

```

*****
7270 Mon Aug 13 11:38:22 2018
new/usr/src/man/man3secdb/getauthattr.3secdb
9728 3secdb man pages need some tlc
*****
1 \" te
2.\" Copyright (c) 2009, Sun Microsystems, Inc. All Rights Reserved.
3.\" The contents of this file are subject to the terms of the Common Development
4.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
5.\" When distributing Covered Code, include this CDDL HEADER in each file and in
6.TH GETAUTHATTR 3SECDB \"Aug 13, 2018\"
6.TH GETAUTHATTR 3SECDB \"Feb 20, 2009\"
7.SH NAME
8.getauthattr, getauthnam, free_authattr, setauthattr, endauthattr, chkauthattr
9 \- get authorization entry
10.SH SYNOPSIS
11.LP
12.nf
13 cc [ \fIflag\fR... ] \fIfile\fR... -lsecdb -lsocket -lnsl [ \fIlibrary\fR...
14 #include <auth_attr.h>
15 #include <secdb.h>

17 \fBauthattr_t * \fR \fBgetauthattr\fR(\fBvoid \fR);
18 .fi

20 .LP
21 .nf
22 \fBauthattr_t * \fR \fBgetauthnam\fR(\fBconst char * \fR \fBiname \fR);
23 .fi

25 .LP
26 .nf
27 \fBvoid \fR \fBfree_authattr\fR(\fBauthattr_t * \fR \fBiauth \fR);
28 .fi

30 .LP
31 .nf
32 \fBvoid \fR \fBsetauthattr\fR(\fBvoid \fR);
33 .fi

35 .LP
36 .nf
37 \fBvoid \fR \fBendauthattr\fR(\fBvoid \fR);
38 .fi

40 .LP
41 .nf
42 \fBint \fR \fBchkauthattr\fR(\fBconst char * \fR \fBiauthname \fR, \fBconst char * \fR
43 .fi

45 .SH DESCRIPTION
46 .sp
46 .LP
47 The \fBgetauthattr()\fR and \fBgetauthnam()\fR functions each return an
48 \fBauth_attr\fR(4) entry. Entries can come from any of the sources specified in
49 the \fBbnsswitch.conf\fR(4) file.
50 .sp
51 .LP
52 The \fBgetauthattr()\fR function enumerates \fBauth_attr\fR entries. The
53 \fBgetauthnam()\fR function searches for an \fBauth_attr\fR entry with a given
54 authorization name \fBiname\fR. Successive calls to these functions return
55 either successive \fBauth_attr\fR entries or \fBINULL\fR.
56 .sp
57 .LP
58 The internal representation of an \fBauth_attr\fR entry is an \fBauthattr_t\fR
59 structure defined in <\fBauth_attr.h\fR> with the following members:

```

```

60 .sp
61 .in +2
62 .nf
63 char *name; /* name of the authorization */
64 char *res1; /* reserved for future use */
65 char *res2; /* reserved for future use */
66 char *short_desc; /* short description */
67 char *long_desc; /* long description */
68 kva_t *attr; /* array of key-value pair attributes */
69 .fi
70 .in -2

72 .sp
73 .LP
74 The \fBsetauthattr()\fR function "rewinds" to the beginning of the enumeration
75 of \fBauth_attr\fR entries. Calls to \fBgetauthnam()\fR can leave the
76 enumeration in an indeterminate state. Therefore, \fBsetauthattr()\fR should be
77 called before the first call to \fBgetauthattr()\fR.
78 .sp
79 .LP
80 The \fBendauthattr()\fR function may be called to indicate that \fBauth_attr\fR
81 processing is complete; the system may then close any open \fBauth_attr\fR
82 file, deallocate storage, and so forth.
83 .sp
84 .LP
85 The \fBchkauthattr()\fR function verifies whether or not a user has a given
86 authorization. It first reads the \fBBAUTHS_GRANTED\fR key in the
87 \fB/etc/security/policy.conf\fR file and returns 1 if it finds a match for the
88 given authorization. If \fBchkauthattr()\fR does not find a match and the
89 \fBusername\fR is the name of the "console user", defined as the owner of
90 \fB/dev/console\fR, it first reads the \fBBCONSOLE_USER\fR key in
91 \fB/etc/security/policy.conf\fR and returns 1 if the given authorization is in
92 any of the profiles specified in the \fBBCONSOLE_USER\fR keyword, then reads the
93 \fBPROFS_GRANTED\fR key in \fB/etc/security/policy.conf\fR and returns 1 if the
94 given authorization is in any profiles specified with the \fBPROFS_GRANTED\fR
95 keyword. If a match is not found from the default authorizations and default
96 profiles, \fBchkauthattr()\fR reads the \fBuser_attr\fR(4) database. If it does
97 not find a match in \fBuser_attr\fR, it reads the \fBprof_attr\fR(4) database,
98 using the list of profiles assigned to the user, and checks if any of the
99 profiles assigned to the user has the given authorization. The
100 \fBchkauthattr()\fR function returns 0 if it does not find a match in any of
101 the three sources or if the user does not exist.
102 .sp
103 .LP
104 A user is considered to have been assigned an authorization if either of the
105 following are true:
106 .RS +4
107 .TP
108 .ie t \ (bu
109 .el o
110 The authorization name matches exactly any authorization assigned in the
111 \fBuser_attr\fR or \fBprof_attr\fR databases (authorization names are
112 case-sensitive).
113 .RE
114 .RS +4
115 .TP
116 .ie t \ (bu
117 .el o
118 The authorization name suffix is not the key word \fBgrant\fR and the
119 authorization name matches any authorization up to the asterisk (*) character
120 assigned in the \fBuser_attr\fR or \fBprof_attr\fR databases.
121 .RE
122 .sp
123 .LP
124 The examples in the following table illustrate the conditions under which a
125 user is assigned an authorization.

```

```

126 .sp
128 .sp
129 .TS
130 box;
131 c | c | c
132 c | c | c .
133 \fB/etc/security/policy.conf\fR or Is user
134 \fBAuthorization name\fR \fBUser_attr\fR or \fBprof_attr\fR entry
135 \f(CW/etc/security/policy.conf\fR or Is user
136 \fBAuthorization name\fR \fBUser_attr\fR or \fB\fR \fBprof_attr\fR entry
137
136 solaris.printer.postscript solaris.printer.postscript Yes
137 solaris.printer.postscript solaris.printer.* Yes
138 solaris.printer.grant solaris.printer.* No
139 .TE

141 .sp
142 .LP
143 The \fBfree_authattr()\fR function releases memory allocated by the
144 \fBgetauthnam()\fR and \fBgetauthattr()\fR functions.
145 .SH RETURN VALUES
146 .sp
147 The \fBgetauthattr()\fR function returns a pointer to an \fBauthattr_t\fR if
148 it successfully enumerates an entry; otherwise it returns \fBNULL\fR,
149 indicating the end of the enumeration.
150 .sp
151 .LP
152 The \fBgetauthnam()\fR function returns a pointer to an \fBauthattr_t\fR if it
153 successfully locates the requested entry; otherwise it returns \fBNULL\fR.
154 .sp
155 .LP
156 The \fBchkauthattr()\fR function returns 1 if the user is authorized and 0 if
157 the user does not exist or is not authorized.
158 .SH USAGE
159 .sp
160 The \fBgetauthattr()\fR and \fBgetauthnam()\fR functions both allocate memory
161 for the pointers they return. This memory should be deallocated with the
162 \fBfree_authattr()\fR call.
163 .sp
164 .LP
165 Individual attributes in the \fBattr\fR structure can be referred to by calling
166 the \fBkva_match\fR(3SECDB) function.
167 .SH WARNINGS
168 .sp
169 Because the list of legal keys is likely to expand, code must be written to
170 ignore unknown key-value pairs without error.
171 .SH FILES
172 .sp
173 .na
174 \fB\fB/etc/nsswitch.conf\fR
175 .ad
176 .RS 29n
177 configuration file lookup information for the name service switch
178 configuration file lookup information for the name server switch
179 .RE

180 .sp
181 .ne 2
182 .na
183 \fB\fB/etc/user_attr\fR

```

```

184 .ad
185 .RS 29n
186 extended user attributes
187 .RE

189 .sp
190 .ne 2
191 .na
192 \fB\fB/etc/security/auth_attr\fR
193 .ad
194 .RS 29n
195 authorization attributes
196 .RE

198 .sp
199 .ne 2
200 .na
201 \fB\fB/etc/security/policy.conf\fR
202 .ad
203 .RS 29n
204 policy definitions
205 .RE

207 .sp
208 .ne 2
209 .na
210 \fB\fB/etc/security/prof_attr\fR
211 .ad
212 .RS 29n
213 profile information
214 .RE

216 .SH ATTRIBUTES
217 .sp
218 See \fBattributes\fR(5) for descriptions of the following attributes:
219 .sp

221 .sp
222 .TS
223 box;
224 c | c
225 l | l .
226 ATTRIBUTE TYPE ATTRIBUTE VALUE
227
228 MT-Level MT-Safe
229 .TE

231 .SH SEE ALSO
232 .sp
233 \fBgetexecattr\fR(3SECDB), \fBgetprofattr\fR(3SECDB), \fBgetuserattr\fR(3SECDB),
234 \fBkva_match\fR(3SECDB), \fBauth_attr\fR(4), \fBnsswitch.conf\fR(4),
235 \fBpolicy.conf\fR(4), \fBprof_attr\fR(4), \fBuser_attr\fR(4),
236 \fBattributes\fR(5), \fBrbac\fR(5)
237 \fBgetexecattr\fR(3SECDB), \fBgetprofattr\fR(3SECDB),
238 \fBgetuserattr\fR(3SECDB), \fBauth_attr\fR(4), \fBnsswitch.conf\fR(4),
239 \fBprof_attr\fR(4), \fBuser_attr\fR(4), \fBattributes\fR(5), \fBrbac\fR(5)

```

```

*****
8612 Mon Aug 13 11:38:22 2018
new/usr/src/man/man3secdb/getexecattr.3secdb
9728 3secdb man pages need some tlc
*****
1  '\" te
2  .\" Copyright 2018 Peter Tribble
3  .\" Copyright (c) 2005, Sun Microsystems, Inc. All Rights Reserved.
4  .\" The contents of this file are subject to the terms of the Common Development
5  .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
6  .\" When distributing Covered Code, include this CDDL HEADER in each file and in
7  .TH GETEXECATTR 3SECDB \"Aug 13, 2018\"
8  .SH NAME
9  getexecattr, free_execattr, setexecattr, endexecattr, getexecuser, getexecprof,
10 match_execattr \- get execution profile entry
11 .SH SYNOPSIS
12 .LP
13 .nf
14 cc [ \fIflag\fR... ] \fIfile\fR... -lsecdb -lsocket -lnsl [ \fIlibrary\fR...
15 #include <exec_attr.h>
16 #include <secdb.h>

18 \fBexecattr_t * \fR\fBgetexecattr(\fR(\fBvoid\fR);
19 .fi

21 .LP
22 .nf
23 \fBvoid\fR \fBfree_execattr(\fR(\fBexecattr_t * \fR(\fIep\fR);
24 .fi

26 .LP
27 .nf
28 \fBvoid\fR \fBsetexecattr(\fR(\fBvoid\fR);
29 .fi

31 .LP
32 .nf
33 \fBvoid\fR \fBendexecattr(\fR(\fBvoid\fR);
34 .fi

36 .LP
37 .nf
38 \fBexecattr_t * \fR\fBgetexecuser(\fR(\fBconst char * \fR(\fIusername\fR, \fBconst c
39 \fBconst char * \fR(\fIid\fR, \fBint \fR \fIsearch_flag\fR);
40 .fi

42 .LP
43 .nf
44 \fBexecattr_t * \fR\fBgetexecprof(\fR(\fBconst char * \fR(\fIprofname\fR, \fBconst c
45 \fBconst char * \fR(\fIid\fR, \fBint \fR \fIsearch_flag\fR);
46 .fi

48 .LP
49 .nf
50 \fBexecattr_t * \fR\fBmatch_execattr(\fR(\fBexecattr_t * \fR(\fIep\fR, \fBchar * \fR(\
51 \fBchar * \fR(\fIitype\fR, \fBchar * \fR(\fIid\fR);
52 .fi

54 .SH DESCRIPTION
54 .sp
55 .LP
56 The \fBgetexecattr()\fR function returns a single \fBexec_attr\fR(4) entry.
57 Entries can come from any of the sources specified in the
58 \fBbnsswitch.conf\fR(4) file.
59 .sp

```

```

60 .LP
61 Successive calls to \fBgetexecattr()\fR return either successive
62 \fBexec_attr\fR entries or \fBINULL\fR. Because \fBgetexecattr()\fR always
63 returns a single entry, the \fBnext\fR pointer in the \fBexecattr_t\fR data
64 structure points to \fBINULL\fR.
65 .sp
66 .LP
67 The internal representation of an \fBexec_attr\fR entry is an \fBexecattr_t\fR
68 structure defined in <\fBexec_attr.h\fR> with the following members:
69 .sp
70 .in +2
71 .nf
72 char *name; /* name of the profile */
73 char *type; /* type of profile */
74 char *policy; /* policy under which the attributes are */
75 char *type; /* type of profile */
76 char *res1; /* reserved for future use */
77 char *res2; /* reserved for future use */
78 char *id; /* unique identifier */
79 kva_t *attr; /* attributes */
80 struct execattr_s *next; /* optional pointer to next profile */
81 .fi
82 .in -2

84 .sp
85 .LP
86 The \fBfree_execattr()\fR function releases memory. It follows the \fBnext\fR
87 pointers in the \fBexecattr_t\fR structure so that the entire linked list is
88 released.
89 .sp
90 .LP
91 The \fBsetexecattr()\fR function "rewinds" to the beginning of the enumeration
92 of \fBexec_attr\fR entries. Calls to \fBgetexecuser()\fR can leave the
93 enumeration in an indeterminate state. Therefore, \fBsetexecattr()\fR should be
94 called before the first call to \fBgetexecattr()\fR.
95 .sp
96 .LP
97 The \fBendexecattr()\fR function can be called to indicate that \fBexec_attr\fR
98 processing is complete; the library can then close any open \fBexec_attr\fR
99 file, deallocate any internal storage, and so forth.
100 .sp
101 .LP
102 The \fBgetexecuser()\fR function returns a linked list of entries that match
103 the \fIitype\fR and \fIid\fR arguments and have a profile that has been assigned
104 to the user specified by \fIusername\fR, as described in \fBpasswd\fR(4).
105 Profiles for the user are obtained from the list of default profiles in
106 \fB/etc/security/policy.conf\fR (see \fBpolicy.conf\fR(4)) and the
107 \fBuser_attr\fR(4) database. Only entries in the name service scope for which
108 the corresponding profile entry is found in the \fBprof_attr\fR(4) database are
109 returned.
110 .sp
111 .LP
112 The \fBgetexecprof()\fR function returns a linked list of entries that match
113 the \fIitype\fR and \fIid\fR arguments and have the profile specified by the
114 \fIprofname\fR argument. Only entries in the name service scope for which the
115 corresponding profile entry is found in the \fBprof_attr\fR database are
116 returned.
117 .sp
118 .LP
119 Using \fBgetexecuser()\fR and \fBgetexecprof()\fR, programmers can search for
120 any \fIitype\fR argument, such as the manifest constant \fBKV_COMMAND\fR. The
121 arguments are logically AND-ed together so that only entries exactly matching
122 all of the arguments are returned. Wildcard matching applies if there is no
123 exact match for an \fBID\fR. Any argument can be assigned the \fBINULL\fR value
124 to indicate that it is not used as part of the matching criteria. The \fB\fR

```

```

125 search_flag controls whether the function returns the first match
126 (\fBGET_ONE\fR), setting the \fBnext\fR pointer to \fBNULL\fR or all matching
127 entries (\fBGET_ALL\fR), using the \fBnext\fR pointer to create a linked list
128 of all entries that meet the search criteria. See \fBEXAMPLES\fR.
129 .sp
130 .LP
131 Once a list of entries is returned by \fBgetexecuser()\fR or
132 \fBgetexecprof()\fR, the convenience function \fBmatch_execattr()\fR can be
133 used to identify an individual entry. It returns a pointer to the individual
134 element with the same profile name (\fBiprofname\fR), type name (\fBitype\fR),
135 and \fBfid\fR. Function parameters set to \fBNULL\fR are not used as part of the
136 matching criteria. In the event that multiple entries meet the matching
137 criteria, only a pointer to the first entry is returned. The
138 \fBkva_match\fR(3SECDB) function can be used to look up a key in a key-value
139 array.
140 .SH RETURN VALUES
141 .sp
142 .LP
143 Those functions returning data only return data related to the active policy.
144 The \fBgetexecattr()\fR function returns a pointer to a \fBexecattr_t\fR if it
145 successfully enumerates an entry; otherwise it returns \fBNULL\fR, indicating
146 the end of the enumeration.
147 .SH USAGE
148 .sp
149 .LP
150 The \fBgetexecattr()\fR, \fBgetexecuser()\fR, and \fBgetexecprof()\fR functions
151 all allocate memory for the pointers they return. This memory should be
152 deallocated with the \fBfree_execattr()\fR call. The \fBmatch_execattr()\fR
153 function does not allocate any memory. Therefore, pointers returned by this
154 function should not be deallocated.
155 .sp
156 .LP
157 Individual attributes may be referenced in the \fBattr\fR structure by calling
158 the \fBkva_match\fR(3SECDB) function.
159 .SH EXAMPLES
160 .sp
161 .LP
162 \fBExample 1 \fRFind all profiles that have the \fBping\fR command.
163 .sp
164 .in +2
165 .nf
166 if ((execprof=getexecprof(NULL, KV_COMMAND, "/usr/sbin/ping",
167     GET_ONE)) == NULL) {
168     /* do error */
169 }
170 .fi
171 .in -2
172 .sp
173 .LP
174 \fBExample 3 \fRTell everything that can be done in the Filesystem Security
175 profile.
176 .sp
177 .in +2
178 .nf
179 if ((execprof=getexecprof("Filesystem Security", NULL, NULL,
180     if ((execprof=getexecprof("Filesystem Security", KV_NULL, NULL,
181     GET_ALL)) == NULL)) {
182     /* do error */
183 }
184 .fi
185 .in -2
186 .sp
187 .LP
188 \fBExample 4 \fRTell if the \fBtar\fR utility is in a profile assigned to user
189 wetmore. If there is no exact profile entry, the wildcard (*), if defined, is
190 returned.

```

```

200 .sp
201 .LP
202 The following tells if the \fBtar\fR utility is in a profile assigned to user
203 wetmore. If there is no exact profile entry, the wildcard (*), if defined, is
204 returned.
205 .sp
206 .in +2
207 .nf
208 if ((execprof=getexecuser("wetmore", KV_COMMAND, "/usr/bin/tar",
209     GET_ONE)) == NULL) {
210     /* do error */
211 }
212 .fi
213 .in -2
214 .sp
215 .SH FILES
216 .sp
217 .ne 2
218 .na
219 \fB/etc/nsswitch.conf\fR
220 .ad
221 .RS 29n
222 configuration file lookup information for the name service switch
223 configuration file lookup information for the name server switch
224 .RE
225 .sp
226 .ne 2
227 .na
228 \fB/etc/user_attr\fR
229 .ad
230 .RS 29n
231 extended user attributes
232 .RE
233 .sp
234 .ne 2
235 .na
236 \fB/etc/security/exec_attr\fR
237 .ad
238 .RS 29n
239 execution profiles
240 .RE
241 .sp
242 .ne 2
243 .na
244 \fB/etc/security/policy.conf\fR
245 .ad
246 .RS 29n
247 policy definitions
248 .RE
249 .sp
250 .ne 2
251 .na
252 \fB/etc/security/prof_attr\fR
253 .ad
254 .RS 29n
255 profile information
256 .RE
257 .sp
258 .SH ATTRIBUTES
259 .sp
260 .LP

```

257 See \fBattributes\fR(5) for descriptions of the following attributes:  
258 .sp

260 .sp  
261 .TS  
262 box:  
263 c | c  
264 l | l .  
265 ATTRIBUTE TYPE ATTRIBUTE VALUE  
266 \_  
267 MT-Level MT-Safe  
268 .TE

270 .SH SEE ALSO

272 .sp

271 .LP

272 \fBgetauthattr\fR(3SECDB), \fBgetprofattr\fR(3SECDB), \fBgetuserattr\fR(3SECDB),

273 \fBkva\_match\fR(3SECDB), \fBexec\_attr\fR(4), \fBpasswd\fR(4),

274 \fBpolicy.conf\fR(4), \fBprof\_attr\fR(4), \fBuser\_attr\fR(4),

275 \fBattributes\fR(5)

274 \fBgetauthattr\fR(3SECDB), \fBgetuserattr\fR(3SECDB), \fBkva\_match\fR(3SECDB),

275 \fBexec\_attr\fR(4), \fBpasswd\fR(4), \fBpolicy.conf\fR(4), \fBprof\_attr\fR(4),

276 \fBuser\_attr\fR(4), \fBattributes\fR(5)

```

*****
5098 Mon Aug 13 11:38:22 2018
new/usr/src/man/man3secdb/getprofattr.3secdb
9728 3secdb man pages need some tlc
*****
1  \" te
2  \" Copyright (c) 2005, Sun Microsystems, Inc. All Rights Reserved.
3  \" The contents of this file are subject to the terms of the Common Development
4  \" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
5  \" When distributing Covered Code, include this CDDL HEADER in each file and in
6  .TH GETPROFATTR 3SECDB \"Aug 13, 2018\"
6  .TH GETPROFATTR 3SECDB \"Mar 31, 2005\"
7  .SH NAME
8  getprofattr, getprofnam, free_profattr, setprofattr, endprofattr, getproflist,
9  free_proflist \- get profile description and attributes
10 .SH SYNOPSIS
11 .LP
12 .nf
13 cc [ \fIflag\fR... ] \fIfile\fR... -lsecdb -lsocket -lnsl [ \fIlibrary\fR... ]
14 #include <prof_attr.h>

16 \fBprofattr_t * \fR \fBgetprofattr\fR(\fBvoid\fR);
17 .fi

19 .LP
20 .nf
21 \fBprofattr_t * \fR \fBgetprofnam\fR(\fBconst char * \fR \fI\fR \fR \fIname\fR);
22 .fi

24 .LP
25 .nf
26 \fBvoid \fR \fBfree_profattr\fR(\fBprofattr_t * \fR \fIpd\fR);
27 .fi

29 .LP
30 .nf
31 \fBvoid \fR \fBsetprofattr\fR(\fBvoid\fR);
32 .fi

34 .LP
35 .nf
36 \fBvoid \fR \fBendprofattr\fR(\fBvoid\fR);
37 .fi

39 .LP
40 .nf
41 \fBvoid \fR \fBgetproflist\fR(\fBconst char * \fR \fIprofname\fR, \fBchar ** \fR \fIip
42 .fi

44 .LP
45 .nf
46 \fBvoid \fR \fBfree_proflist\fR(\fBchar ** \fR \fIproflist\fR, \fBint \fR \fIprofcnt
47 .fi

49 .SH DESCRIPTION
50 .sp
51 The \fBgetprofattr()\fR and \fBgetprofnam()\fR functions each return a
52 \fBprof_attr\fR entry. Entries can come from any of the sources specified in
53 the \fBnsswitch.conf\fR(4) file.
54 .sp
55 .LP
56 The \fBgetprofattr()\fR function enumerates \fBprof_attr\fR entries. The
57 \fBgetprofnam()\fR function searches for a \fBprof_attr\fR entry with a given
58 \fIname\fR. Successive calls to these functions return either successive
59 \fBprof_attr\fR entries or \fBINULL\fR.

```

```

60 .sp
61 .LP
62 The internal representation of a \fBprof_attr\fR entry is a \fBprofattr_t\fR
63 structure defined in <\fBprof_attr.h\fR> with the following members:
64 .sp
65 .in +2
66 .nf
67 char *name; /* Name of the profile */
68 char *res1; /* Reserved for future use */
69 char *res2; /* Reserved for future use */
70 char *desc; /* Description/Purpose of the profile */
71 kva_t *attr; /* Profile attributes */
72 .fi
73 .in -2

75 .sp
76 .LP
77 The \fBfree_profattr()\fR function releases memory allocated by the
78 \fBgetprofattr()\fR and \fBgetprofnam()\fR functions.
79 .sp
80 .LP
81 The \fBsetprofattr()\fR function "rewinds" to the beginning of the enumeration
82 of \fBprof_attr\fR entries. Calls to \fBgetprofnam()\fR can leave the
83 enumeration in an indeterminate state. Therefore, \fBsetprofattr()\fR should
84 be called before the first call to \fBgetprofattr()\fR.
85 .sp
86 .LP
87 The \fBendprofattr()\fR function may be called to indicate that \fBprof_attr\fR
88 processing is complete; the system may then close any open \fBprof_attr\fR
89 file, deallocate storage, and so forth.
90 .sp
91 .LP
92 The \fBgetproflist()\fR function searches for the list of sub-profiles found in
93 the given \fIprofname\fR and allocates memory to store this list in
94 \fIproflist\fR. The given \fIprofname\fR will be included in the list of
95 sub-profiles. The \fIprofcnt\fR argument indicates the number of items
96 currently valid in \fIproflist\fR. Memory allocated by \fBgetproflist()\fR
97 should be freed using the \fBfree_proflist()\fR function.
98 .sp
99 .LP
100 The \fBfree_proflist()\fR function frees memory allocated by the
101 \fBgetproflist()\fR function. The \fIprofcnt\fR argument specifies the number
102 of items to free from the \fIproflist\fR argument.
103 .SH RETURN VALUES
104 .sp
105 .LP
106 The \fBgetprofattr()\fR function returns a pointer to a \fBprofattr_t\fR if it
107 successfully enumerates an entry; otherwise it returns \fBINULL\fR, indicating
108 the end of the enumeration.
109 .sp
110 .LP
111 The \fBgetprofnam()\fR function returns a pointer to a \fBprofattr_t\fR if it
112 successfully locates the requested entry; otherwise it returns \fBINULL\fR.
113 .SH USAGE
114 .sp
115 .LP
116 .sp
117 .LP
118 Individual attributes in the \fBprofattr_t\fR structure can be referred to by
119 individual attributes in the \fBprof_attr_t\fR structure can be referred to by
120 calling the \fBkva_match\fR(3SECDB) function.
121 .sp
122 .LP
123 The \fBgetprofattr()\fR and \fBgetprofnam()\fR functions both allocate memory

```

```
123 for the pointers they return. This memory should be deallocated with the
124 \fBfree_profattr()\fR function.
125 .SH FILES
129 .sp
126 .ne 2
127 .na
128 \fB\fB/etc/security/prof_attr\fR\fR
129 .ad
130 .RS 27n
131 profiles and their descriptions
132 .RE

134 .SH ATTRIBUTES
139 .sp
135 .LP
136 See \fBattributes\fR(5) for descriptions of the following attributes:
137 .sp

139 .sp
140 .TS
141 box;
142 c | c
143 l | l .
144 ATTRIBUTE TYPE ATTRIBUTE VALUE
145 _
146 MT-Level MT-Safe
147 .TE

149 .SH SEE ALSO
155 .sp
150 .LP
151 \fBbauths\fR(1), \fBprofiles\fR(1), \fBgetexecattr\fR(3SECDB),
152 \fBgetauthattr\fR(3SECDB), \fBkva_match\fR(3SECDB), \fBprof_attr\fR(4),
153 \fBattributes\fR(5)
158 \fBgetauthattr\fR(3SECDB), \fBprof_attr\fR(4)
```

```

*****
4833 Mon Aug 13 11:38:22 2018
new/usr/src/man/man3secdb/getuserattr.3secdb
9728 3secdb man pages need some tlc
*****
1  \" te
2  .\" Copyright (c) 2005, Sun Microsystems, Inc. All Rights Reserved.
3  .\" The contents of this file are subject to the terms of the Common Development
4  .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
5  .\" When distributing Covered Code, include this CDDL HEADER in each file and in
6  .TH GETUSERATTR 3SECDB \"Aug 13, 2018\"
6  .TH GETUSERATTR 3SECDB \"Mar 31, 2005\"
7  .SH NAME
8  getuserattr, getusernam, getuseruid, free_userattr, setuserattr, enduserattr,
9  fgetuserattr \- get user_attr entry
10 .SH SYNOPSIS
11 .LP
12 .nf
13 cc [ \fIflag\fR... ] \fIfile\fR... -lsecdb -lsocket -lnsl [ \fIlibrary\fR... ]
14 #include <user_attr.h>

16 \fBuserattr_t * \fR \fBgetuserattr\fR(\fBvoid\fR);
17 .fi

19 .LP
20 .nf
21 \fBuserattr_t * \fR \fBgetusernam\fR(\fBconst char * \fR \fIname\fR);
22 .fi

24 .LP
25 .nf
26 \fBuserattr_t * \fR \fBgetuseruid\fR(\fBuid_t \fR \fIuid\fR);
27 .fi

29 .LP
30 .nf
31 \fBvoid \fR \fBfree_userattr\fR(\fBuserattr_t * \fR \fIuserattr\fR);
32 .fi

34 .LP
35 .nf
36 \fBvoid \fR \fBsetuserattr\fR(\fBvoid\fR);
37 .fi

39 .LP
40 .nf
41 \fBvoid \fR \fBenduserattr\fR(\fBvoid\fR);
42 .fi

44 .LP
45 .nf
46 \fBuserattr_t * \fR \fBfgetuserattr\fR(\fBFILE * \fR \fIif\fR);
47 .fi

49 .SH DESCRIPTION
50 .sp
50 .LP
51 The \fBgetuserattr()\fR, \fBgetusernam()\fR, and \fBgetuseruid()\fR functions
52 each return a \fBuser_attr\fR(4) entry. Entries can come from any of the
53 sources specified in the \fBnsswitch.conf\fR(4) file. The \fBgetuserattr()\fR
54 function enumerates \fBuser_attr\fR entries. The \fBgetusernam()\fR function
55 searches for a \fBuser_attr\fR entry with a given user name \fIname\fR. The
56 \fBgetuseruid()\fR function searches for a \fBuser_attr\fR entry with a given
57 user ID \fIuid\fR. Successive calls to these functions return either successive
58 \fBuser_attr\fR entries or \fBINULL\fR.
59 .sp

```

```

60 .LP
61 The \fBfgetuserattr()\fR function does not use \fBnsswitch.conf\fR but reads
62 and parses the next line from the stream \fIif\fR. This stream is assumed to
63 have the format of the \fBuser_attr\fR files.
64 .sp
65 .LP
66 The \fBfree_userattr()\fR function releases memory allocated by the
67 \fBgetusernam()\fR, \fBgetuserattr()\fR, and \fBfgetuserattr()\fR functions.
68 .sp
69 .LP
70 The internal representation of a \fBuser_attr\fR entry is a \fBuserattr_t\fR
71 structure defined in <\fBuser_attr.h\fR> with the following members:
72 .sp
73 .in +2
74 .nf
75 char *name; /* name of the user */
76 char *qualifier; /* reserved for future use */
77 char *res1; /* reserved for future use */
78 char *res2; /* reserved for future use */
79 kva_t *attr; /* list of attributes */
80 .fi
81 .in -2

83 .sp
84 .LP
85 The \fBsetuserattr()\fR function "rewinds" to the beginning of the enumeration
86 of \fBuser_attr\fR entries. Calls to \fBgetusernam()\fR may leave the
87 enumeration in an indeterminate state, so \fBsetuserattr()\fR should be called
88 before the first call to \fBgetuserattr()\fR.
89 .sp
90 .LP
91 The \fBenduserattr()\fR function may be called to indicate that \fBuser_attr\fR
92 processing is complete; the library may then close any open \fBuser_attr\fR
93 file, deallocate any internal storage, and so forth.
94 .SH RETURN VALUES
95 .sp
96 .LP
96 The \fBgetuserattr()\fR function returns a pointer to a \fBuserattr_t\fR if it
97 successfully enumerates an entry; otherwise it returns \fBINULL\fR, indicating
98 the end of the enumeration.
99 .sp
100 .LP
101 The \fBgetusernam()\fR function returns a pointer to a \fBuserattr_t\fR if it
102 successfully locates the requested entry; otherwise it returns \fBINULL\fR.
103 .SH USAGE
104 .sp
104 .LP
105 The \fBgetuserattr()\fR and \fBgetusernam()\fR functions both allocate memory
106 for the pointers they return. This memory should be deallocated with the
107 \fBfree_userattr()\fR function.
108 .sp
109 .LP
110 Individual attributes can be referenced in the \fBattr\fR structure by calling
111 the \fBkva_match\fR(3SECDB) function.
112 .SH WARNINGS
113 .sp
113 .LP
114 Because the list of legal keys is likely to expand, code must be written to
115 ignore unknown key-value pairs without error.
116 .SH FILES
117 .sp
117 .ne 2
118 .na
119 \fB\fB/etc/user_attr\fR
120 .ad

```



```
121 .RS 22n
122 extended user attributes
123 .RE

125 .sp
126 .ne 2
127 .na
128 \fB\fB/etc/nsswitch.conf\fR\fR
129 .ad
130 .RS 22n
131 configuration file lookup information for the name service switch
136 configuration file lookup information for the name server switch
132 .RE

134 .SH ATTRIBUTES
140 .sp
135 .LP
136 See \fBattributes\fR(5) for descriptions of the following attributes:
137 .sp

139 .sp
140 .TS
141 box;
142 c | c
143 l | l .
144 ATTRIBUTE TYPE ATTRIBUTE VALUE
145 -
146 MT-Level MT-Safe
147 .TE

149 .SH SEE ALSO
156 .sp
150 .LP
151 \fBgetauthattr\fR(3SECDB), \fBgetexecattr\fR(3SECDB),
152 \fBgetprofattr\fR(3SECDB), \fBkva_match\fR(3SECDB), \fBuser_attr\fR(4),
153 \fBattributes\fR(5)
159 \fBgetprofattr\fR(3SECDB), \fBuser_attr\fR(4), \fBattributes\fR(5)
```

```

*****
2207 Mon Aug 13 11:38:22 2018
new/usr/src/man/man3secdb/kva_match.3secdb
9728 3secdb man pages need some tlc
*****
1 \" te
2.\" Copyright (c) 1999 Sun Microsystems, Inc. All Rights Reserved
3.\" The contents of this file are subject to the terms of the Common Development
4.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
5.\" When distributing Covered Code, include this CDDL HEADER in each file and in
6.TH KVA_MATCH 3SECDB "Aug 13, 2018"
6.TH KVA_MATCH 3SECDB "Aug 12, 1999"
7.SH NAME
8 kva_match \- look up a key in a key-value array
9.SH SYNOPSIS
10.LP
11.nf
12 cc [ \fiflag\fR... ] \fifile\fR... -lsecdb [ \filibrary\fR... ]
12 cc [ \fiflag\fR... ] \fifile\fR... -lsecdb [ \filibrary\fR... ]
13 #include <secdb.h>

15 \fBchar *\fR\fBkva_match\fR(\fBkva_t *\fR\fIkva\fR, \fBchar *\fR\fIkey\fR);
16 .fi

18 .SH DESCRIPTION
19 .sp
19 .LP
20 The \fBkva_match()\fR function searches a \fBkva_t\fR structure, which is part
21 of the \fBauthattr_t\fR, \fBexecattr_t\fR, \fBprofattr_t\fR, or
22 \fBuserattr_t\fR structures. The function takes two arguments: a pointer to a
23 key value array, and a key. If the key is in the array, the function returns a
24 pointer to the first corresponding value that matches that key. Otherwise, the
25 function returns \fINULL\fR.
26 .SH RETURN VALUES
28 .sp
27 .LP
28 Upon successful completion, the function returns a pointer to the value sought.
29 Otherwise, it returns \fINULL\fR.
30 .SH ATTRIBUTES
33 .sp
31 .LP
32 See \fBattributes\fR(5) for descriptions of the following attributes:
33 .sp

35 .sp
36 .TS
37 box;
38 c | c
39 l | l .
40 ATTRIBUTE TYPE ATTRIBUTE VALUE
41 -
42 MT-Level MT-Safe
43 .TE

45 .SH SEE ALSO
49 .sp
46 .LP
47 \fBgetauthattr\fR(3SECDB), \fBgetexecattr\fR(3SECDB),
48 \fBgetprofattr\fR(3SECDB), \fBgetuserattr\fR(3SECDB)
49 .SH NOTES
54 .sp
50 .LP
51 The \fBkva_match()\fR function returns a pointer to data that already exists in
52 the key-value array. It does not allocate its own memory for this pointer but
53 obtains it from the key-value array that is passed as its first argument.

```