15.9.2 Application Armor Facts

Application Armor (AppArmor) can be used a some Linux distributions to protect your Linux systems from untrusted or insecure processes.

This lesson covers the following topics:

- Understanding AppArmor profiles
- AppArmor Modes
- AppArmor status commands
- AppArmor management commands

Understanding AppArmor Profiles

AppArmor is a Mandatory Access Control (MAC) method used to protect your Linux systems from untrusted or insecure processes. It does this by locking down an application and the files to be accessed with absolute path names, followed by the common read and write access modes. In other words, the Linux kernel queries AppArmor before each system call to determine whether the process being requested is authorized to do the given operation.

To accomplish all of this, AppArmor applies a set of rules, known as a profile, on each program being protected. These plain-text profiles are stored in the /etc/apparmor.d directory and contain a list of access control rules on the applicable resources that each program can make use of. Additional profiles can be typically downloaded or new profiles can be created using the aa-genprof utility.

AppArmor Modes

AppArmor can operate in the following modes:

Mode	Description
enforce	When using this mode, the setting specified in the profiles will prevent applications from taking any restricted actions.
complain	This mode, also known as the learning mode, uses the profiles loaded, but they are not enforced. Instead, any policy violation attempts are stored in a log file. This is useful, for example, when you want to test an AppArmor profile, because it lets you see any errors that would occur in enforce mode which would be helpful for fine tuning the profile.

AppArmor Status Commands

Use the following commands to view the status of AppArmor and if needed, start AppArmor:

Most of the following commands must be run using root permissions. If you are not logged in as root, you can use sudo to gain root permissions.

Command	Description
systemctl status apparmor	Checks the status of the AppArmor service and also tells you if AppArmor is enabled on boot.
systemctl start apparmor	Use this command to start the apparmor service.
systemctl enable apparmor	This command configures AppArmor to start each time your system is booted.

AppArmor Management Commands

AppArmor supports the following management commands:

Most of the following commands must be run using root permissions. If you are not logged in as root, you can use sudo to gain root permissions.

Command	Description	Example
aa-status	Displays a list of the AppArmor modules that are loaded, and which ones are running in enforce mode and which ones are running in complain mode.	**aa-status**
aa-disable	Disables one or more profiles. This command will unload the profile from the kernel and prevent the profile from being loaded on AppArmor startup. The aa-enforce and aa-complain utilities may be used to change this behavior. This command also moves the profile from the /etc/apparmor.d directory into the /etc/apparmor.d/disable directory.	**aa-disable usr.bin.firefox** Disables the FireFox profile.

aa-complain	Used to set the enforcement mode for one or more profiles to the complain mode. In this mode the security policy is not enforced but instead, access violations are logged to the system log. Note that 'deny' rules will be enforced even in complain mode. This is often used to help troubleshoot applications being protected with AppArmor.	**aa-complain usr.bin.firefox** Sets the Firefox profile to the complain mode.
aa-enforce	Sets one or more profiles to enforce mode. This command is only relevant in conjunction with the aa-complain utility which sets a profile to complain mode and the aa-disable utility which unloads and disables a profile. The default mode for a security policy is enforce and the aa-complain utility must be run to change this behavior.	**aa-enforce usr.bin.firefox** Changes the previously disabled or complain Firefox profile to enable.
aa-unconfined	Uses the ss or netstat command to determine which processes have open network sockets and do not have appArmor profiles loaded into the kernel. This command supports the following options: - **aa-unconfined --with-ss** Runs the unconfined command using the ‘ss’ command to find and list the processes listening on the network sockets. This is the default option used if no options are explicitly entered. - **aa-unconfined --with-netstat** Performs the same function as -ss, but instead of using the ss command it uses the older netstat command to find and list the processes listening on the network sockets. - **aa-unconfined --paranoid** The most thorough option which will display all of the processes from the /proc filesystem with TCP or UDP ports that are not protected by AppArmor profiles.	**aa-unconfined --with-ss** **aa-unconfined --with-netstat** **aa-unconfined --paranoid**

AppArmor Tunables

AppArmor provides a way to fine tune how AppArmor functions without having to adjust your profiles. The items that can be tuned are stored in files located in /etc/apparmor.d/tunables directory and are text-based files.

Example
One common item often tuned is the location of user’s home directory.
By default, many implementations of AppArmor will list /home as the location for all user’s home directory. However, if you have some users who are using a different location for home directors, such as /exports/home, you can edit the tunable file named home to include the additional location.

For example, by default, the home file may only include **@{HOMEDIRS}=/home/**. To include the alternative path of /exports/home, you would simply edit the file named home, and on the @home line, add a space and the path to the alternative home directory, such as **@{HOMEDIRS}=/home/ /exports/home**.