Apparmor

Apparmor is a Mandatory Access Control (or MAC) system. It uses LSM kernel enhancements to restrict programs to certain resources. AppArmor does this with profiles loaded into the kernel when the system starts. Apparmor has two types of profile modes, enforcement and complain. Profiles in enforcement mode *enforce* that profile's rules and report violation attempts in syslog or auditd. Profiles in complain mode don't enforce any profile rules, just log violation attempts.

In Ubuntu Apparmor is installed by default. It confines applications to profiles to determine what files and permissions that a program needs access to. Some applications will come with their own properties and more can be found in the apparmor-profiles package.

You can install apparmor-profiles by running sudo apt-get install apparmor-profiles.

I found a good example of Apparmor on the Ubuntu forums that I rewrote for this post.

Apparmor is a security framework that prevents applications from turning evil. For example: If I run Firefox and visit a bad site that tries to install malware that will delete my home folder, Apparmor has limits on Firefox though preventing it from doing anything I don't want (like accessing my music, documents, etc). This way even if your application is compromised, no harm can be done.

**How it works**

The apparmor-utils package contains command line tools for configuring Apparmor. Using it you can change Apparmor's execution mode, find the status of a profile create new profiles, etc.

These are the most common commands:

***NOTE:***: Profiles are stored in /etc/apparmor.d/

* You can check Apparmor's status with sudo apparmor\_status. You will get a list of all profiles \* loaded, all profiles in enforce mode, all profiles in complain mode, what processes are defined in enforce/complain, etc.
* To put a profile in *complain* mode you use sudo aa-complain /path/to/bin, where /path/to/bin is the programs bin folder. For example, running: sudo aa-complain /usr/bin/firefox will put Firefox in complain mode.
* You use sudo aa-enforce /path/to/bin to enforce a programs profile.
* You can load *all* profiles into complain/enforce modes with sudo aa-complain /etc/apparmor.d/\* and sudo aa-enforce.d/\* respectively.

To load a profile into the kernel you would use apparmor\_parser. You can reload profiles using the -r parameter.

* To load a profile use: cat /etc/apparmor.d/profile.name | sudo apparmor\_parser -a, which effectively prints the contents of profile.name into Apparmor's parser.
* To reload a profile you use the -r parameter, like so: cat /etc/apparmor.d/profile.name | sudo apparmor\_parser -r
* To reload all of Apparmor's profiles use: sudo service apparmor reload

To disable a profile you *link* it to /etc/apparmor.d/disable/ using ln like this: sudo ln -s /etc/apparmor.d/profile.name /etc/apparmor.d/disable/ then run: sudo apparmor\_parser -R /etc/apparmor.d/profile.name.

***NOTE:*** Do not confuse **apparmor\_parser -r** with **apparmor\_parser -R** THEY ARE NOT THE SAME THING!

* To re-enable a profile, remove the symbolic link to it in /etc/apparmor.d/disable/ then load it using the -a parameter. sudo rm /etc/apparmor.d/disable/profile.name cat /etc/apparmor.d/profile.name | sudo apparmor\_parser -a
* You can disable Apparmor with sudo service apparmor stop and remove the kernel module using sudo update-rc.d -f apparmor defaults
* Start Apparmor with sudo service apparmor start and load kernel modules with sudo update-rc.d apparmor defaults

**Profiles**

Profiles are stored in /etc/apparmor.d/ and are named after the full path to the executable they profile, replacing '/' with '.'. For example /etc/apparmor.d/bin.ping is the profile for ping in /bin.

There are two main types of entries used in profiles:

1. *Path Entries* determine what files an application can access.
2. *Capability* entries determine what privileges a process can use.

Lets look at the profile for ping, located in etc/apparmor.d/bin.ping, as an example.

#include <tunables/global>

/bin/ping flags=(complain) {

#include <abstractions/base>

#include <abstractions/consoles>

#include <abstractions/nameservice>

capability net\_raw,

capability setuid,

network inet raw,

/bin/ping mixr,

/etc/modules.conf r,

}

#include <tunables/global> Includes the file global in the directory tunables, this allows statements pertaining to multiple applications to be placed in a common file.

/bin/ping flags=(complain)sets the path to the profiled program and sets the mode to complain.

capability net\_raw allows the application access to the CAP\_NET\_RAW Posix.1e capability.

/bin/ping mixr allows the application read and execute access to the file.

/etc/modules.conf r, The r gives the application *read* privileges for /etc/modules.conf

***NOTE:*** After creating/editing a profile, you need to reload the profile for changes to take effect.

Here is a list of permissions you can use:

* r - read
* w - write
* ux - Unconstrained Execute
* Ux - Unconstrained Execute -- scrub the environment
* px - Discrete profile execute
* Px - Discrete profile execute -- scrub the environment
* ix - Inherit execute
* m - allow PROT\_EXEC with mmap(2) calls
* l - link