

new/usr/src/cmd/cmd-inet/usr.bin/pppd/plugins/passprompt.c

1

```
*****
3005 Mon Jan 21 16:28:07 2019
new/usr/src/cmd/cmd-inet/usr.bin/pppd/plugins/passprompt.c
10125 smatch fixes for cmd-inet
*****
1 /*
2  * passprompt.c - pppd plugin to invoke an external PAP password prompter
3  *
4  * Copyright 1999 Paul Mackerras, Alan Curry.
5  *
6  * This program is free software; you can redistribute it and/or
7  * modify it under the terms of the GNU General Public License
8  * as published by the Free Software Foundation; either version
9  * 2 of the License, or (at your option) any later version.
10 */

12 /*
13  * Copyright (c) 2018, Joyent, Inc.
14  */

16 #include <errno.h>
17 #include <unistd.h>
18 #include <fcntl.h>
19 #include <sys/wait.h>
20 #include <syslog.h>
21 #include "pppd.h"

23 static char promptprog[PATH_MAX+1];

25 static option_t options[] = {
26     { "promptprog", o_string, promptprog,
27       "External PAP password prompting program",
28       OPT_STATIC, NULL, PATH_MAX },
29     { NULL }
30 };

32 static int promptpass(char *user, char *passwd)
33 {
34     int p[2];
35     pid_t kid;
36     int readgood, wstat;
37     int red;

39     if (promptprog[0] == 0 || access(promptprog, X_OK) < 0)
40         return -1; /* sorry, can't help */

42     /* This occurs when we're probed for the ability to supply a password */
43     if (user != NULL && passwd == NULL)
44         return 1;

46     if (pipe(p)) {
47         warn("Can't make a pipe for %s", promptprog);
48         return 0;
49     }
50     if ((kid = fork()) == (pid_t) -1) {
51         warn("Can't fork to run %s", promptprog);
52         (void) close(p[0]);
53         (void) close(p[1]);
54         return 0;
55     }
56     if (kid == (pid_t)0) {
57         /* we are the child, exec the program */
58         char *argv[5], fdstr[32];

60         sys_close();
61         closelog();
```

new/usr/src/cmd/cmd-inet/usr.bin/pppd/plugins/passprompt.c

2

```
62     if (detached && p[1] <= 2) {
63         (void) dup2(p[1], 3);
64         p[1] = 3;
65     }
66     (void) close(p[0]);
67     if (detached) {
68         red = open("/etc/ppp/prompt-errors", O_WRONLY | O_APPEND | O_CREAT,
69                 0600);
70         (void) dup2(red, 1);
71         (void) dup2(red, 2);
72     }
73     (void) seteuid(getuid());
74     (void) setegid(getgid());
75     argv[0] = promptprog;
76     argv[1] = user == NULL ? "" : user;
77     argv[2] = remote_name;
78     (void) slprintf(fdstr, sizeof (fdstr), "%d", p[1]);
79     slprintf(fdstr, sizeof (fdstr), "%d", p[1]);
80     argv[3] = fdstr;
81     argv[4] = NULL;
82     (void) execv(*argv, argv);
83     _exit(127);
84 }

85     /* we are the parent, read the password from the pipe */
86     (void) close(p[1]);
87     readgood = 0;
88     do {
89         red = read(p[0], passwd + readgood, MAXSECRETLEN-1 - readgood);
90         if (red == 0)
91             break;
92         if (red < 0) {
93             if (errno == EINTR)
94                 continue;
95             error("Can't read secret from %s: %m", promptprog);
96             readgood = -1;
97             break;
98         }
99         readgood += red;
100     } while (readgood < MAXSECRETLEN - 1);
101     passwd[readgood] = 0;
102     (void) close(p[0]);

104     /* now wait for child to exit */
105     while (waitpid(kid, &wstat, 0) < 0) {
106         if (errno != EINTR) {
107             warn("error waiting for %s: %m", promptprog);
108             break;
109         }
110     }

112     if (readgood < 0)
113         return 0;
114     if (readgood > 0 && passwd[--readgood] == '\n')
115         passwd[readgood] = '\0';
116     if (!WIFEXITED(wstat))
117         warn("%s terminated abnormally", promptprog);
118     if (WEXITSTATUS(wstat) != 0)
119         warn("%s exited with code %d", promptprog, WEXITSTATUS(wstat));

121     return 1;
122 }
_____unchanged_portion_omitted_____
```

```

*****
56935 Mon Jan 21 16:28:08 2019
new/usr/src/cmd/cmd-inet/usr.sbin/ipadm/ipadm.c
10125 smatch fixes for cmd-inet
*****
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 */

22 /*
23  * Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
24  * Copyright 2017 Nexenta Systems, Inc.
25  * Copyright (c) 2018, Joyent, Inc.
26  * Copyright 2017 Joyent, Inc.
27  * Copyright 2017 Gary Mills
28  * Copyright (c) 2016, Chris Fraire <cfraire@me.com>.
29 */

30 #include <arpa/inet.h>
31 #include <errno.h>
32 #include <getopt.h>
33 #include <inet/ip.h>
34 #include <inet/iptun.h>
35 #include <inet/tunables.h>
36 #include <libldadm.h>
37 #include <libdliptun.h>
38 #include <libdllink.h>
39 #include <libinetutil.h>
40 #include <libipadm.h>
41 #include <locale.h>
42 #include <netdb.h>
43 #include <netinet/in.h>
44 #include <ofmt.h>
45 #include <stdarg.h>
46 #include <stddef.h>
47 #include <stdio.h>
48 #include <stdlib.h>
49 #include <string.h>
50 #include <strings.h>
51 #include <sys/stat.h>
52 #include <sys/types.h>
53 #include <zone.h>

55 #define STR_UNKNOWN_VAL "?"
56 #define LIFC_DEFAULT (LIFC_NOXMIT | LIFC_TEMPORARY | LIFC_ALLZONES |\
57 LIFC_UNDER_IPMP)

59 typedef void cmdfunc_t(int, char **, const char *);
60 static cmdfunc_t do_create_if, do_delete_if, do_enable_if, do_disable_if;

```

```

61 static cmdfunc_t do_show_if;
62 static cmdfunc_t do_set_prop, do_show_prop, do_set_ifprop;
63 static cmdfunc_t do_show_ifprop, do_reset_ifprop, do_reset_prop;
64 static cmdfunc_t do_show_addrprop, do_set_addrprop, do_reset_addrprop;
65 static cmdfunc_t do_create_addr, do_delete_addr, do_show_addr;
66 static cmdfunc_t do_enable_addr, do_disable_addr;
67 static cmdfunc_t do_up_addr, do_down_addr, do_refresh_addr;

69 typedef struct cmd {
70     char          *c_name;
71     cmdfunc_t     *c_fn;
72     const char    *c_usage;
73 } cmd_t;
    unchanged_portion_omitted_

1607 /*
1608  * return true if the address for lifname comes to us from the global zone
1609  * with 'allowed-ips' constraints.
1610  */
1611 static boolean_t
1612 is_from_gz(const char *lifname)
1613 {
1614     ipadm_if_info_t    *if_info;
1615     char                phname[LIFNAMSIZ], *cp;
1616     boolean_t          ret = _B_FALSE;
1617     ipadm_status_t     status;
1618     zoneid_t           zoneid;
1619     ushort_t           zflags;

1621     if ((zoneid = getzoneid()) == GLOBAL_ZONEID)
1622         return (_B_FALSE); /* from-gz only makes sense in a NGZ */

1624     if (zone_getattr(zoneid, ZONE_ATTR_FLAGS, &zflags, sizeof(zflags)) < 0)
1625         return (_B_FALSE);

1627     if (!(zflags & ZF_NET_EXCL))
1628         return (_B_TRUE); /* everything is from the GZ for shared-ip */

1630     (void) strncpy(phname, lifname, sizeof(phname));
1631     if ((cp = strchr(phname, ':')) != NULL)
1632         *cp = '\0';
1633     status = ipadm_if_info(iph, phname, &if_info, 0, LIFC_DEFAULT);
1634     if (status != IPADM_SUCCESS)
1635         return (ret);

1637     if (if_info->ifc_flags & IFIF_L3PROTECT)
1638         ret = _B_TRUE;
1639     if (if_info)
1639         ipadm_free_if_info(if_info);
1640     return (ret);
1641 }
    unchanged_portion_omitted_

```

new/usr/src/cmd/cmd-inet/usr.sbin/ping/ping.c

1

```
*****
70900 Mon Jan 21 16:28:08 2019
new/usr/src/cmd/cmd-inet/usr.sbin/ping/ping.c
10125 smatch fixes for cmd-inet
*****
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 * Copyright 2009 Sun Microsystems, Inc. All rights reserved.
22 * Use is subject to license terms.
23 */
24
25 /*
26 * Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989 AT&T
27 * All Rights Reserved.
28 */
29
30 /*
31 * University Copyright- Copyright (c) 1982, 1986, 1988
32 * The Regents of the University of California.
33 * All Rights Reserved.
34 *
35 * University Acknowledgment- Portions of this document are derived from
36 * software developed by the University of California, Berkeley, and its
37 * contributors.
38 */
39
40 /*
41 * Copyright (c) 2018, Joyent, Inc.
42 * Copyright (c) 2017, Joyent, Inc.
43 */
44 #include <assert.h>
45 #include <stdio.h>
46 #include <strings.h>
47 #include <errno.h>
48 #include <fcntl.h>
49 #include <unistd.h>
50 #include <signal.h>
51 #include <limits.h>
52 #include <math.h>
53 #include <locale.h>
54 #include <thread.h>
55 #include <synch.h>
56
57 #include <sys/time.h>
58 #include <sys/param.h>
59 #include <sys/socket.h>
60 #include <sys/sockio.h>
```

new/usr/src/cmd/cmd-inet/usr.sbin/ping/ping.c

2

```
61 #include <sys/stropts.h>
62 #include <sys/file.h>
63 #include <sys/sysmacros.h>
64 #include <sys/debug.h>
65
66 #include <arpa/inet.h>
67 #include <net/if.h>
68 #include <netinet/in_sysm.h>
69 #include <netinet/in.h>
70 #include <netinet/ip.h>
71 #include <netinet/ip_icmp.h>
72 #include <netinet/ip_var.h>
73 #include <netinet/ip6.h>
74 #include <netinet/icmp6.h>
75 #include <netinet/udp.h>
76 #include <netdb.h>
77 #include <stdlib.h>
78 #include <priv_utils.h>
79
80 #include <libinetutil.h>
81 #include "ping.h"
82
83 /*
84 * This macro is used to compare 16bit, wrapping sequence numbers. Inspired by
85 * TCP's SEQ_LEQ macro.
86 */
87 #define PINGSEQ_LEQ(a, b) ((int16_t)((a)-(b)) <= 0)
88
89 #define MAX_WAIT 10 /* max sec. to wait for response */
90 #define MAX_TRAFFIC_CLASS 255 /* max traffic class for IPv6 */
91 #define MAX_FLOW_LABEL 0xFFFFF /* max flow label for IPv6 */
92 #define MAX_TOS 255 /* max type-of-service for IPv4 */
93
94 #define TIMEOUT 20 /* default timeout value */
95 #define DEFAULT_DATALEN 56
96
97 #define MULTICAST_NOLOOP 1 /* multicast options */
98 #define MULTICAST_TTL 2
99 #define MULTICAST_IF 4
100
101 #define IF_INDEX 0 /* types of -i argument */
102 #define IF_NAME 1
103 #define IF_ADDR 2
104 #define IF_ADDR6 3
105
106 #ifdef BSD
107 #define setbuf(s, b) setlinebuf((s))
108 #endif /* BSD */
109
110
111 /* interface identification */
112 union if_id {
113     int index; /* interface index (e.g., 1, 2) */
114     char *name; /* interface name (e.g., le0, hme0) */
115     union any_in_addr addr; /* interface address (e.g., 10.123.4.5) */
116 };
117
118 unchanged_portion_omitted
119
120
121
122
123 /*
124 * recv_icmp_packet()'s job is to listen to icmp packets and filter out
125 * those ping is interested in.
126 */
127 static void
128 recv_icmp_packet(struct addrinfo *ai_dst, int recv_sock6, int recv_sock,
129                 ushort_t udp_src_port6, ushort_t udp_src_port)
130 {
```

```

1931     struct msghdr in_msg;
1932     struct iovec iov;
1933     struct sockaddr_in6 from6;
1934     fd_set fds;
1935     int result;
1936     int cc;
1937     boolean_t always_true = _B_TRUE; /* lint doesn't like while(_B_TRUE) */

1939     while (always_true) {
1940         (void) FD_ZERO(&fds);
1941         if (recv_sock6 != -1)
1942             FD_SET(recv_sock6, &fds);
1943         if (recv_sock != -1)
1944             FD_SET(recv_sock, &fds);

1946         result = select(MAX(recv_sock6, recv_sock) + 1, &fds,
1947             (fd_set *)NULL, (fd_set *)NULL, (struct timeval *)NULL);
1948         if (result == -1) {
1949             if (errno == EINTR) {
1950                 continue;
1951             } else {
1952                 fprintf(stderr, "%s: select %s\n", progname,
1953                     strerror(errno));
1954                 exit(EXIT_FAILURE);
1955             }
1956         } else if (result > 0) {
1957             in_msg.msg_name = &from6;
1958             in_msg.msg_namelen = sizeof (from6);
1959             iov.iov_base = in_pkt;
1960             iov.iov_len = sizeof (in_pkt);
1961             in_msg.msg_iov = &iov;
1962             in_msg.msg_iovlen = 1;
1963             in_msg.msg_control = ancillary_data;
1964             in_msg.msg_controllen = sizeof (ancillary_data);

1966             /* Do we have an ICMP6 packet waiting? */
1967             if ((recv_sock6 != -1) &&
1968                 (FD_ISSET(recv_sock6, &fds))) {
1969                 cc = recvmsg(recv_sock6, &in_msg, 0);
1970                 if (cc < 0) {
1971                     if (errno != EINTR) {
1972                         fprintf(stderr,
1973                             "%s: recvmsg %s\n",
1974                             progname, strerror(errno));
1975                     }
1976                     continue;
1977                 } else if (cc > 0) {
1978                     check_reply6(ai_dst, &in_msg, cc,
1979                         udp_src_port6);
1980                 }
1981             }
1982             /* Do we have an ICMP packet waiting? */
1983             if ((recv_sock != -1) && (FD_ISSET(recv_sock, &fds))) {
1984                 cc = recvmsg(recv_sock, &in_msg, 0);
1985                 if (cc < 0) {
1986                     if (errno != EINTR) {
1987                         fprintf(stderr,
1988                             "%s: recvmsg %s\n",
1989                             progname, strerror(errno));
1990                     }
1991                     continue;
1992                 } else if (cc > 0) {
1993                     if (cc > 0) {
1994                         check_reply(ai_dst, &in_msg, cc,
1995                             udp_src_port);
1996                     }
1997                 }
1998             }
1999         }
2000     }

```

```

1996     }
1997     }
1998     /*
1999     * If we were probing last IP address of the target host and
2000     * received a reply for each probe sent to this address,
2001     * then we are done!
2002     */
2003     if ((npackets > 0) && (current_targetaddr->next == NULL) &&
2004         (nreceived_last_target == npackets)) {
2005         timer_done = _B_TRUE;
2006         finish();
2007     }
2008     } /* infinite loop */
2009 }

```

_____unchanged_portion_omitted_____

new/usr/src/cmd/cmd-inet/usr.sbin/wificonfig/wificonfig.c

1

```
*****
123002 Mon Jan 21 16:28:08 2019
new/usr/src/cmd/cmd-inet/usr.sbin/wificonfig/wificonfig.c
10125 smatch fixes for cmd-inet
*****
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 2008 Sun Microsystems, Inc. All rights reserved.
23 * Use is subject to license terms.
24 */
26 /*
27  * Copyright (c) 2018, Joyent, Inc.
28  */
29 #pragma ident "%Z%M% %I% %E% SMI"
30 #include <stdio.h>
31 #include <stdlib.h>
32 #include <stddef.h>
33 #include <errno.h>
34 #include <ctype.h>
35 #include <stdarg.h>
36 #include <fcntl.h>
37 #include <unistd.h>
38 #include <net/if.h>
39 #include <sys/types.h>
40 #include <sys/stat.h>
41 #include <sys/wait.h>
42 #include <sys/ipc.h>
43 #include <sys/ddi.h>
44 #include <stropts.h>
45 #include <assert.h>
46 #include <termios.h>
47 #include <time.h>
48 #include <string.h>
49 #include <strings.h>
50 #include <auth_attr.h>
51 #include <auth_list.h>
52 #include <libdevinfo.h>
53 #include <secdb.h>
54 #include <priv.h>
55 #include <pwd.h>
56 #include <umem.h>
57 #include <locale.h>
58 #include <libintl.h>
59 #include <dirent.h>
60 #include <inet/wifi_ioctl.h>
```

new/usr/src/cmd/cmd-inet/usr.sbin/wificonfig/wificonfig.c

2

```
62 /*
63  * Debug information
64  */
65 #ifdef DEBUG
66 int wifi_debug = 0;
67 void wifi_dbgprintf(char *fmt, ...);
68 #define PRTDBG(msg) if (wifi_debug > 1) wifi_dbgprintf msg
69 #else /* DEBUG */
70 #define PRTDBG(msg)
71 #endif /* DEBUG */
73 #define MAX_HISTORY_NUM 10
74 #define MAX_PREFERENCE_NUM 10
75 #define MAX_SCANBUF_LEN 256
76 #define MAX_CONFIG_FILE_LENGTH 256
77 #define MAX_LOADPF_LENGTH 256
78 #define LOADPROFILE_TIMEOUT 10
79 #define RECORD_ADD 0
80 #define RECORD_DEL 1
81 /*
82  * Wificonfig exit status
83  */
84 #define WIFI_EXIT_DEF 0
85 #define WIFI_FATAL_ERR 1
86 #define WIFI_IMPROPER_USE 2
87 #define WIFI_MINOR_ERR 3
89 #define WIFI_LOCKF "/var/run/lockf_wifi"
91 typedef enum {
92     PREFERENCE,
93     HISTORY,
94     ACTIVEP,
95     PROFILE,
96     OTHER
97 } list_type_t;
98 unchanged_portion_omitted
2725 /*
2726  * do_rmprefer: Remove an item in {preference} list
2727  */
2728 /*ARGSUSED*/
2729 static boolean_t
2730 do_rmprefer(int fd, int argc, char **argv)
2731 {
2732     int i = 0;
2733     section_t *p_section = NULL;
2734     aelist_t *plist = NULL;
2735     ae_t *pae = NULL;
2737     PRTDBG(("do_rmprefer(%d, 0%x)\n", argc, argv));
2738     if (argc <= 0) {
2739         do_print_usage();
2740         exit(WIFI_IMPROPER_USE);
2741     }
2743     /*
2744      * if a "all" is inputted, all the items in the preference
2745      * list will be deleted.
2746      */
2747     if (strcmp(argv[0], "all") == 0) {
2748         p_section = find_section(gp_config_file, WIFI_PREFER);
2749         if (p_section != NULL)
2750             plist = p_section->list;
```

```
2752         if ((p_section == NULL) || (plist == NULL))
2753             return (B_FALSE);
2754         pae = plist->ael_head;
2755         while (pae != NULL) {
2756             ae_t *next = pae->ae_next;
2757             free(pae);
2758             pae = next;
2759             pae = pae->ae_next;
2760         }
2761         plist->ael_head = plist->ael_tail = NULL;
2762         plist->ael_argc = 0;
2763     } else if (gp_config_file != NULL) {
2764         for (i = 0; i < argc; i++) {
2765             if (del_prefer(gp_config_file, argv[i], B_TRUE)
2766                 == B_FALSE) {
2767                 return (B_FALSE);
2768             }
2769         }
2770     }
2771     return (B_TRUE);
}
_____unchanged_portion_omitted_____
```