NAME

apparmor_parser − loads AppArmor profiles into the kernel

SYNOPSIS

apparmor_parser [options] <command> [profiles]...

apparmor_parser [options] <command>

apparmor_parser [−hv] [−−help] [−−version]

DESCRIPTION

apparmor_parser is used as a general tool to compile, and manage AppArmor policy, including loading new **apparmor.d** (5) profiles into the Linux kernel.

AppArmor profiles restrict the operations available to processes.

The **profiles** are loaded into the Linux kernel by the **apparmor_parser** program. The **profiles** may be specified by file name or a directory name containing a set of profiles. If a directory is specified then the **apparmor_parser** will try to do a profile load for each file in the directory that is not a dot file, or explicitly black listed (*.dpkg−new, *.dpkg−old, *.dpkg−dist, *.dpkg−bak, *.dpkg−remove, *.pacsave, *.pacnew, *.rpmnew, *.rpmsave, *.orig, *.rej, *~). The **apparmor_parser** will fall back to taking input from standard input if a profile or directory is not supplied.

The input supplied to **apparmor_parser** should be in the format described in **apparmor.d** (5).

COMMANDS

The command set is broken into four subcategories.

unprivileged commands
: Commands that don't require any privilege and don't operate on profiles.

unprivileged profile commands
: Commands that operate on a profile either specified on the command line or read from stdin if no profile was specified.

privileged commands
: Commands that require the MAC_ADMIN capability within the affected AppArmor namespace to load policy into the kernel or filesystem write permissions to update the affected privileged files (cache etc).

privileged profile commands
: Commands that require privilege and operate on profiles.

Unprivileged commands

−V, −−version
: Print the version number and exit.

−h, −−help
: Give a quick reference guide.

Unprivileged profile commands

−N, −−names
: Produce a list of policies from a given set of profiles (implies −K).

−p, −−preprocess
: Apply preprocessing to the input profile(s) by flattening includes into the output profile and dump to stdout.

−S, −−stdout
: Writes a binary (cached) profile to stdout (implies −K and −T).

−o file, −−ofile file
: Writes a binary (cached) profile to the specified file (implies −K and −T)

Privileged commands

−−purge−cache
Unconditionally clear out cached profiles.

Privileged profile commands

−a, −−add
Insert the AppArmor definitions given into the kernel. This is the default action. This gives an error message if a AppArmor definition by the same name already exists in the kernel, or if the parser doesn't understand its input. It reports when an addition succeeded.

−r, −−replace
This flag is required if an AppArmor definition by the same name already exists in the kernel; used to replace the definition already in the kernel with the definition given on standard input.

−R, −−remove
This flag is used to remove an AppArmor definition already in the kernel. Note that it still requires a complete AppArmor definition as described in **apparmor.d** (5) even though the contents of the definition aren't used.

OPTIONS

−B, −−binary
Treat the profile files specified on the command line (or stdin if none specified) as binary cache files, produced with the −S or −o options, and load to the kernel as specified by −a, −r, and −R (implies −K and −T).

−C, −−Complain
Force the profile to load in complain mode.

−b n, −−base n
Set the base directory for resolving #include directives defined as relative paths.

−I n, −−Include n
Add element n to the search path when resolving #include directives defined as an absolute paths.

−f n, −−apparmorfs n
Set the location of the apparmor security filesystem (default is ‘‘/sys/kernel/security/apparmor’’).

−M n, −−features−file n
Use the features file located at path ‘‘n’’ (default is /etc/apparmor.d/cache/.features). If the −−cache−loc option is present, the ‘‘.features’’ file in the specified cache directory is used.

−m n, −−match−string n
Only use match features ‘‘n’’.

−n n, −−namespace−string n
Force a profile to load in the namespace ‘‘n’’.

−X, −−readimpliesX
In the case of profiles that are loading on systems were READ_IMPLIES_EXEC is set in the kernel for a given process, load the profile so that any ‘‘r’’ flags are processed as ‘‘mr’’.

−k, −−show−cache
Report the cache processing (hit/miss details) when loading or saving cached profiles.

−K, −−skip−cache
Perform no caching at all: disables −W, implies −T.

−T, −−skip−read−cache
By default, if a profile's cache is found in the location specified by −−cache−loc and the timestamp is newer than the profile, it will be loaded from the cache. This option disables this cache loading behavior.

−W, −−write−cache
Write out cached profiles to the location specified in −−cache−loc. Off by default. In cases where abstractions have been changed, and the parser is running with ‘‘−−replace’’, it may make sense to

also use ‘‘−−skip−read−cache’’ with the ‘‘−−write−cache’’ option.

−−skip−bad−cache
Skip updating the cache if it contains cached profiles in a bad or inconsistent state

−L, −−cache−loc
Set the location(s) of the cache directory. This option can accept a comma separated list of directories, which will be searched in order to find a matching cache. The first matching cache file found is used even if a directory later in the search order may contain a newer cache file.

If multiple directories are specified and −−write−cache has been specified then cache writes will be made to the first directory in the list, all other directories will be treated as read only.

If a cache directory name needs to have a comma as part of the name, it can be specified by using a backslash to escape the comma character in the directory name.

If not specified the cache location defaults to /var/cache/apparmor

−−print−cache−dir
Print the cache directory location. This path will be a subdirectory of the directory specified by −−cache−loc. The subdirectory used will be influenced by the features available in the currently running kernel or by the features specified with the −−match−string or −−features−file options.

−Q, −−skip−kernel−load
Perform all actions except the actual loading of a profile into the kernel. This is useful for testing profile generation, caching, etc, without making changes to the running kernel profiles.

This also removes the need for privilege to execute the commands that manage policy in the kernel

−q, −−quiet
Do not report on the profiles as they are loaded, and not show warnings.

−v, −−verbose
Report on the profiles as they are loaded, and show warnings.

−−warn=n
Enable various warnings during policy compilation. A single dump flag can be specified per −−warn option, but the −−warn flag can be passed multiple times.

```
apparmor_parser --warn=rules-not-enforced ...
```

Use −−help=warn to see a full list of which warn flags are supported.

−d, −−debug
Given once, only checks the profiles to ensure syntactic correctness. Given twice, dumps its interpretation of the profile for checking.

−D n, −−dump=n
Debug flag for dumping various structures and passes of policy compilation. A single dump flag can be specified per −−dump option, but the dump flag can be passed multiple times. Note progress flags tend to also imply the matching stats flag.

```
apparmor_parser --dump=dfa-stats --dump=trans-stats <file>
```

Use −−help=dump to see a full list of which dump flags are supported

−j n, −−jobs=n
Set the number of jobs used to compile the specified policy. Where n can be

```
#    - a specific number of jobs
auto - the # of cpus in the in the system
x#   - # * number of cpus
```

Eg.
−j8 OR −−jobs=8 allows for 8 parallel jobs

−jauto OR −−jobs=auto sets the jobs to the # of cpus
−jx4 OR −−jobs=x4 sets the jobs to # of cpus * 4
−jx1 is equivalent to −jauto

The default value is the number of cpus in the system.

−−max−jobs n
Set a hard cap on the value that can be specified by the −−jobs flag. It takes the same set of options available to the −−jobs option, and defaults to 8*cpus

−O n, −−optimize=n
Set the optimization flags used by policy compilation. A single optimization flag can be toggled per −O option, but the optimize flag can be passed multiple times. Turning off some phases of the optimization can make it so that policy can't complete compilation due to size constraints (it is entirely possible to create a dfa with millions of states that will take days or longer to compile).

Note: The parser is set to use a balanced default set of flags, that will result in reasonable compression but not take excessive amounts of time to complete.

Use −−help=optimize to see a full list of which optimization flags are supported.

−−abort−on−error Abort processing of profiles on the first error encountered, otherwise the parser will continue to try to compile other profiles if specified.
Note: If an error is encountered while processing profiles the last error encountered will be used to set the exit code.

−−skip−bad−cache−rebuild The default behavior of the parser is to check if a cached version of a profile exists and if it does it attempt to load it into the kernel. If that load is rejected, then the parser will attempt to rebuild the cache file, and load again.
This option tells the parser to not attempt to rebuild the cache on failure, instead the parser continues on with processing the remaining profiles.

−−config−file
Specify the config file to use instead of /etc/apparmor/parser.conf. This option will be processed early before regular options regardless of the order it is specified in.

−−print−config−file
Print the config file location that will be used.

CONFIG FILE

An optional config file /etc/apparmor/parser.conf can be used to specify the default options for the parser, which then can be overridden using the command line options.

The config file ignores leading whitespace and treats lines that begin with # as comments. Config options are specified one per line using the same format as the longform command line options (without the preceding −−).

Eg.
#comment

```
optimize=no-expr-tree
optimize=compress-fast
```

As with the command line some options accumulate and others override, ie. when there are conflicting versions of switch the last option is the one chosen.

Eg.
Optimize=no−minimize
Optimize=minimize

would result in Optimize=minimize being set.

The Include, Dump, and Optimize options accululate except for the inversion option (no-X vs. X), and a couple options that work by setting/clearing multiple options (compress-small). In that case the option will

override the flags it sets but will may accumulate with others.

All other options override previously set values.

BUGS

If you find any bugs, please report them at <https://bugs.launchpad.net/apparmor/+filebug>.

SEE ALSO

apparmor (7), **apparmor.d** (5), **aa_change_hat** (2), and <https://wiki.apparmor.net>.