Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'xauth.1' command

$ man xauth.1

XAUTH(1) General Commands Manual XAUTH(1)

NAME

xauth - X authority file utility

SYNOPSIS

xauth [-f authfile] [-vqibn] [command arg ...]

DESCRIPTION

The xauth program is used to edit and display the authorization infor? mation used in connecting to the X server. This program is usually used to extract authorization records from one machine and merge them in on another (as is the case when using remote logins or granting ac? cess to other users). Commands (described below) may be entered inter? actively, on the xauth command line, or in scripts. Note that this program does not contact the X server except when the generate command is used. Normally xauth is not used to create the authority file entry in the first place; the program that starts the X server (often xdm or startx) does that.

OPTIONS

The following options may be used with xauth. They may be given indi? vidually (e.g., -q -i) or may combined (e.g., -qi).

-f authfile

This option specifies the name of the authority file to use. By default, xauth will use the file specified by the XAUTHORITY environment variable or .Xauthority in the user's home direc?

tory.

-q This option indicates that xauth should operate quietly and not print unsolicited status messages. This is the default if an xauth command is given on the command line or if the standard output is not directed to a terminal.

-v This option indicates that xauth should operate verbosely and print status messages indicating the results of various opera? tions (e.g., how many records have been read in or written out). This is the default if xauth is reading commands from its standard input and its standard output is directed to a terminal.

-i This option indicates that xauth should ignore any authority file locks. Normally, xauth will refuse to read or edit any authority files that have been locked by other programs (usu? ally xdm or another xauth).

-b This option indicates that xauth should attempt to break any authority file locks before proceeding. Use this option only to clean up stale locks.

-n This option indicates that xauth should not attempt to resolve any hostnames, but should simply always print the host address as stored in the authority file.

-V This option shows the version number of the xauth executable.

COMMANDS

The following commands may be used to manipulate authority files:

add displayname protocolname hexkey

An authorization entry for the indicated display using the given protocol and key data is added to the authorization file. The data is specified as an even-lengthed string of hexadecimal digits, each pair representing one octet. The first digit of each pair gives the most significant 4 bits of the octet, and the second digit of the pair gives the least significant 4 bits. For example, a 32 character hexkey would represent a 128-bit value. A protocol name consisting of just a single pe?

riod is treated as an abbreviation for MIT-MAGIC-COOKIE-1.

generate displayname protocolname [trusted|untrusted]

[timeout seconds] [group group-id] [data hexdata]

This command is similar to add. The main difference is that instead of requiring the user to supply the key data, it connects to the server specified in displayname and uses the SECURITY extension in order to get the key data to store in the authorization file. If the server cannot be contacted or if it does not support the SECURITY extension, the command fails. Otherwise, an authorization entry for the indicated display using the given protocol is added to the authorization file. A protocol name consisting of just a single period is treated as an abbreviation for MIT-MAGIC-COOKIE-1.

If the trusted option is used, clients that connect using this authorization will have full run of the display, as usual. If untrusted is used, clients that connect using this authorization will be considered untrusted and prevented from stealing or tampering with data belonging to trusted clients. See the SECURITY extension specification for full details on the restrictions imposed on untrusted clients. The default is untrusted.

The timeout option specifies how long in seconds this authorization will be valid. If the authorization remains unused (no clients are connected with it) for longer than this time period, the server purges the authorization, and future attempts to connect using it will fail. Note that the purging done by the server does not delete the authorization entry from the authorization file. The default timeout is 60 seconds.

The group option specifies the application group that clients connecting with this authorization should belong to. See the application group extension specification for more details. The default is to not belong to an application group.

The data option specifies data that the server should use to

generate the authorization. Note that this is not the same data that gets written to the authorization file. The inter? pretation of this data depends on the authorization protocol. The hexdata is in the same format as the hexkey described in the add command. The default is to send no data.

[n]extract filename displayname...

Authorization entries for each of the specified displays are written to the indicated file. If the nextract command is used, the entries are written in a numeric format suitable for non-binary transmission (such as secure electronic mail). The extracted entries can be read back in using the merge and nmerge commands. If the filename consists of just a single dash, the entries will be written to the standard output.

[n]list [displayname...]

Authorization entries for each of the specified displays (or all if no displays are named) are printed on the standard out? put. If the nlist command is used, entries will be shown in the numeric format used by the nextract command; otherwise, they are shown in a textual format. Key data is always dis? played in the hexadecimal format given in the description of the add command.

[n]merge [filename...]

Authorization entries are read from the specified files and are merged into the authorization database, superseding any match? ing existing entries. If the nmerge command is used, the nu? meric format given in the description of the extract command is used. If a filename consists of just a single dash, the stan? dard input will be read if it hasn't been read before.

remove displayname...

Authorization entries matching the specified displays are re? moved from the authority file.

source filename

The specified file is treated as a script containing xauth com?

mands to execute. Blank lines and lines beginning with a sharp sign (#) are ignored. A single dash may be used to indicate the standard input, if it hasn't already been read.

info Information describing the authorization file, whether or not any changes have been made, and from where xauth commands are being read is printed on the standard output.

exit If any modifications have been made, the authority file is written out (if allowed), and the program exits. An end of file is treated as an implicit exit command.

quit The program exits, ignoring any modifications. This may also be accomplished by pressing the interrupt character.

version This command shows the version number of the xauth executable.

help [string]

A description of all commands that begin with the given string (or all commands if no string is given) is printed on the stan? dard output.

? A short list of the valid commands is printed on the standard output.

DISPLAY NAMES

Display names for the add, [n]extract, [n]list, [n]merge, and remove commands use the same format as the DISPLAY environment variable and the common -display command line argument. Display-specific informa? tion (such as the screen number) is unnecessary and will be ignored. Same-machine connections (such as local-host sockets, shared memory, and the Internet Protocol hostname localhost) are referred to as host? name/unix:displaynumber so that local entries for different machines may be stored in one authority file.

EXAMPLE

The most common use for xauth is to extract the entry for the current display, copy it to another machine, and merge it into the user's au? thority file on the remote machine:

% xauth extract - $DISPLAY | ssh otherhost xauth merge -

The following command contacts the server :0 to create an authorization

using the MIT-MAGIC-COOKIE-1 protocol. Clients that connect with this authorization will be untrusted.

% xauth generate :0 .

ENVIRONMENT

This xauth program uses the following environment variables:

XAUTHORITY

to get the name of the authority file to use if the -f option isn't used.

HOME to get the user's home directory if XAUTHORITY isn't defined.

FILES

$HOME/.Xauthority

default authority file if XAUTHORITY isn't defined.

SEE ALSO

X(7), Xsecurity(7), xhost(1), Xserver(1), xdm(1), startx(1), Xau(3).

BUGS

Users that have unsecure networks should take care to use encrypted file transfer mechanisms to copy authorization entries between ma? chines. Similarly, the MIT-MAGIC-COOKIE-1 protocol is not very useful in unsecure environments. Sites that are interested in additional se? curity may need to use encrypted authorization mechanisms such as Ker? beros.

Spaces are currently not allowed in the protocol name. Quoting could be added for the truly perverse.

AUTHOR

Jim Fulton, MIT X Consortium