NAME
pam_timestamp − Authenticate using cached successful authentication attempts

SYNOPSIS
pam_timestamp.so [timestampdir=*directory*] [timestamp_timeout=*number*] [verbose] [debug]

DESCRIPTION
In a nutshell, *pam_timestamp* caches successful authentication attempts, and allows you to use a recent successful attempt as the basis for authentication. This is similar mechanism which is used in **sudo**.

When an application opens a session using *pam_timestamp*, a timestamp file is created in the *timestampdir* directory for the user. When an application attempts to authenticate the user, a *pam_timestamp* will treat a sufficiently recent timestamp file as grounds for succeeding.

OPTIONS
timestampdir=*directory*
> Specify an alternate directory where *pam_timestamp* creates timestamp files.

timestamp_timeout=*number*
> How long should *pam_timestamp* treat timestamp as valid after their last modification date (in seconds). Default is 300 seconds.

verbose
> Attempt to inform the user when access is granted.

debug
> Turns on debugging messages sent to **syslog**(3).

MODULE TYPES PROVIDED
The **auth** and **session** module types are provided.

RETURN VALUES
PAM_AUTH_ERR
> The module was not able to retrieve the user name or no valid timestamp file was found.

PAM_SUCCESS
> Everything was successful.

PAM_SESSION_ERR
> Timestamp file could not be created or updated.

NOTES
Users can get confused when they are not always asked for passwords when running a given program. Some users reflexively begin typing information before noticing that it is not being asked for.

EXAMPLES
auth sufficient pam_timestamp.so verbose
auth required   pam_unix.so

session required pam_unix.so
session optional pam_timestamp.so

FILES
/var/run/pam_timestamp/...
> timestamp files and directories

SEE ALSO
pam_timestamp_check(8), **pam.conf**(5), **pam.d**(5), **pam**(7)

AUTHOR
pam_timestamp was written by Nalin Dahyabhai.