NAME

pam_fail_delay – request a delay on failure

SYNOPSIS

#include <security/pam_appl.h>

int pam_fail_delay(pam_handle_t **pamh*, **unsigned int** *usec***);**

DESCRIPTION

It is often possible to attack an authentication scheme by exploiting the time it takes the scheme to deny access to an applicant user. In cases of *short* timeouts, it may prove possible to attempt a *brute force* dictionary attack -- with an automated process, the attacker tries all possible passwords to gain access to the system. In other cases, where individual failures can take measurable amounts of time (indicating the nature of the failure), an attacker can obtain useful information about the authentication process. These latter attacks make use of procedural delays that constitute a *covert channel* of useful information.

To minimize the effectiveness of such attacks, it is desirable to introduce a random delay in a failed authentication process. **Linux-PAM** provides such a facility. The delay occurs upon failure of the **pam_authenticate**(3) and **pam_chauthtok**(3) functions. It occurs *after* all authentication modules have been called, but *before* control is returned to the service application.

The function, **pam_fail_delay**(3), is used to specify a required minimum for the length of the failure-delay; the *usec* argument. This function can be called by the service application and/or the authentication modules, both may have an interest in delaying a reapplication for service by the user. The length of the delay is computed at the time it is required. Its length is pseudo-gausianly distributed about the *maximum* requested value; the resultant delay will differ by as much as 25% of this maximum requested value (both up and down).

On return from **pam_authenticate**(3) or **pam_chauthtok**(3), independent of success or failure, the new requested delay is reset to its default value: zero.

EXAMPLE

For example, a **login** application may require a failure delay of roughly 3 seconds. It will contain the following code:

```
pam_fail_delay(pamh, 3000000 /* micro-seconds */ );
pam_authenticate(pamh, 0);
```

if the modules do not request a delay, the failure delay will be between 2.25 and 3.75 seconds.

However, the modules, invoked in the authentication process, may also request delays:

```
(module #1)   pam_fail_delay(pamh, 2000000);

(module #2)   pam_fail_delay(pamh, 4000000);
```

in this case, it is the largest requested value that is used to compute the actual failed delay: here between 3 and 5 seconds.

RETURN VALUE

Following a successful call to **pam_fail_delay**(3), **PAM_SUCCESS** is returned. All other returns should be considered serious failures.

ERRORS

May be translated to text with **pam_strerror**(3).

CONFORMING TO

Under consideration by the X/Open group for future inclusion in the PAM RFC. 1996/1/10

BUGS

none known.

SEE ALSO

pam_start(3), **pam_get_item**(3) and **pam_strerror**(3).

Also, see the three **Linux-PAM** Guides, for **System administrators**, **module developers**, and **application developers**.