```
**********************************************************
   74435 Mon Mar  4 09:53:29 2019
new/usr/src/lib/libpam/pam_framework.c
10104 pam_set_data() dereferences pointer before checking for NULL
**********************************************************
```

```
   1 /*
   2  * CDDL HEADER START
   3  *
   4  * The contents of this file are subject to the terms of the
   5  * Common Development and Distribution License (the "License").
   6  * You may not use this file except in compliance with the License.
   7  *
   8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9  * or http://www.opensolaris.org/os/licensing.
  10  * See the License for the specific language governing permissions
  11  * and limitations under the License.
  12  *
  13  * When distributing Covered Code, include this CDDL HEADER in each
  14  * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15  * If applicable, add the following below this CDDL HEADER, with the
  16  * fields enclosed by brackets "[]" replaced with your own identifying
  17  * information: Portions Copyright [yyyy] [name of copyright owner]
  18  *
  19  * CDDL HEADER END
  20  */
  21 /*
  22  * Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
  23  * Use is subject to license terms.
  24  */

  26 /*
  27  * Copyright (c) 2019, Joyent, Inc.
  28  */

  30 #include <syslog.h>
  31 #include <dlfcn.h>
  32 #include <sys/types.h>
  33 #include <sys/stat.h>
  34 #include <stdlib.h>
  35 #include <strings.h>
  36 #include <malloc.h>
  37 #include <unistd.h>
  38 #include <fcntl.h>
  39 #include <errno.h>

  41 #include <security/pam_appl.h>
  42 #include <security/pam_modules.h>
  43 #include <sys/mman.h>

  45 #include <libintl.h>

  47 #include "pam_impl.h"

  49 static char *pam_snames [PAM_NUM_MODULE_TYPES] = {
  50         PAM_ACCOUNT_NAME,
  51         PAM_AUTH_NAME,
  52         PAM_PASSWORD_NAME,
  53         PAM_SESSION_NAME
  54 };
______unchanged_portion_omitted_

 792 /*
 793  * Set module specific data
 794  */

 796 int
```



```
 797 pam_set_data(pam_handle_t *pamh, const char *module_data_name, void *data,
 798     void (*cleanup)(pam_handle_t *pamh, void *data, int pam_end_status))
 799 {
 800         struct pam_module_data *psd;

 802         pam_trace(PAM_DEBUG_DATA,
 803             "pam_set_data(%p:%s:%d)=%p", (void *)pamh,
 804             (module_data_name != NULL) ? module_data_name : "NULL",
 805             (pamh != NULL) ? pamh->pam_inmodule : -1, data);
 800             module_data_name ? module_data_name : "NULL", pamh->pam_inmodule,
 801             data);
 806         if (pamh == NULL || (pamh->pam_inmodule != WO_OK) ||
 807             module_data_name == NULL) {
 808                 return (PAM_SYSTEM_ERR);
 809         }

 811         /* check if module data already exists */

 813         for (psd = pamh->ssd; psd; psd = psd->next) {
 814                 if (strcmp(psd->module_data_name, module_data_name) == 0) {
 815                         /* clean up original data before setting the new data */
 816                         if (psd->cleanup) {
 817                                 psd->cleanup(pamh, psd->data, PAM_SUCCESS);
 818                         }
 819                         psd->data = (void *)data;
 820                         psd->cleanup = cleanup;
 821                         return (PAM_SUCCESS);
 822                 }
 823         }

 825         psd = malloc(sizeof (struct pam_module_data));
 826         if (psd == NULL)
 827                 return (PAM_BUF_ERR);

 829         psd->module_data_name = strdup(module_data_name);
 830         if (psd->module_data_name == NULL) {
 831                 free(psd);
 832                 return (PAM_BUF_ERR);
 833         }

 835         psd->data = (void *)data;
 836         psd->cleanup = cleanup;
 837         psd->next = pamh->ssd;
 838         pamh->ssd = psd;
 839         return (PAM_SUCCESS);
 840 }
______unchanged_portion_omitted_
```