returns the name of the chosen mechanism.

is_success ( )
: Once **need_step()** returns false, then you can check if the authentication succeeded by calling this method which returns a boolean value.

Callbacks

There are three different ways in which a callback may be passed

CODEREF

If the value passed is a code reference then, when needed, it will be called and the connection object will be passed as the first argument. In addition some callbacks may be passed additional arguments.

ARRAYREF

If the value passed is an array reference, the first element in the array must be a code reference. When the callback is called the code reference will be called with the connection object passed as the first argument and all other values from the array passed after.

SCALAR

All other values passed will be used directly. ie it is the same as passing an code reference that, when called, returns the value.

SEE ALSO

Authen::SASL::Perl, Authen::SASL::XS, Authen::SASL::Cyrus

AUTHOR

Graham Barr <gbarr@pobox.com>

Please report any bugs, or post any suggestions, to the perl-ldap mailing list <perl−ldap@perl.org>

COPYRIGHT