
8672 Mon May 26 17:42:22 2014
 new/usr/src/man/man3secdb/getexecattr.3secdb
 4835 getexecattr(3secdb): Minor formatting issues

```

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 GETEXECATTR 3SECDDB "Mar 31, 2005"
7  .SH NAME
8  getexecattr, free_execattr, setexecattr, endexecattr, getexecuser, getexecprof,
9  match_execattr \- get execution profile entry
10 .SH SYNOPSIS
11 .LP
12 .nf
13 cc [ \fIfIflag\fR... ] \fIfIfile\fR... -lsecd -lsocket -lnsl [ \fIfIlibrary\fR...
14 #include <exec_attr.h>
15 #include <secdb.h>

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

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

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

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

35 .LP
36 .nf
37 \fBexecattr_t * \fR \fBgetexecuser\fR(\fBconst char * \fR \fBusername\fR, \fBconst c
38 \fBconst char * \fR \fBuid\fR, \fBint \fR \fBisearch_flag\fR);
39 .fi

41 .LP
42 .nf
43 \fBexecattr_t * \fR \fBgetexecprof\fR(\fBconst char * \fR \fBiprofname\fR, \fBconst c
44 \fBconst char * \fR \fBuid\fR, \fBint \fR \fBisearch_flag\fR);
45 .fi

47 .LP
48 .nf
49 \fBexecattr_t * \fR \fBmatch_execattr\fR(\fBexecattr_t * \fR \fBiep\fR, \fBchar * \fR \
50 \fBchar * \fR \fBitype\fR, \fBchar * \fR \fBuid\fR);
51 .fi

53 .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 \fBNULL\fR. Because \fBgetexecattr()\fR always
63 returns a single entry, the \fBnext\fR pointer in the \fBexecattr_t\fR data
64 structure points to \fBNULL\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                  /* relevant */
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 \fBitype\fR and \fBuid\fR arguments and have a profile that has been assigned
104 to the user specified by \fBusername\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 \fBitype\fR and \fBuid\fR arguments and have the profile specified by the
114 \fBiprofname\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 \fBitype\fR argument, such as the manifest constant \fBKBV_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 \fBNULL\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 (\fIprofname\fR), type name (\fIitype\fR),
134 element with the same profile name ( \fIprofname\fR), type name ( \fIitype\fR),
135 and \fIid\fR. Function parameters set to \fINULL\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(3SECDDB) 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 \fINULL\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
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(3SECDDB) function.
159 .SH EXAMPLES
160 .LP
161 \fBExample 1 \fRFind all profiles that have the \fBping\fR command.
162 .sp
163 .in +2
164 .nf
165 if ((execprof=getexecprof(NULL, KV_COMMAND, "/usr/sbin/ping",
166     GET_ONE)) == NULL) {
167     /* do error */
168 }

```

unchanged_portion_omitted