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

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

.LP
.nf
\fBauthattr_t *\fR\fBgetauthnam\fR(\fBconst char *\fR\fIname\fR);
.fi

.LP
.nf
\fBvoid\fR \fBfree_authattr\fR(\fBauthattr_t *\fR\fIauth\fR);
.fi

.LP
.nf
\fBvoid\fR \fBsetauthattr\fR(\fBvoid\fR);
.fi

.LP
.nf
\fBvoid\fR \fBendauthattr\fR(\fBvoid\fR);
.fi

.LP
.nf
\fBint\fR \fBchkauthattr\fR(\fBconst char *\fR\fIauthname\fR, \fBconst char *\fR
.fi

.SH DESCRIPTION
.sp
.LP
The \fBgetauthattr()\fR and \fBgetauthnam()\fR functions each return an
\fBauth_attr\fR(4) entry. Entries can come from any of the sources specified in
the \fBnsswitch.conf\fR(4) file.
.sp
.LP
The \fBgetauthattr()\fR function enumerates \fBauth_attr\fR entries. The
\fBgetauthnam()\fR function searches for an \fBauth_attr\fR entry with a given
authorization name \fIname\fR. Successive calls to these functions return
either successive \fBauth_attr\fR entries or \fINULL\fR.
.sp
.LP
Th internal representation of an \fBauth_attr\fR entry is an \fBauthattr_t\fR
structure defined in  <\fBauth_attr.h\fR> with the following members:
```

```
.sp
.in +2
.nf
char   *name;        /* name of the authorization */
char   *res1;        /* reserved for future use */
char   *res2;        /* reserved for future use */
char   *short_desc;  /* short description */
char   *long_desc;   /* long description */
kva_t  *attr;        /* array of key-value pair attributes */
.fi
.in -2

.sp
.LP
The \fBsetauthattr()\fR function "rewinds" to the beginning of the enumeration
of \fBauth_attr\fR entries.  Calls to \fBgetauthnam()\fR can leave the
enumeration in an indeterminate state. Therefore, \fBsetauthattr()\fR should be
called before the first call to \fBgetauthattr()\fR.
.sp
.LP
The \fBendauthattr()\fR function may be called to indicate that \fBauth_attr\fR
processing is complete; the system may then close any open \fBauth_attr\fR
file, deallocate storage, and so forth.
.sp
.LP
The \fBchkauthattr()\fR function verifies whether or not a user has a given
authorization. It first reads the \fBAUTHS_GRANTED\fR key in the
\fB/etc/security/policy.conf\fR file and returns 1 if it finds a match for the
given authorization. If \fBchkauthattr()\fR does not find a match and the
\fIusername\fR is the name of the "console user", defined as the owner of
\fB/dev/console\fR, it first reads the \fBCONSOLE_USER\fR key in
\fB/etc/security/policy.conf\fR and returns 1 if the given authorization is in
any of the profiles specified in the \fBCONSOLE_USER\fR keyword, then reads the
\fBPROFS_GRANTED\fR key in \fB/etc/security/policy.conf\fR and returns 1 if the
given authorization is in any profiles specified with the \fBPROFS_GRANTED\fR
keyword. If a match is not found from the default authorizations and default
profiles, \fBchkauthattr()\fR reads the \fBuser_attr\fR(4) database. If it does
not find a match in \fBuser_attr\fR, it reads the \fBprof_attr\fR(4) database,
using the list of profiles assigned to the user, and checks if any of the
profiles assigned to the user has the given authorization.  The
\fBchkauthattr()\fR function returns 0 if it does not find a match in any of
the three sources or if the user does not exist.
.sp
.LP
A user is considered to have been assigned an authorization if either of the
following are true:
.RS +4
.TP
.ie t \(bu
.el o
The authorization name matches exactly any authorization assigned in the
\fBuser_attr\fR or \fBprof_attr\fR databases (authorization names are
case-sensitive).
.RE
.RS +4
.TP
.ie t \(bu
.el o
The authorization name suffix is not the key word \fBgrant\fR and the
authorization name matches any authorization up to the asterisk (*) character
assigned in the \fBuser_attr\fR or \fBprof_attr\fR databases.
.RE
.sp
.LP
The examples in the following table illustrate the conditions under which a
user is assigned an authorization.
```

```
.sp

.sp
.TS
box;
c | c | c
c | c | c .
        \fB/etc/security/policy.conf\fR or      Is user
\fBAuthorization name\fR        \fBuser_attr\fR or \fBprof_attr\fR entry
        \f(CW/etc/security/policy.conf\fR or    Is user
_
\fBAuthorization name\fR        \fBuser_attr\fR or \fB\fR \fBprof_attr\fR entry
_
solaris.printer.postscript      solaris.printer.postscript      Yes
solaris.printer.postscript      solaris.printer.*       Yes
solaris.printer.grant   solaris.printer.*       No
.TE

.sp
.LP
The \fBfree_authattr()\fR function releases memory allocated by the
\fBgetauthnam()\fR and \fBgetauthattr()\fR functions.
.SH RETURN VALUES
.sp
.LP
The \fBgetauthattr()\fR function returns a pointer to an \fBauthattr_t\fR if
it successfully enumerates an entry; otherwise it returns \fINULL\fR,
indicating the end of the enumeration.
.sp
.LP
The \fBgetauthnam()\fR function returns a pointer to an \fBauthattr_t\fR if it
successfully locates the requested entry; otherwise it returns \fINULL\fR.
.sp
.LP
The \fBchkauthattr()\fR function returns 1 if the user is authorized and 0 if
the user does not exist or is not authorized.
.SH USAGE
.sp
.LP
The \fBgetauthattr()\fR and \fBgetauthnam()\fR functions both allocate memory
for the pointers they return. This memory should be deallocated with the
\fBfree_authattr()\fR call.
.sp
.LP
Individual attributes in the \fBattr\fR structure can be referred to by calling
the \fBkva_match\fR(3SECDB) function.
.SH WARNINGS
.sp
.LP
Because the list of legal keys is likely to expand, code must be written to
ignore unknown key-value pairs without error.
.SH FILES
.sp
.ne 2
.na
\fB\fB/etc/nsswitch.conf\fR\fR
.ad
.RS 29n
configuration file lookup information for the name service switch
configuration file lookup information for the name server switch
.RE

.sp
.ne 2
.na
\fB\fB/etc/user_attr\fR\fR
```

```
.ad
.RS 29n
extended user attributes
.RE

.sp
.ne 2
.na
\fB\fB/etc/security/auth_attr\fR\fR
.ad
.RS 29n
authorization attributes
.RE

.sp
.ne 2
.na
\fB\fB/etc/security/policy.conf\fR\fR
.ad
.RS 29n
policy definitions
.RE

.sp
.ne 2
.na
\fB\fB/etc/security/prof_attr\fR\fR
.ad
.RS 29n
profile information
.RE

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
MT-Level        MT-Safe
.TE

.SH SEE ALSO
.sp
.LP
\fBgetexecattr\fR(3SECDB), \fBgetprofattr\fR(3SECDB), \fBgetuserattr\fR(3SECDB),
\fBkva_match\fR(3SECDB), \fBauth_attr\fR(4), \fBnsswitch.conf\fR(4),
\fBpolicy.conf\fR(4), \fBprof_attr\fR(4), \fBuser_attr\fR(4),
\fBattributes\fR(5), \fBrbac\fR(5)
\fBgetexecattr\fR(3SECDB), \fBgetprofattr\fR(3SECDB),
\fBgetuserattr\fR(3SECDB), \fBauth_attr\fR(4), \fBnsswitch.conf\fR(4),
\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
**********************************************************
'\" te
.\" Copyright 2018 Peter Tribble
.\" Copyright (c) 2005, Sun Microsystems, Inc.  All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH GETEXECATTR 3SECDB "Aug 13, 2018"
.TH GETEXECATTR 3SECDB "May 27, 2014"
.SH NAME
getexecattr, free_execattr, setexecattr, endexecattr, getexecuser, getexecprof,
match_execattr \- get execution profile entry
.SH SYNOPSIS
.LP
.nf
cc [ \fIflag\fR... ] \fIfile\fR... -lsecdb  -lsocket  -lnsl  [ \fIlibrary\fR...
#include <exec_attr.h>
#include <secdb.h>

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

.LP
.nf
\fBvoid\fR \fBfree_execattr\fR(\fBexecattr_t *\fR\fIep\fR);
.fi

.LP
.nf
\fBvoid\fR \fBsetexecattr\fR(\fBvoid\fR);
.fi

.LP
.nf
\fBvoid\fR \fBendexecattr\fR(\fBvoid\fR);
.fi

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

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

.LP
.nf
\fBexecattr_t *\fR\fBmatch_execattr\fR(\fBexecattr_t *\fR\fIep\fR, \fBchar *\fR\
     \fBchar *\fR\fItype\fR, \fBchar *\fR\fIid\fR);
.fi

.SH DESCRIPTION
.sp
.LP
The \fBgetexecattr()\fR function returns a single \fBexec_attr\fR(4) entry.
Entries can come from any of the sources specified in the
\fBnsswitch.conf\fR(4) file.
.sp
```

```
.LP
Successive calls to \fBgetexecattr()\fR return either successive
\fBexec_attr\fR entries or \fINULL\fR. Because \fBgetexecattr()\fR always
returns a single entry, the \fBnext\fR pointer in the \fBexecattr_t\fR data
structure points to \fINULL\fR.
.sp
.LP
The internal representation of an \fBexec_attr\fR entry is an \fBexecattr_t\fR
structure defined in <\fBexec_attr.h\fR> with the following members:
.sp
.in +2
.nf
char              *name;   /* name of the profile */
char              *type;   /* type of profile */
char              *policy; /* policy under which the attributes are */
                           /* relevant*/
char              *type;   /* type of profile */
char              *res1;   /* reserved for future use */
char              *res2;   /* reserved for future use */
char              *id;     /* unique identifier */
kva_t             *attr;   /* attributes */
struct execattr_s *next;   /* optional pointer to next profile */
.fi
.in -2

.sp
.LP
The \fBfree_execattr()\fR function releases memory. It follows the \fBnext\fR
pointers in the \fBexecattr_t\fR structure so that the entire linked list is
released.
.sp
.LP
The \fBsetexecattr()\fR function "rewinds" to the beginning of the enumeration
of \fBexec_attr\fR entries. Calls to \fBgetexecuser()\fR can leave the
enumeration in an indeterminate state. Therefore, \fBsetexecattr()\fR should be
called before the first call to \fBgetexecattr()\fR.
.sp
.LP
The \fBendexecattr()\fR function can be called to indicate that \fBexec_attr\fR
processing is complete; the library can then close any open \fBexec_attr\fR
file, deallocate any internal storage, and so forth.
.sp
.LP
The \fBgetexecuser()\fR function returns a linked list of entries that match
the \fItype\fR and \fIid\fR arguments and have a profile that has been assigned
to the user specified by \fIusername\fR, as described in \fBpasswd\fR(4).
Profiles for the user are obtained from the list of default profiles in
\fB/etc/security/policy.conf\fR (see \fBpolicy.conf\fR(4)) and the
\fBuser_attr\fR(4) database. Only entries in the name service scope for which
the corresponding profile entry is found in the \fBprof_attr\fR(4) database are
returned.
.sp
.LP
The \fBgetexecprof()\fR function returns a linked list of entries that match
the \fItype\fR and \fIid\fR arguments and have the profile specified by the
\fIprofname\fR argument. Only entries in the name service scope for which the
corresponding profile entry is found in the \fBprof_attr\fR database are
returned.
.sp
.LP
Using \fBgetexecuser()\fR and \fBgetexecprof()\fR, programmers can search for
any \fItype\fR argument, such as the manifest constant \fBKV_COMMAND\fR. The
arguments are logically AND-ed together so that only entries exactly matching
all of the arguments are returned. Wildcard matching applies if there is no
exact match for an \fBID\fR. Any argument can be assigned the \fINULL\fR value
to indicate that it is not used as part of the matching criteria. The \fB\fR
```

```
search_flag controls whether the function returns the first match
(\fBGET_ONE\fR), setting the \fBnext\fR pointer to \fINULL\fR or all matching
entries (\fBGET_ALL\fR), using the \fBnext\fR pointer to create a linked list
of all entries that meet the search criteria. See \fBEXAMPLES\fR.
.sp
.LP
Once a list of entries is returned by \fBgetexecuser()\fR or
\fBgetexecprof()\fR, the convenience function \fBmatch_execattr()\fR can be
used to identify an individual entry. It returns a pointer to the individual
element with the same profile name (\fIprofname\fR), type name (\fItype\fR),
and \fIid\fR. Function parameters set to \fINULL\fR are not used as part of the
matching criteria. In the event that multiple entries meet the matching
criteria, only a pointer to the first entry is returned. The
\fBkva_match\fR(3SECDB) function can be used to look up a key in a key-value
array.
.SH RETURN VALUES
.sp
.LP
Those functions returning data only return data related to the active policy.
The \fBgetexecattr()\fR function returns a pointer to a \fBexecattr_t\fR if it
successfully enumerates an entry; otherwise it returns \fINULL\fR, indicating
the end of the enumeration.
.SH USAGE
.sp
.LP
The \fBgetexecattr()\fR, \fBgetexecuser()\fR, and \fBgetexecprof()\fR functions
all allocate memory for the pointers they return. This memory should be
deallocated with the \fBfree_execattr()\fR call. The \fBmatch_execattr()\fR
function does not allocate any memory. Therefore, pointers returned by this
function should not be deallocated.
.sp
.LP
Individual attributes may be referenced in the \fBattr\fR structure by calling
the \fBkva_match\fR(3SECDB) function.
.SH EXAMPLES
.LP
\fBExample 1 \fRFind all profiles that have the \fBping\fR command.
.sp
.in +2
.nf
if ((execprof=getexecprof(NULL, KV_COMMAND, "/usr/sbin/ping",
    GET_ONE)) == NULL) {
        /* do error */
}
______unchanged_portion_omitted_
.fi
.in -2

.LP
\fBExample 3 \fRTell everything that can be done in the Filesystem Security
profile.
.sp
.in +2
.nf
if ((execprof=getexecprof("Filesystem Security", NULL, NULL,
if ((execprof=getexecprof("Filesystem Security", KV_NULL, NULL,
    GET_ALL))==NULL)) {
        /* do error */
}
.fi
.in -2

.LP
\fBExample 4 \fRTell if the \fBtar\fR utility is in a profile assigned to user
wetmore. If there is no exact profile entry, the wildcard (*), if defined, is
returned.
```

```
.sp
.LP
The following tells if the \fBtar\fR utility is in a profile assigned to user
wetmore. If there is no exact profile entry, the wildcard (*), if defined, is
returned.

.sp
.in +2
.nf
if ((execprof=getexecuser("wetmore", KV_COMMAND, "/usr/bin/tar",
    GET_ONE))==NULL) {
        /* do error */
}
.fi
.in -2

.SH FILES
.sp
.ne 2
.na
\fB\fB/etc/nsswitch.conf\fR\fR
.ad
.RS 29n
configuration file lookup information for the name service switch
configuration file lookup information for the name server switch
.RE

.sp
.ne 2
.na
\fB\fB/etc/user_attr\fR\fR
.ad
.RS 29n
extended user attributes
.RE

.sp
.ne 2
.na
\fB\fB/etc/security/exec_attr\fR\fR
.ad
.RS 29n
execution profiles
.RE

.sp
.ne 2
.na
\fB\fB/etc/security/policy.conf\fR\fR
.ad
.RS 29n
policy definitions
.RE

.sp
.ne 2
.na
\fB\fB/etc/security/prof_attr\fR\fR
.ad
.RS 29n
profile information
.RE

.SH ATTRIBUTES
.sp
.LP
```

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

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
MT-Level        MT-Safe
.TE

.SH SEE ALSO
.sp
.LP
\fBgetauthattr\fR(3SECDB), \fBgetprofattr\fR(3SECDB), \fBgetuserattr\fR(3SECDB),
\fBkva_match\fR(3SECDB), \fBexec_attr\fR(4), \fBpasswd\fR(4),
\fBpolicy.conf\fR(4), \fBprof_attr\fR(4), \fBuser_attr\fR(4),
\fBattributes\fR(5)
\fBgetauthattr\fR(3SECDB), \fBgetuserattr\fR(3SECDB), \fBkva_match\fR(3SECDB),
\fBexec_attr\fR(4), \fBpasswd\fR(4), \fBpolicy.conf\fR(4), \fBprof_attr\fR(4),
\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
**********************************************************
'\" te
.\" Copyright (c) 2005, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH GETPROFATTR 3SECDB "Aug 13, 2018"
.TH GETPROFATTR 3SECDB "Mar 31, 2005"
.SH NAME
getprofattr, getprofnam, free_profattr, setprofattr, endprofattr, getproflist,
free_proflist \- get profile description and attributes
.SH SYNOPSIS
.LP
.nf
cc [ \fIflag\fR... ] \fIfile\fR... -lsecdb  -lsocket  -lnsl [ \fIlibrary\fR... ]
#include <prof_attr.h>

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

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

.LP
.nf
\fBvoid\fR \fBfree_profattr\fR(\fBprofattr_t *\fR\fIpd\fR);
.fi

.LP
.nf
\fBvoid\fR \fBsetprofattr\fR(\fBvoid\fR);
.fi

.LP
.nf
\fBvoid\fR \fBendprofattr\fR(\fBvoid\fR);
.fi

.LP
.nf
\fBvoid\fR \fBgetproflist\fR(\fBconst char *\fR\fIprofname\fR, \fBchar **\fR\fIp
.fi

.LP
.nf
\fBvoid\fR \fBfree_proflist\fR(\fBchar **\fR\fIproflist\fR, \fBint\fR \fIprofcnt
.fi

.SH DESCRIPTION
.sp
.LP
The \fBgetprofattr()\fR and \fBgetprofnam()\fR functions each return a
\fBprof_attr\fR entry. Entries can come from any of the sources specified in
the \fBnsswitch.conf\fR(4) file.
.sp
.LP
The \fBgetprofattr()\fR function enumerates \fBprof_attr\fR entries. The
\fBgetprofnam()\fR function searches for a \fBprof_attr\fR entry with a given
\fIname\fR. Successive calls to these functions return either successive
\fBprof_attr\fR entries or \fINULL\fR.
```

```
.sp
.LP
The internal representation of a \fBprof_attr\fR entry is a \fBprofattr_t\fR
structure defined in <\fBprof_attr.h\fR> with the following members:
.sp
.in +2
.nf
char     *name;   /* Name of the profile */
char     *res1;   /* Reserved for future use */
char     *res2;   /* Reserved for future use */
char     *desc;   /* Description/Purpose of the profile */
kva_t *attr;   /* Profile attributes */
.fi
.in -2

.sp
.LP
The \fBfree_profattr()\fR function releases memory allocated by the
\fBgetprofattr()\fR and \fBgetprofnam()\fR functions.
.sp
.LP
The \fBsetprofattr()\fR function "rewinds" to the beginning of the enumeration
of \fBprof_attr\fR entries. Calls to \fBgetprofnam()\fR can leave the
enumeration in an indeterminate state. Therefore, \fBsetprofattr()\fR should
be called before the first call to \fBgetprofattr()\fR.
.sp
.LP
The \fBendprofattr()\fR function may be called to indicate that \fBprof_attr\fR
processing is complete; the system may then close any open \fBprof_attr\fR
file, deallocate storage, and so forth.
.sp
.LP
The \fBgetproflist()\fR function searches for the list of sub-profiles found in
the given \fIprofname\fR and allocates memory to store this list in
\fIproflist\fR. The given \fIprofname\fR will be included in the list of
sub-profiles. The \fIprofcnt\fR argument indicates the number of items
currently valid in \fIproflist\fR. Memory allocated by \fBgetproflist()\fR
should be freed using the \fBfree_proflist()\fR function.
.sp
.LP
The \fBfree_proflist()\fR function frees memory allocated by the
\fBgetproflist()\fR function.  The \fIprofcnt\fR argument specifies the number
of items to free from the \fIproflist\fR argument.
.SH RETURN VALUES
.sp
.LP
The \fBgetprofattr()\fR function returns a pointer to a \fBprofattr_t\fR if it
successfully enumerates an entry; otherwise it returns \fINULL\fR, indicating
the end of the enumeration.
.sp
.LP
The \fBgetprofnam()\fR function returns a pointer to a \fBprofattr_t\fR if it
successfully locates the requested entry; otherwise it returns \fINULL\fR.
.SH USAGE
.sp
.LP
Individual attributes in the \fBprofattr_t\fR structure can be referred to by
Individual attributes in the \fBprof_attr_t\fR structure can be referred to by
calling the \fBkva_match\fR(3SECDB) function.
.sp
.LP
Because the list of legal keys is likely to expand, any code must be written
to ignore unknown key-value pairs without error.
.sp
.LP
The \fBgetprofattr()\fR and \fBgetprofnam()\fR functions both allocate memory
```

```
for the pointers they return. This memory should be deallocated with the
\fBfree_profattr()\fR function.
.SH FILES
.sp
.ne 2
.na
\fB\fB/etc/security/prof_attr\fR\fR
.ad
.RS 27n
profiles and their descriptions
.RE

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
MT-Level        MT-Safe
.TE

.SH SEE ALSO
.sp
.LP
\fBauths\fR(1), \fBprofiles\fR(1), \fBgetexecattr\fR(3SECDB),
\fBgetauthattr\fR(3SECDB), \fBkva_match\fR(3SECDB), \fBprof_attr\fR(4),
\fBattributes\fR(5)
\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
**********************************************************
’\" te
.\" Copyright (c) 2005, Sun Microsystems, Inc.  All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH GETUSERATTR 3SECDB "Aug 13, 2018"
.TH GETUSERATTR 3SECDB "Mar 31, 2005"
.SH NAME
getuserattr, getusernam, getuseruid, free_userattr, setuserattr, enduserattr,
fgetuserattr \- get user_attr entry
.SH SYNOPSIS
.LP
.nf
cc [ \fIflag\fR... ] \fIfile\fR... -lsecdb  -lsocket  -lnsl [ \fIlibrary\fR... ]
#include <user_attr.h>

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

.LP
.nf
\fBuserattr_t *\fR\fBgetusernam\fR(\fBconst char *\fR\fI\fR\fIname\fR);
.fi

.LP
.nf
\fBuserattr_t *\fR\fBgetuseruid\fR(\fBuid_t\fR \fIuid\fR);
.fi

.LP
.nf
\fBvoid\fR \fBfree_userattr\fR(\fBuserattr_t *\fR\fIuserattr\fR);
.fi

.LP
.nf
\fBvoid\fR \fBsetuserattr\fR(\fBvoid\fR);
.fi

.LP
.nf
\fBvoid\fR \fBenduserattr\fR(\fBvoid\fR);
.fi

.LP
.nf
\fBuserattr_t *\fR\fBfgetuserattr\fR(\fBFILE *\fR\fIf\fR);
.fi

.SH DESCRIPTION
.sp
.LP
The \fBgetuserattr()\fR, \fBgetusernam()\fR, and \fBgetuseruid()\fR functions
each return a \fBuser_attr\fR(4) entry. Entries can come from any of the
sources specified in the \fBnsswitch.conf\fR(4) file. The \fBgetuserattr()\fR
function enumerates \fBuser_attr\fR entries. The \fBgetusernam()\fR function
searches for a \fBuser_attr\fR entry with a given user name \fIname\fR. The
\fBgetuseruid()\fR function searches for a \fBuser_attr\fR entry with a given
user ID \fIuid\fR. Successive calls to these functions return either successive
\fBuser_attr\fR entries or \fINULL\fR.
.sp
```

```
.LP
The \fBfgetuserattr()\fR function does not use \fBnsswitch.conf\fR but reads
and parses the next line from the stream \fIf\fR. This stream is assumed to
have the format of the \fBuser_attr\fR files.
.sp
.LP
The \fBfree_userattr()\fR function releases memory allocated by the
\fBgetusernam()\fR, \fBgetuserattr()\fR, and \fBfgetuserattr()\fR functions.
.sp
.LP
The internal representation of a \fBuser_attr\fR entry is a \fBuserattr_t\fR
structure defined in <\fBuser_attr.h\fR> with the following members:
.sp
.in +2
.nf
char  *name;       /* name of the user */
char  *qualifier;  /* reserved for future use */
char  *res1;       /* reserved for future use */
char  *res2;       /* reserved for future use */
kva_t *attr;       /* list of attributes */
.fi
.in -2

.sp
.LP
The \fBsetuserattr()\fR function "rewinds" to the beginning of the enumeration
of \fBuser_attr\fR entries.  Calls to \fBgetusernam()\fR may leave the
enumeration in an indeterminate state, so \fBsetuserattr()\fR should be called
before the first call to \fBgetuserattr()\fR.
.sp
.LP
The \fBenduserattr()\fR function may be called to indicate that \fBuser_attr\fR
processing is complete; the library may then close any open \fBuser_attr\fR
file, deallocate any internal storage, and so forth.
.SH RETURN VALUES
.sp
.LP
The \fBgetuserattr()\fR function returns a pointer to a \fBuserattr_t\fR if it
successfully enumerates an entry; otherwise it returns \fINULL\fR, indicating
the end of the enumeration.
.sp
.LP
The \fBgetusernam()\fR function returns a pointer to a \fBuserattr_t\fR if it
successfully locates the requested entry; otherwise it returns \fINULL\fR.
.SH USAGE
.sp
.LP
The \fBgetuserattr()\fR and \fBgetusernam()\fR functions both allocate memory
for the pointers they return. This memory should be deallocated with the
\fBfree_userattr()\fR function.
.sp
.LP
Individual attributes can be referenced in the \fBattr\fR structure by calling
the \fBkva_match\fR(3SECDB) function.
.SH WARNINGS
.SH WARININGS
.sp
.LP
Because the list of legal keys is likely to expand, code must be written to
ignore unknown key-value pairs without error.
.SH FILES
.sp
.ne 2
.na
\fB\fB/etc/user_attr\fR\fR
.ad
```

```
.RS 22n
extended user attributes
.RE

.sp
.ne 2
.na
\fB\fB/etc/nsswitch.conf\fR\fR
.ad
.RS 22n
configuration file lookup information for the name service switch
configuration file lookup information for the name server switch
.RE

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
MT-Level        MT-Safe
.TE

.SH SEE ALSO
.sp
.LP
\fBgetauthattr\fR(3SECDB), \fBgetexecattr\fR(3SECDB),
\fBgetprofattr\fR(3SECDB), \fBkva_match\fR(3SECDB), \fBuser_attr\fR(4),
\fBattributes\fR(5)
\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
**********************************************************
'\" te
.\"  Copyright (c) 1999 Sun Microsystems, Inc.  All Rights Reserved
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH KVA_MATCH 3SECDB "Aug 13, 2018"
.TH KVA_MATCH 3SECDB "Aug 12, 1999"
.SH NAME
kva_match \- look up a key in a key-value array
.SH SYNOPSIS
.LP
.nf
cc [ \fIflag\fR... ] \fIfile\fR... -lsecdb [ \fIlibrary\fR... ]
cc [ \fIflag\fR... ] \fIfile\fR...- lsecdb [ \fIlibrary\fR... ]
#include <secdb.h>

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

.SH DESCRIPTION
.sp
.LP
The \fBkva_match()\fR function searches a \fBkva_t\fR structure, which is part
of the  \fBauthattr_t\fR, \fBexecattr_t\fR, \fBprofattr_t\fR, or
\fBuserattr_t\fR structures. The function takes two arguments: a pointer to a
key value array, and a key.  If the key is in the array, the function returns a
pointer to the first corresponding value that matches that key.  Otherwise, the
function returns \fINULL\fR.
.SH RETURN VALUES
.sp
.LP
Upon successful completion, the function returns a pointer to the value sought.
Otherwise, it returns \fINULL\fR.
.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
MT-Level        MT-Safe
.TE

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