

Section 35 - The cram_md5 authenticator

35. The cram_md5 authenticator

The CRAM-MD5 authentication mechanism is described in RFC 2195. The server sends a challenge string to the client, and the response consists of a user name and the CRAM-MD5 digest of the challenge string combined with a secret string (password) which is known to both server and client. Thus, the secret is not sent over the network as plain text, which makes this authenticator more secure than plaintext. However, the downside is that the secret has to be available in plain text at either end.

35.1 Using cram_md5 as a server

This authenticator has one server option, which must be set to configure the authenticator as a server:

```
server_secretUse: cram_md5Type: string&dagger;Default: unset
```

When the server receives the client's response, the user name is placed in the expansion variable \$auth1, and server_secret is expanded to obtain the password for that user. The server then computes the CRAM-MD5 digest that the client should have sent, and checks that it received the correct string. If the expansion of server_secret is forced to fail, authentication fails. If the expansion fails for some other reason, a temporary error code is returned to the client.

For compatibility with previous releases of Exim, the user name is also placed in \$1. However, the use of this variables for this purpose is now deprecated, as it can lead to confusion in string expansions that also use numeric variables for other things.

For example, the following authenticator checks that the user name given by the client is "ph10", and if so, uses "secret" as the password. For any other user name, authentication fails. fixed_cram:

```
driver = cram_md5
public_name = CRAM-MD5
server_secret = ${if eq{$auth1}{ph10}{secret}fail}
server_set_id = $auth1
```

If authentication succeeds, the setting of server_set_id preserves the user name in \$authenticated_id. A more typical configuration might look up the secret string in a file, using the user name as the key. For example: lookup_cram:

```
driver = cram_md5
public_name = CRAM-MD5
server_secret = ${lookup{$auth1}|search{/etc/authpwd}{$value}fail}
server_set_id = $auth1
```

Note that this expansion explicitly forces failure if the lookup fails because \$1 contains an unknown user name.

35.2 Using cram_md5 as a client

When used as a client, the cram_md5 authenticator has two options:

```
client_nameUse: cram_md5Type: string&dagger;Default: the primary host name
```

This string is expanded, and the result used as the user name data when computing the response to the server's challenge.

```
client_secretUse: cram_md5Type: string&dagger;Default: unset
```

This option must be set for the authenticator to work as a client. Its value is expanded and the result used as the secret string when computing the response.

Different user names and secrets can be used for different servers by referring to \$host or \$host_address in the options. Forced failure of either expansion string is treated as an indication that this authenticator is not prepared to handle this case. Exim moves on to the next configured client authenticator. Any other expansion failure causes Exim to give up trying to send the message to the current server.

A simple example configuration of a cram_md5 authenticator, using fixed strings, is: fixed_cram:

```
driver = cram_md5
public_name = CRAM-MD5
client_name = ph10
client_secret = secret
```

