

#####  
##### (###)

#####

#####: [43126](#)

##### © 2001, 2002, 2003 #####, ###.

#####  
#####  
#####  
#####  
##### #66001#01###8035 # #####/##### (#####), ###  
#####

#####  
#####.

#####.

#####, ###/1 # #####  
#####  
#####.

###, ###  
###, ###, ###,  
#####.

#####  
#####  
#####  
#####  
#####.

2013#11#07 #####.

#####

#####  
##### (### #  
#####), # #####  
#####  
#####.

## #####

1. #####	2
2. ##### # #####	2
3. ##### ##	6
4. ##### ##	10
5. ##### ## #	13
6. ##### #	16
7. ##### #	17
#. #####	17
#. #####	21
#. ##### #	24
#####	26

## 1. #####

##### (##) #####  
 #####, ##### # #####, #####  
 #####  
 ##, # #####  
 #####.

## # 1995  
 ##, # ## # 1997  
 #####  
 ## #/##### (##), ##  
 ## # (##) (#####)  
 #####. ##  
 #####.

#### # 5., #  
 #### # 4., #  
 ####, ####, ## #TM.

## 2. ##### # #####

## 2.1. #####

##### # ##, #####. ##  
 ##### # ##, ##  
 ##### # ##, #  
 ##### (## #) #####. #####  
 ##### #  
 ##### # 1999 #####.

#####  
(###)

##### (#####) #####, #####  
#####.

##### (#####) ##### ###, #####, #####  
##### ###, ##### #####-  
##### # ##### ##### ##-  
#####.

```
##### (#####) #####, ##### ## ##### ##-
##### ## ##### ## ##### #
##### ## ##### ## #####-
##### #####.
```

```
##### (#####) ##### ## ##### ### ##### #####
##### #####, ##### #####-
### #####, ##### #
#### (#####) #####-
### ##, #####.
```

##### (#####) ##### #####, ##### #####

##### (#####) #####, # ##### ##### #####—

##### (#####) ##### #####, ##### # ##### #####,

##### (#####) ##### ## ##### # #####

```
#####  
(###)
```

---

- ##### alice.
- ##### root.
- ##### **##(1)** #####, ### # #####.
- ##### xi3kiune.
- ##### root, # ##### **##(1)** ##### ##—  
##### root.

### 2.2.2. #####

```
# ##### eve, ##### ###(1)  
##### # login.example.com, # ##### bob, ### #####  
##### !
```

```
% whoami  
eve  
% ssh bob@login.example.com  
bob@login.example.com's password: god  
Last login: Thu Oct 11 09:52:57 2001 from 192.168.0.1  
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994  
The Regents of the University of California. All rights reserved.  
FreeBSD 4.4-STABLE (LOGIN) #4: Tue Nov 27 18:10:34 PST 2001  
  
Welcome to FreeBSD!  
%
```

- ##### eve.
- ##### **###(1)** #####.
- ##### **###(8)** ## ##### login.example.com
- ##### bob.
- ##### god.
- ##### ## #####, ## ##### root.

### 2.2.3. #####

```
#####, ##### ## ##### ## sshd:
```

sshd	auth	required	pam_nologin.so	no_warn
sshd	auth	required	pam_unix.so	no_warn try_first_pass
sshd	account	required	pam_login_access.so	
sshd	account	required	pam_unix.so	

sshd	session	required	pam_lastlog.so	no_fail
sshd	password	required	pam_permit.so	

- ##### sshd (#####  
#####(8)).
- auth, account, session # password #####.
- pam\_nologin.so, pam\_unix.so, pam\_login\_access.so, pam\_lastlog.so # pam\_permit.so #####—  
##. ## #####, ### pam\_unix.so ##### ## #####—  
##### (#####).

### 3. #####

#### 3.1. #####

### ## #####  
##### # #####, ##### ## #####.

auth

#####. ###, #####, #####—  
### # #####. ###  
###:

- ##### [##\\_#####\(3\)](#) #####, #####—  
##### # #####, ##### #  
#####.
- ##### [##\\_#####\(3\)](#) #####, #####,  
###, ##### # ##### #  
#####.

account

#####. ###  
##### #, ## #, #####  
# # #. ##

- ##### [##\\_####\\_####\(3\)](#) #####, ##### #  
#####.

session

#####. ###, ##### #  
##### #, #####, ###  
#####:



##### #####, ### ##### (### #####)#####  
##### ##### ##### ##### #####.

#### ##### # #####<sup>TM</sup> #####, ##### # ##  
## #####, ##### # ##### # ##### # ##### #  
#####.

### 3.3. ##### # #####

#####, ##### # ##### #—  
##### #, ##### # ##### **###\_####(3)**. #####—  
####, ### ##### #, ##### # #####—  
#####. ### ##### #: ##—  
##### (#####  
#####) #####.

##### #, # # # # #. ##—  
#####  
###, ##### (#####) # #—  
##### #, #, #####, # #—  
##### #.

#####—  
### #. #####:

#### binding

#### #, # # # # #  
# # # # #.  
#### #, # # # # #  
#, # # # # #.

#### <sup>TM</sup> 9 (####<sup>TM</sup> 5.9),  
# #.

#### required

####, #####  
#####, ##### #  
#####. #### #, #####—  
## #, # #.

#### requisite

####, #####  
#####, ##### #  
#####. #### #, #—  
#####, #.



#####  
(###)

---

sufficient

#### #####, # ## ##  
# #####, ## #####  
#####, # #####. #####, ##  
#####.

### ## #####, #####  
##### # #####, #####  
##### binding, #####.

optional

#####, ## #####.  
### # optional, ## #####.

#####, ##  
#####, # #####, #####  
##### # #####, ##  
## # #####, ## (#  
### #####, ##### binding ## sufficient, ##  
##### requisite). #####  
## #####, # ##  
#####.

#####, ##  
##### #  
## # #####  
#####  
##### # ##  
#####.

### 3.4. #####

#####. #####, ##  
#####  
#####.

1. #####, #####  
## ## root ##  
##### root.
2. ##### [##](#) [####\(3\)](#) #####  
#####  
#####.
3. #####, ##### (#####,  
#####), # #####  
##### [##\\_##\\_####\(3\)](#).

4. #####(3) #####.
5. #####(3) #####, #####—  
##### # #####. #####, ##  
#####, #####(3) ##### PAM\_NEW\_AUTHTOK\_REQD, #  
PAM\_SUCCESS.
6. ##### PAM\_NEW\_AUTHTOK\_REQD, ##  
#####(3) #####, #####  
#####.
7. #####, #####, #####  
#####(3) #####. #####  
#####, #####, # #####—  
#####.
8. #####, #####  
#####(3) #####.
9. #####, #####, #####,  
#####.
10. #####, #####, ##  
#####(3) #####.
11. #####(3) #####  
# #, # # # # # # # # # # #  
#####.

## 4. #####

### 4.1. #####

#### 4.1.1. ##### /etc/pam.conf

##### /etc/pam.conf. ##  
#####. #####  
####:

login	auth	required	pam_nologin.so	no_warn
-------	------	----------	----------------	---------

#####  
#####  
#####.

#####  
#####

```
#####  
#####,  
##### ##  
#####  
#####  
##### # #####TM ##### pam.conf ##### ##  
#####  
#####  
#####; ####
```

```
##### # #####, #####  
#####. #####  
##### # #####, #####, # #####.  
##### # ##### /etc/pam.d/.
```

```
auth    required    pam_nologin.so no_warn
```

```
# cd -/etc/pam.d
# ln --s su sudo
```

```
#### ## ##### ##### ##### # ##### ##, ## ##### pam.d
##### ##### ##### ## ##### #####
##### #####.
```

```
### ## ##### ##, ##### ## ##### ##### # ##### #####. ## ##,
##### ## ##### # ## ## ##### ##### # ##### #####?
```

#### 4.2. #####

11



#####  
(###)

	PAM_SUCCESS	PAM_IGNORE	other
#####	#	#	### = ###;
#####	#	#	### = ###; ###;
#####	## (!###) ###;	#	#
#####	#	#	#

#### ##### fail ##### ##### # ##### #####, ###  
#####, #####, #####, #####  
#####, #####. # ##### PAM\_SUCCESS.

##### ##, ### ##### PAM\_NEW\_AUTHTOK\_REQD #####—  
#####, #####, #####  
##, # ## ##### PAM\_NEW\_AUTHTOK\_REQD, ##### —  
##### PAM\_NEW\_AUTHTOK\_REQD.

##### ##, ### **###\_#####(3)** #####, ### binding #  
sufficient ##### required.

##### ##, ### **###\_#####(3)** —  
##### (### ##, # ##  
### ##), # ##, ###  
##### binding # sufficient ##### required.

## 5. #####

### 5.1. **###\_#####(8)**

##### **###\_#####(8)** ##### ##, ## —  
##### PAM\_AUTH\_ERR. ##  
##### (#####) #####  
sufficient.

### 5.2. **###\_#####(8)**

##### **###\_#####(8)** ##### #  
### PAM\_TEXT\_INFO. # ##, ##  
#####, #####  
#####.

### 5.3. **###\_#####(8)**

##### **###\_#####(8)** #####  
#####, # #####

#### #####. #### ## #####  
#####  
#####

#### 5.4. ###\_#####(8)

##### [###\\_#####\(8\)](#)

#### 5.5. ###\_#####(8)

##### [###\\_#####\(8\)](#) #####  
##### (##### wheel [##\(1\)](#)). #  
##### [##\(1\)](#),  
#####  
#####

#### 5.6. ###\_#####(8)

##### [###\\_#####\(8\)](#) #####  
#####  
#####  
##### [###\\_#####\(8\)](#) #####  
#####

#### 5.7. ###\_###5(8)

##### [###\\_###5\(8\)](#)

#### 5.8. ###\_###(8)

##### [###\\_###\(8\)](#)

#### 5.9. ###\_#####(8)

##### [###\\_#####\(8\)](#)

#### 5.10. ###\_#####\_#####(8)

##### [###\\_#####\\_#####\(8\)](#) #####  
#####  
##### [#####\(5\)](#).

#### 5.11. ###\_#####(8)

##### [###\\_#####\(8\)](#) #####  
##### [#####\(8\)](#),  
#####

5.12. ###\_####(8)

5.13. ###\_#####(8)

#####  
#####(8) ##### # auth  
requisite ## sufficient ### \_####(8), ##  
#####.

#####\_#####(8)

```
#####(8) #####;#####-
##### PAM_SUCCESS. #####
#####,#####
#####.
```

##### ### #####(8)

##### ### #####(8)

15

### ### #####) ##### 0. ### ##### ### #####, #####  
 ### #(1) ### #####(1), # ##### ##### root #####  
 #####.

## 5.19. ###\_#####(8)

##### ###\_#####(8)

## 5.20. ###\_####(8)

##### ###\_####(8) ##### # ##### #, ##—  
 ### #####. ##### #  
 ##### #, ##### #(1), #  
 ##### #.

## 5.21. ###\_###(8)

##### ###\_###(8) ##### #, ### #. #####  
 ##### #, #####—  
 ### #, # ~/.ssh, #—  
 #####. ##### #####(1) #, #####  
 #####. ##### #, #####  
 ### #, ### # (##### #(1) ###  
 ##### #, ##### #), ### #.

## 5.22. ###\_#####(8)

##### ###\_#####(8)

## 5.23. ###\_####(8)

##### ###\_####(8) ##### ® # #—  
 #####(3) # #  
 #, # #. #—  
 # (##### #  
 #####) # #. #, # #  
 #####—  
 #####.

## 6. ##### #

#### #.



#####  
(###)

7. #####

#### #####.

#. #####

##### (1) # #####. ##—  
#####  
##### (3), ##### security/openpam.h. ###  
#####  
#####. #####—  
#####; ##, ##### # ##### #,  
##### PAM, ##### # ##—  
#####.

```
/*_  
* Copyright (c) 2002,2003 Networks Associates Technology, Inc.  
* All rights reserved.  
*  
* This software was developed for the FreeBSD Project by ThinkSec AS and  
* Network Associates Laboratories, the Security Research Division of  
* Network Associates, Inc. under DARPA/SPAWAR contract N66001-01-C-8035  
* ("CBOSS"), as part of the DARPA CHATS research program.  
*  
* Redistribution and use in source and binary forms, with or without  
* modification, are permitted provided that the following conditions  
* are met:  
* 1. Redistributions of source code must retain the above copyright  
* notice, this list of conditions and the following disclaimer.  
* 2. Redistributions in binary form must reproduce the above copyright  
* notice, this list of conditions and the following disclaimer in the  
* documentation and/or other materials provided with the distribution.  
* 3. The name of the author may not be used to endorse or promote  
* products derived from this software without specific prior written  
* permission.  
*  
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND  
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR -  
PURPOSE  
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE  
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR -  
CONSEQUENTIAL  
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE -  
GOODS  
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, -  
STRICT  
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY  
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
```

```
* SUCH DAMAGE.
*
* $P4: -//depot/projects/openpam/bin/su/su.c#10 $
* $FreeBSD: head/en_US.ISO8859-1/articles/pam/su.c 38826 2012-05-17 19:12:14Z hrs $
*/

#include <sys/param.h>
#include <sys/wait.h>

#include <err.h>
#include <pwd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <syslog.h>
#include <unistd.h>

#include <security/pam_appl.h>
#include <security/openpam.h> /* for openpam_ttyconv() */

extern char **environ;

static pam_handle_t *pamh;
static struct pam_conv pame;

static void
usage(void)
{
    fprintf(stderr, -"Usage: su [login [args]]\n");
    exit(1);
}

int
main(int argc, char *argv[])
{
    char hostname[MAXHOSTNAMELEN];
    const char *user, *tty;
    char **args, **pam_envlist, **pam_env;
    struct passwd *pwd;
    int o, pam_err, status;
    pid_t pid;

    while ((o = getopt(argc, argv, -"h")) != -1)
        switch (o) {
            case -'h':
            default:
                usage();
        }

    argc -= optind;
    argv += optind;
```

```
#####  
(###)
```

---

```
if (argc > 0) {  
    user = *argv;  
    --argc;  
    ++argv;  
} else {  
    user = "-root";  
}  
  
/* initialize PAM */  
pamc.conv = &openpam_ttyconv;  
pam_start("su", user, &pamc, &pamh);  
  
/* set some items */  
gethostname(hostname, sizeof(hostname));  
if ((pam_err = pam_set_item(pamh, PAM_RHOST, hostname)) != PAM_SUCCESS)  
    goto pamerr;  
user = getlogin();  
if ((pam_err = pam_set_item(pamh, PAM_RUSER, user)) != PAM_SUCCESS)  
    goto pamerr;  
tty = ttyname(STDERR_FILENO);  
if ((pam_err = pam_set_item(pamh, PAM_TTY, tty)) != PAM_SUCCESS)  
    goto pamerr;  
  
/* authenticate the applicant */  
if ((pam_err = pam_authenticate(pamh, 0)) != PAM_SUCCESS)  
    goto pamerr;  
if ((pam_err = pam_acct_mgmt(pamh, 0)) == PAM_NEW_AUTHTOK_REQD)  
    pam_err = pam_chauthtok(pamh, PAM_CHANGE_EXPIRED_AUTHTOK);  
if (pam_err != PAM_SUCCESS)  
    goto pamerr;  
  
/* establish the requested credentials */  
if ((pam_err = pam_setcred(pamh, PAM_ESTABLISH_CRED)) != PAM_SUCCESS)  
    goto pamerr;  
  
/* authentication succeeded; open a session */  
if ((pam_err = pam_open_session(pamh, 0)) != PAM_SUCCESS)  
    goto pamerr;  
  
/* get mapped user name; PAM may have changed it */  
pam_err = pam_get_item(pamh, PAM_USER, (const void **)&user);  
if (pam_err != PAM_SUCCESS || (pwd = getpwnam(user)) == NULL)  
    goto pamerr;  
  
/* export PAM environment */  
if ((pam_envlist = pam_getenvlist(pamh)) != NULL) {  
    for (pam_env = pam_envlist; *pam_env != NULL; ++pam_env) {  
        putenv(*pam_env);  
        free(*pam_env);  
    }  
    free(pam_envlist);  
}
```

```
/* build argument list */
if ((args = calloc(argc + 2, sizeof *args)) == NULL) {
    warn("calloc()");
    goto err;
}
*args = pwd->pw_shell;
memcpy(args + 1, argv, argc * sizeof *args);

/* fork and exec */
switch ((pid = fork())) {
case --1:
    warn("fork()");
    goto err;
case 0:
    /* child: give up privs and start a shell */

    /* set uid and groups */
    if (initgroups(pwd->pw_name, pwd->pw_gid) == --1) {
        warn("initgroups()");
        _exit(1);
    }
    if (setgid(pwd->pw_gid) == --1) {
        warn("setgid()");
        _exit(1);
    }
    if (setuid(pwd->pw_uid) == --1) {
        warn("setuid()");
        _exit(1);
    }
    execve(*args, args, environ);
    warn("execve()");
    _exit(1);
default:
    /* parent: wait for child to exit */
    waitpid(pid, &status, 0);

    /* close the session and release PAM resources */
    pam_err = pam_close_session(pamh, 0);
    pam_end(pamh, pam_err);

    exit(WEXITSTATUS(status));
}

pamerr:
fprintf(stderr, -"Sorry\n");
err:
pam_end(pamh, pam_err);
exit(1);
}
```

#. #####

```
/*_
 * Copyright (c) 2002 Networks Associates Technology, Inc.
 * All rights reserved.
 *
 * This software was developed for the FreeBSD Project by ThinkSec AS and
 * Network Associates Laboratories, the Security Research Division of
 * Network Associates, Inc. under DARPA/SPAWAR contract N66001-01-C-8035
 * ("CBOSS"), as part of the DARPA CHATS research program.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. The name of the author may not be used to endorse or promote
 * products derived from this software without specific prior written
 * permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR -
 * PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR -
 * CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE -
 * GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, -
 * STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * $P4: -//depot/projects/openpam/modules/pam_unix/pam_unix.c#3 $
 * $FreeBSD: head/en_US.ISO8859-1/articles/pam/pam_unix.c 38826 2012-05-17 19:12:14Z hrs $
 */

#include <sys/param.h>
```

```
#include <pwd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>

#include <security/pam_modules.h>
#include <security/pam_appl.h>

#ifndef _OPENPAM
static char password_prompt[] = "-Password:";
#endif

#ifndef PAM_EXTERN
#define PAM_EXTERN
#endif

PAM_EXTERN int
pam_sm_authenticate(pam_handle_t *pamh, int flags,
int argc, const char *argv[])
{
#ifndef _OPENPAM
struct pam_conv *conv;
struct pam_message msg;
const struct pam_message *msgp;
struct pam_response *resp;
#endif
struct passwd *pwd;
const char *user;
char *crypt_password, *password;
int pam_err, retry;

/* identify user */
if ((pam_err = pam_get_user(pamh, &user, NULL)) != PAM_SUCCESS)
return (pam_err);
if ((pwd = getpwnam(user)) == NULL)
return (PAM_USER_UNKNOWN);

/* get password */
#ifndef _OPENPAM
pam_err = pam_get_item(pamh, PAM_CONV, (const void **)&conv);
if (pam_err != PAM_SUCCESS)
return (PAM_SYSTEM_ERR);
msg.msg_style = PAM_PROMPT_ECHO_OFF;
msg.msg = password_prompt;
msgp = &msg;
#endif
for (retry = 0; retry < 3; ++retry) {
#ifndef _OPENPAM
pam_err = pam_get_authtok(pamh, PAM_AUTHTOK,
(const char **)&password, NULL);
#else
resp = NULL;

```

#####  
(###)

---

```
pam_err = (*conv->conv)(1, &msgp, &resp, conv->appdata_ptr);
if (resp != NULL) {
    if (pam_err == PAM_SUCCESS)
        password = resp->resp;
    else
        free(resp->resp);
        free(resp);
}
#endif
if (pam_err == PAM_SUCCESS)
    break;
}
if (pam_err == PAM_CONV_ERR)
    return (pam_err);
if (pam_err != PAM_SUCCESS)
    return (PAM_AUTH_ERR);

/* compare passwords */
if ((!pwd->pw_passwd[0] && (flags & PAM_DISALLOW_NULL_AUTHTOK)) ||
    (crypt_password = crypt(password, pwd->pw_passwd)) == NULL ||
    strcmp(crypt_password, pwd->pw_passwd) != 0)
    pam_err = PAM_AUTH_ERR;
else
    pam_err = PAM_SUCCESS;
#endif _OPENPAM
free(password);
#endif
return (pam_err);
}

PAM_EXTERN int
pam_sm_setcred(pam_handle_t *pamh, int flags,
int argc, const char *argv[])
{
    return (PAM_SUCCESS);
}

PAM_EXTERN int
pam_sm_acct_mgmt(pam_handle_t *pamh, int flags,
int argc, const char *argv[])
{
    return (PAM_SUCCESS);
}

PAM_EXTERN int
pam_sm_open_session(pam_handle_t *pamh, int flags,
int argc, const char *argv[])
{
    return (PAM_SUCCESS);
}
```

```
PAM_EXTERN int
pam_sm_close_session(pam_handle_t *pamh, int flags,
    int argc, const char *argv[])
{
    return (PAM_SUCCESS);
}

PAM_EXTERN int
pam_sm_chauthtok(pam_handle_t *pamh, int flags,
    int argc, const char *argv[])
{
    return (PAM_SERVICE_ERR);
}

#ifdef PAM_MODULE_ENTRY
PAM_MODULE_ENTRY("pam_unix");
#endif
```

```
##### , #####, ##### ##### ##-
### #####_#####(3) ## #####. ## #####, # #####
##### ##### ##, ## ##### ##### #####,
##### ## ##### ## #####. ##### ## ## #####
#####, ##### ##### ## # #####_#####(3) ##-
### #####, ## #####, ## ##### ##### ##-
##### # #####.
```

24



#####  
(###)

```
* 3. The name of the author may not be used to endorse or promote
* products derived from this software without specific prior written
* permission.
*
* THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS" AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR -
PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR -
CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE -
GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, -
STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* $FreeBSD: head/en_US.ISO8859-1/articles/pam/converse.c 38826 2012-05-17 19:12:14Z hrs $
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#include <security/pam_appl.h>

int
converse(int n, const struct pam_message **msg,
struct pam_response **resp, void *data)
{
    struct pam_response *aresp;
    char buf[PAM_MAX_RESP_SIZE];
    int i;

    data = data;
    if (n <= 0 || n > PAM_MAX_NUM_MSG)
        return (PAM_CONV_ERR);
    if ((aresp = calloc(n, sizeof *aresp)) == NULL)
        return (PAM_BUF_ERR);
    for (i = 0; i < n; ++i) {
        aresp[i].resp_retcode = 0;
        aresp[i].resp = NULL;
        switch (msg[i]->msg_style) {
            case PAM_PROMPT_ECHO_OFF:
                aresp[i].resp = strdup(getpass(msg[i]->msg));
                if (aresp[i].resp == NULL)
                    goto fail;
                break;
            case PAM_PROMPT_ECHO_ON:
```

#####

```
fputs(msg[i]->msg, stderr);
if (fgets(buf, sizeof buf, stdin) == NULL)
    goto fail;
aresp[i].resp = strdup(buf);
if (aresp[i].resp == NULL)
    goto fail;
break;
case PAM_ERROR_MSG:
fputs(msg[i]->msg, stderr);
if (strlen(msg[i]->msg) > 0 &&
    msg[i]->msg[strlen(msg[i]->msg) - 1] != '\n')
    fputc('\n', stderr);
break;
case PAM_TEXT_INFO:
fputs(msg[i]->msg, stdout);
if (strlen(msg[i]->msg) > 0 &&
    msg[i]->msg[strlen(msg[i]->msg) - 1] != '\n')
    fputc('\n', stdout);
break;
default:
goto fail;
}
}
*resp = aresp;
return (PAM_SUCCESS);
fail:
    for (i = 0; i < n; ++i) {
        if (aresp[i].resp != NULL) {
            memset(aresp[i].resp, 0, strlen(aresp[i].resp));
            free(aresp[i].resp);
        }
    }
    memset(aresp, 0, n * sizeof *aresp);
*resp = NULL;
return (PAM_CONV_ERR);
}
```

#####

#####

[1] [Making Login Services Independent of Authentication Technologies](#). #####  
#####. ### #####.

[2] [X/Open Single Sign-on Preliminary Specification](#). ### #####. 1#85912#144#6.  
#### 1997.

[3] [Pluggable Authentication Modules](#). #####. 6 ##### 1999.

#####  
(###)

---

#####

[4] [PAM Administration](#). ### #####.

#####

[5] [##### OpenPAM](#). #####. ##### ##.

[6] [##### Linux-PAM](#). ##### #. #####.

[7] [##### Solaris PAM](#). ### #####.

