AppArmor

2.1

www.novell.com

Novell AppArmor Administration Guide

SUSE
A NOVELL BUSINESS

Novell AppArmor Administration Guide

Copyright © 2006-2007 Novell, Inc.

Contents

About This Guide

Novell® AppArmor is designed to provide easy-to-use application security for both servers and workstations. Novell AppArmor is an access control system that lets you specify per program which files the program may read, write, and execute. AppArmor secures applications by enforcing good application behavior without relying on attack signatures, so it can prevent attacks even if they are exploiting previously unknown vulnerabilities.

Novell AppArmor consists of:

- A library of AppArmor profiles for common Linux* applications describing what files the program needs to access.

- A library of AppArmor profile foundation classes (profile building blocks) needed for common application activities, such as DNS lookup and user authentication.

- A tool suite for developing and enhancing AppArmor profiles, so that you can change the existing profiles to suit your needs and create new profiles for your own local and custom applications.

- Several specially modified applications that are AppArmor enabled to provide enhanced security in the form of unique subprocess confinement, including Apache and Tomcat.

- The Novell AppArmor–loadable kernel module and associated control scripts to enforce AppArmor policies on your openSUSE® system.

This guide covers the following topics:

Immunizing Programs
 Describes the operation of Novell AppArmor and describes the types of programs that should have Novell AppArmor profiles created for them.

Profile Components and Syntax
 Introduces the profile components and syntax.

Building and Managing Profiles with YaST
 Describes how to use the AppArmor YaST modules to build, maintain and update profiles.

Building Profiles from the Command Line
> Describes how to use the AppArmor command line tools to build, maintain and update profiles.

Profiling Your Web Applications Using ChangeHat
> Enables you to create subprofiles for the Apache Web server that allow you to tightly confine small sections of Web application processing.

Managing Profiled Applications
> Describes how to perform Novell AppArmor profile maintenance, which involves tracking common issues and concerns.

Support
> Indicates support options for this product.

Glossary
> Provides a list of terms and their definitions.

1 Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation and enter your comments there.

2 Documentation Conventions

The following typographical conventions are used in this manual:

- `/etc/passwd`: filenames and directory names

- *`placeholder`*: replace *`placeholder`* with the actual value

- `PATH`: the environment variable PATH

- `ls`, `--help`: commands, options, and parameters

- `user`: users or groups

- Alt, Alt + F1: a key to press or a key combination; keys are shown in uppercase as on a keyboard

- *File*, *File > Save As*: menu items, buttons

- *Dancing Penguins* (Chapter *Penguins*, ↑Another Manual): This is a reference to a chapter in another manual.

3 Source Code

The source code of openSUSE is publicly available. To download the source code, proceed as outlined under http://www.novell.com/products/suselinux/source_code.html. If requested we send you the source code on a DVD. We need to charge a \$15 or €15 fee for creation, handling and postage. To request a DVD of the source code, send an e-mail to sourcedvd@suse.de [mailto:sourcedvd@suse.de] or mail the request to:

```
SUSE Linux Products GmbH
Product Management openSUSE
Maxfeldstr. 5
D-90409 Nürnberg
Germany
```

1

Immunizing Programs

Novell® AppArmor provides immunization technologies that protect applications from the inherent vulnerabilities they possess. After installing Novell AppArmor, setting up Novell AppArmor profiles, and rebooting the computer, your system becomes immunized because it begins to enforce the Novell AppArmor security policies. Protecting programs with Novell AppArmor is referred to as *immunizing*.

Novell AppArmor sets up a collection of default application profiles to protect standard Linux services. To protect other applications, use the Novell AppArmor tools to create profiles for the applications that you want protected. This chapter introduces the philosophy of immunizing programs. Proceed to Chapter 2, *Profile Components and Syntax* (page 11), Chapter 3, *Building and Managing Profiles with YaST* (page 27), or Chapter 4, *Building Profiles from the Command Line* (page 49) if you are ready to build and manage Novell AppArmor profiles.

Novell AppArmor provides streamlined access control for network services by specifying which files each program is allowed to read, write, and execute, and which type of network it is allowed to access. This ensures that each program does what it is supposed to do and nothing else. Novell AppArmor quarantines programs to protect the rest of the system from being damaged by a compromised process.

Novell AppArmor is a host intrusion prevention or mandatory access control scheme. Previously, access control schemes were centered around users because they were built for large timeshare systems. Alternatively, modern network servers largely do not permit users to log in, but instead provide a variety of network services for users, such as Web, mail, file, and print servers. Novell AppArmor controls the access given to network services and other programs to prevent weaknesses from being exploited.

TIP: Background Information for Novell AppArmor

To get a more in-depth overview of AppArmor and the overall concept behind it, refer to Appendix A, *Background Information on AppArmor Profiling* (page 127).

1.1 Introducing the AppArmor Framework

This section provides a very basic understanding of what is happening "behind the scenes" (and under the hood of the YaST interface) when you run AppArmor.

An AppArmor profile is a plain text file containing path entries and access permissions. See Section 2.1, "Breaking a Novell AppArmor Profile into Its Parts" (page 12) for a detailed reference profile. The directives contained in this text file are then enforced by the AppArmor routines to quarantine the process or program.

The following tools interact in the building and enforcement of AppArmor profiles and policies:

aa-unconfined

aa-unconfined detects any application running on your system that listens for network connections and is not protected by an AppArmor profile. Refer to Section "aa-unconfined—Identifying Unprotected Processes" (page 73) for detailed information about this tool.

aa-autodep

aa-autodep creates a basic skeleton of a profile that needs to be fleshed out before it is put to productive use. The resulting profile is loaded and put into complain mode, reporting any behavior of the application that is not (yet) covered by AppArmor rules. Refer to Section "aa-autodep—Creating Approximate Profiles" (page 56) for detailed information about this tool.

aa-genprof

aa-genprof generates a basic profile and asks you to refine this profile by executing the application, generating log events that need to be taken care of by AppArmor policies. You are guided through a series of questions to deal with the log events

that have been triggered during the application's execution. After the profile has been generated, it is loaded and put into enforce mode. Refer to Section "aa-gen-prof—Generating Profiles" (page 59) for detailed information about this tool.

aa-logprof

aa-logprof interactively scans and reviews the log entries generated by an application that is confined by an AppArmor profile in complain mode. It assists you in generating new entries in the profile concerned. Refer to Section "aa-logprof—Scanning the System Log" (page 67) for detailed information about this tool.

aa-complain

aa-complain toggles the mode of an AppArmor profile from enforce to complain. Exceptions to rules set in a profile are logged, but the profile is not enforced. Refer to Section "aa-complain—Entering Complain or Learning Mode" (page 57) for detailed information about this tool.

aa-enforce

aa-enforce toggles the mode of an AppArmor profile from complain to enforce. Exceptions to rules set in a profile are logged, but not permitted—the profile is enforced. Refer to Section "aa-enforce—Entering Enforce Mode" (page 58) for detailed information about this tool.

Once a profile has been built and is loaded, there are two ways in which it can get processed:

complain

In complain mode, violations of AppArmor profile rules, such as the profiled program accessing files not permitted by the profile, are detected. The violations are permitted, but also logged. To improve the profile, turn complain mode on, run the program through a suite of tests to generate log events that characterize the program's access needs, then postprocess the log with the AppArmor tools (YaST or aa-log-prof) to transform log events into improved profiles.

enforce

In enforce mode, violations of AppArmor profile rules, such as the profiled program accessing files not permitted by the profile, are detected. The violations are logged and not permitted. The default is for enforce mode to be enabled. To log the violations only, but still permit them, use complain mode. Enforce toggles with complain mode.

1.2 Determining Programs to Immunize

Now that you have familiarized yourself with AppArmor, start selecting the applications for which to build profiles. Programs that need profiling are those that mediate privilege. The following programs have access to resources that the person using the program does not have, so they grant the privilege to the user when used:

cron Jobs
> Programs that are run periodically by cron. Such programs read input from a variety of sources and can run with special privileges, sometimes with as much as `root` privilege. For example, cron can run `/usr/sbin/logrotate` daily to rotate, compress, or even mail system logs. For instructions for finding these types of programs, refer to .

Web Applications
> Programs that can be invoked through a Web browser, including CGI Perl scripts, PHP pages, and more complex Web applications. For instructions for finding these types of programs, refer to .

Network Agents
> Programs (servers and clients) that have open network ports. User clients, such as mail clients and Web browsers mediate privilege. These programs run with the privilege to write to the user's home directory and they process input from potentially hostile remote sources, such as hostile Web sites and e-mailed malicious code. For instructions for finding these types of programs, refer to .

Conversely, unprivileged programs do not need to be profiled. For instance, a shell script might invoke the `cp` program to copy a file. Because `cp` does not have its own profile, it inherits the profile of the parent shell script, so can copy any files that the parent shell script's profile can read and write.

1.3 Immunizing cron Jobs

To find programs that are run by cron, inspect your local cron configuration. Unfortunately, cron configuration is rather complex, so there are numerous files to inspect. Periodic cron jobs are run from these files:

```
/etc/crontab
/etc/cron.d/*
/etc/cron.daily/*
/etc/cron.hourly/*
/etc/cron.monthly/*
/etc/cron.weekly/*
```

For `root`'s cron jobs, edit the tasks with `crontab -e` and list `root`'s cron tasks with `crontab -l`. You must be `root` for these to work.

Once you find these programs, you can use the *Add Profile Wizard* to create profiles for them. Refer to Section 3.1, "Adding a Profile Using the Wizard" (page 29).

1.4 Immunizing Network Applications

An automated method for finding network server daemons that should be profiled is to use the aa-unconfined tool. You can also simply view a report of this information in the YaST module (refer to Section "Application Audit Report" (page 97) for instructions).

The aa-unconfined tool uses the command `netstat -nlp` to inspect your open ports from inside your computer, detect the programs associated with those ports, and inspect the set of Novell AppArmor profiles that you have loaded. aa-unconfined then reports these programs along with the Novell AppArmor profile associated with each program or reports "none" if the program is not confined.

NOTE

If you create a new profile, you must restart the program that has been profiled to have it be effectively confined by AppArmor.

Below is a sample aa-unconfined output:

```
2325 /sbin/portmap not confined
3702❶ /usr/sbin/sshd❷ confined
   by '/usr/sbin/sshd❸ (enforce)'
4040 /usr/sbin/ntpd confined by '/usr/sbin/ntpd (enforce)'
4373 /usr/lib/postfix/master confined by '/usr/lib/postfix/master (enforce)'

4505 /usr/sbin/httpd2-prefork confined by '/usr/sbin/httpd2-prefork (enforce)'
5274 /sbin/dhcpcd not confined
5592 /usr/bin/ssh not confined
7146 /usr/sbin/cupsd confined by '/usr/sbin/cupsd (complain)'
```

❶   The first portion is a number. This number is the process ID number (PID) of the listening program.

❷   The second portion is a string that represents the absolute path of the listening program

❸   The final portion indicates the profile confining the program, if any.

NOTE

aa-unconfined requires `root` privileges and should not be run from a shell that is confined by an AppArmor profile.

---

aa-unconfined does not distinguish between one network interface and another, so it reports all unconfined processes, even those that might be listening to an internal LAN interface.

Finding user network client applications is dependent on your user preferences. The aa-unconfined tool detects and reports network ports opened by client applications, but only those client applications that are running at the time the aa-unconfined analysis is performed. This is a problem because network services tend to be running all the time, while network client applications tend only to be running when the user is interested in them.

Applying Novell AppArmor profiles to user network client applications is also dependent on user preferences. Therefore, we leave profiling of user network client applications as an exercise for the user.

To aggressively confine desktop applications, the aa-unconfined command supports a `paranoid` option, which reports all processes running and the corresponding App-Armor profiles that might or might not be associated with each process. The user can then decide whether each of these programs needs an AppArmor profile.

If you have new or modified profiles, you can submit them to the apparmor-general@forge.novell.com [mailto:apparmor-general@forge.novell.com] mailing list along with a use case for the application behavior that you exercised. The AppArmor team reviews and may submit the work into openSUSE. We cannot guarantee that every profile will be included, but we make a sincere effort to include as much as possible so that end users can contribute to the security profiles that ship in openSUSE.

Alternatively, use the AppArmor profile repository to make your profiles available to other users and to download profiles created by other AppArmor users and the AppArmor developers. Refer to Section 2.5, "Using the External AppArmor Profile Repository" (page 23) for more information on how to use the AppArmor profile repository.

1.4.1 Immunizing Web Applications

To find Web applications, investigate your Web server configuration. The Apache Web server is highly configurable and Web applications can be stored in many directories, depending on your local configuration. openSUSE, by default, stores Web applications in `/srv/www/cgi-bin/`. To the maximum extent possible, each Web application should have an Novell AppArmor profile.

Once you find these programs, you can use the AppArmor *Add Profile Wizard* to create profiles for them. Refer to Section 3.1, "Adding a Profile Using the Wizard" (page 29).

Because CGI programs are executed by the Apache Web server, the profile for Apache itself, `usr.sbin.httpd2-prefork` for Apache2 on openSUSE, must be modified to add execute permissions to each of these programs. For instance, adding the line `/srv/www/cgi-bin/my_hit_counter.pl rpx` grants Apache permission to execute the Perl script `my_hit_counter.pl` and requires that there be a dedicated profile for `my_hit_counter.pl`. If `my_hit_counter.pl` does not have a dedicated profile associated with it, the rule should say `/srv/www/cgi-bin/my_hit_counter.pl rix` to cause `my_hit_counter.pl` to inherit the `usr.sbin.httpd2-prefork` profile.

Some users might find it inconvenient to specify execute permission for every CGI script that Apache might invoke. Instead, the administrator can grant controlled access to collections of CGI scripts. For instance, adding the line `/srv/www/cgi-bin/*.{pl,py,pyc} rix` allows Apache to execute all files in `/srv/www/cgi-bin/` ending in `.pl` (Perl scripts) and `.py` or `.pyc` (Python scripts). As above, the `ix` part of the rule causes Python scripts to inherit the Apache profile, which is appropriate if you do not want to write individual profiles for each Python script.

NOTE

If you want the subprocess confinement module (`apache2-mod-apparmor`) functionality when Web applications handle Apache modules (`mod_perl` and `mod_php`), use the ChangeHat features when you add a profile in YaST or at the command line. To take advantage of the subprocess confinement, refer to Section 5.1, "Apache ChangeHat" (page 76).

Profiling Web applications that use `mod_perl` and `mod_php` requires slightly different handling. In this case, the "program" is a script interpreted directly by the module within the Apache process, so no exec happens. Instead, the Novell AppArmor version of Apache calls `change_hat()` using a subprofile (a "hat") corresponding to the name of the URI requested.

NOTE

The name presented for the script to execute might not be the URI, depending on how Apache has been configured for where to look for module scripts. If you have configured your Apache to place scripts in a different place, the different names appear in log file when Novell AppArmor complains about access violations. See Chapter 6, *Managing Profiled Applications* (page 87).

For `mod_perl` and `mod_php` scripts, this is the name of the Perl script or the PHP page requested. For example, adding this subprofile allows the `localtime.php` page to execute and access the local system time:

```
/usr/bin/httpd2-prefork {
  # ...
  ^/cgi-bin/localtime.php {
    /etc/localtime                  r,
    /srv/www/cgi-bin/localtime.php  r,
    /usr/lib/locale/**              r,
  }
}
```

If no subprofile has been defined, the Novell AppArmor version of Apache applies the `DEFAULT_URI` hat. This subprofile is basically sufficient to display an HTML Web page. The `DEFAULT_URI` hat that Novell AppArmor provides by default is the following:

```
^DEFAULT_URI {
  /usr/sbin/suexec2 ixr,
  /var/log/apache2/** rwl,
  /home/*/public_html/**         r,
  /srv/www/htdocs/**             r,
  /srv/www/icons/*.{gif,jpg,png} r,
  /usr/share/apache2/**          r,
}
```

To use a single Novell AppArmor profile for all Web pages and CGI scripts served by Apache, a good approach is to edit the `DEFAULT_URI` subprofile.

1.4.2 Immunizing Network Agents

To find network server daemons and network clients (such as fetchmail, Firefox, amaroK or Banshee) that should be profiled, you should inspect the open ports on your machine, consider the programs that are answering on those ports, and provide profiles for as many of those programs as possible. If you provide profiles for all programs with open network ports, an attacker cannot get to the file system on your machine without passing through a Novell AppArmor profile policy.

Scan your server for open network ports manually from outside the machine using a scanner, such as nmap, or from inside the machine using the `netstat --inet -n -p` command. Then inspect the machine to determine which programs are answering on the discovered open ports.

TIP

Refer to the man page of the `netstat` command for a detailed reference of all possible options.

Profile Components and Syntax

2

You are ready to build Novell AppArmor profiles after you select the programs to profile. To do so, it is important to understand the components and syntax of profiles. AppArmor profiles contain several building blocks that help build simple and reusable profile code: `#include` files, abstractions, program chunks, and capability entries. `#include` statements are used to pull in parts of other AppArmor profiles to simplify the structure of new profiles. Abstractions are `#include` statements grouped by common application tasks. Program chunks are chunks of profiles that are specific to program suites. Capability entries are profile entries for any of the POSIX.1e Linux capabilities.

For help determining the programs to profile, refer to Section 1.2, "Determining Programs to Immunize" (page 4). To start building AppArmor profiles with YaST, proceed to Chapter 3, *Building and Managing Profiles with YaST* (page 27). To build profiles using the AppArmor command line interface, proceed to Chapter 4, *Building Profiles from the Command Line* (page 49).

2.1 Breaking a Novell AppArmor Profile into Its Parts

Novell AppArmor profile components are called Novell AppArmor rules. Currently there are three main types of Novell AppArmor rules, path entries, capability entries, and network entries. Path entries specify what the process can access in the file system and capability entries provide a more fine-grained control over what a confined process is allowed to do through other system calls that require privileges. Includes are a type of meta rule or directives that pull in path and capability entries from other files.

The easiest way of explaining what a profile consists of and how to create one is to show the details of a sample profile, in this case for a hypothetical application called `/usr/bin/foo`:

```
#include <tunables/global>❶

# a comment naming the application to confine
/usr/bin/foo❷
{❸
   #include <abstractions/base>❹

   capability setgid❺,
   network inet tcp❻,

   /bin/mount          ux,
   /dev/{,u}❼random     r,
   /etc/ld.so.cache    r,
   /etc/foo.conf       r,
   /etc/foo/*          r,
   /lib/ld-*.so*       mr,
   /lib/lib*.so*       mr,
   /proc/[0-9]**       r,
   /usr/lib/**         mr,
   /tmp/❽               r,
   /tmp/foo.pid        wr,
   /tmp/foo.*          lrw,
   /@{HOME}❾/.foo_file  rw,
   /@{HOME}/.foo_lock  kw,

  # a comment about foo's subprofile, bar.
   ^bar❿ {
    /lib/ld-*.so*       mr,
    /usr/bin/bar        px,
    /var/spool/*        rwl,
```

```
    }
}
```

❶   This loads a file containing variable definitions.

❷   The normalized path to the program that is confined.

❸   The curly braces ({ }) serve as a container for include statements, subprofiles, path entries, capability entries, and network entries.

❹   This directive pulls in components of AppArmor profiles to simplify profiles.

❺   Capability entry statements enable each of the 29 POSIX.1e draft capabilities.

❻   A directive determining the kind of network access allowed to the application. For details, refer to Section 2.1.1, "Network Access Control" (page 14).

❼   The curly braces ({ }) make this rule apply to the path both with and without the content enclosed by the braces.

❽   A path entry specifying what areas of the file system the program can access. The first part of a path entry specifies the absolute path of a file (including regular expression globbing) and the second part indicates permissible access modes (r for read, w for write, and x for execute). A whitespace of any kind (spaces or tabs) can precede pathnames or separate the pathname from the access modes. Spaces between the access mode and the trailing comma is optional. Find a comprehensive overview of the available access modes in Section 2.1.3, "File Permission Access Modes" (page 17).

❾   This variable expands to a value that can be changed without changing the entire profile.

❿   This section references a subprofile of the application, also known as a "hat". For more details on AppArmor's ChangeHat feature, refer to Chapter 5, *Profiling Your Web Applications Using ChangeHat* (page 75).

TIP: Using Variables in Profiles

With the current AppArmor tools, variables as presented in the above example can only be used when manually editing and maintaining a profile.

A typical example when variables come in handy are network scenarios in which user home directories are not mounted in the standard location /home/ username, but under a custom location. Find the variable definitions for this

use case (`@{HOME}` and `@{HOMEDIRS}`) in the `/etc/apparmor.d/tunables/home` file.

When a profile is created for a program, the program can access only the files, modes, and POSIX capabilities specified in the profile. These restrictions are in addition to the native Linux access controls.

Example: To gain the capability `CAP_CHOWN`, the program must have both access to `CAP_CHOWN` under conventional Linux access controls (typically, be a `root`-owned process) and have the capability chown in its profile. Similarly, to be able to write to the file `/foo/bar` the program must have both the correct user ID and mode bits set in the files attributes (see the `chmod` and `chown` man pages) and have `/foo/bar` `w` in its profile.

Attempts to violate Novell AppArmor rules are recorded in `/var/log/audit/audit.log` if the `audit` package is installed or otherwise in `/var/log/messages`. In many cases, Novell AppArmor rules prevent an attack from working because necessary files are not accessible and, in all cases, Novell AppArmor confinement restricts the damage that the attacker can do to the set of files permitted by Novell AppArmor.

2.1.1 Network Access Control

AppArmor allows mediation of network access based on the address type and family. The following illustrates the network access rule syntax:

```
network [[<domain>❶][<type❷>][<protocol❸>]]
```

❶   Supported domains: `inet`, `ax25`, `ipx`, `appletalk`, `netrom`, `bridge`, `x25`, `inet6`, `rose`, `netbeui`, `security`, `key`, `packet`, `ash`, `econet`, `atmsvc`, `sna`, `irda`, `pppox`, `wanpipe`, `bluetooth`

❷   Supported types: `stream`, `dgram`, `seqpacket`, `rdm`, `raw`, `packet`

❸   Supported protocols: `tcp`, `udp`, `icmp`

The AppArmor tools support only family and type specification. The AppArmor module emits only `network` *domain* *type* in "access denied" messages. And only these are output by the profile generation tools, both YaST and command line.

The following examples illustrate possible network-related rules to be used in AppArmor profiles. Note that the syntax of the two last ones is not currently supported by the AppArmor tools.

```
network❶,
network inet❷,
network inet6❸,
network inet stream❹,
network inet tcp❺,
network tcp❻,
```

❶ Allow all networking. No restrictions applied with regards to domain, type, or protocol.

❷ Allow general use of IPv4 networking.

❸ Allow general use of IPv6 networking.

❹ Allow the use of IPv4 TCP networking.

❺ Allow the use of IPv4 TCP networking, paraphrasing the rule above.

❻ Allow the use of both IPv4 and IPv6 TCP networking.

2.1.2 Paths and Globbing

AppArmor explicitly distinguishes directory path names from file path names. Use a trailing / for any directory path that needs to be explicitly distinguished:

```
/some/random/example/* r
```
Allow read access to files in the /some/random/example directory.

```
/some/random/example/ r
```
Allow read access to the directory only.

```
/some/**/ r
```
Give read access to any directories below /some.

```
/some/random/example/** r
```
Give read access to files and directories under /some/random/example.

```
/some/random/example/**[^/] r
```
Give read access to files under /some/random/example. Explicitly exclude directories ([^/]).

Globbing (or regular expression matching) is when you modify the directory path using wild cards to include a group of files or subdirectories. File resources can be specified with a globbing syntax similar to that used by popular shells, such as csh, Bash, and zsh.

*	Substitutes for any number of any characters, except /.
	Example: An arbitrary number of file path elements.
**	Substitutes for any number of characters, including /.
	Example: An arbitrary number of path elements, including entire directories.
?	Substitutes for any single character, except /.
[abc]	Substitutes for the single character a, b, or c.
	Example: a rule that matches /home[01]/*/.plan allows a program to access .plan files for users in both /home0 and /home1.
[a-c]	Substitutes for the single character a, b, or c.
{ab,cd}	Expands to one rule to match ab and one rule to match cd.
	Example: a rule that matches /{usr,www}/pages/** grants access to Web pages in both /usr/pages and /www/pages.
[^a]	Substitutes for any character except a.

2.1.3 File Permission Access Modes

File permission access modes consist of combinations of the following eleven modes:

r	Read mode
w	Write mode (mutually exclusive to a)
a	Append mode (mutually exclusive to w)
k	File locking mode
px	Discrete profile execute mode
Px	Discrete profile execute mode—clean exec
ux	Unconstrained execute mode
Ux	Unconstrained execute mode—clean exec
ix	Inherit execute mode
m	Allow PROT_EXEC with mmap(2) calls
l	Link mode

Read Mode (r)
> Allows the program to have read access to the resource. Read access is required for shell scripts and other interpreted content and determines if an executing process can core dump or be attached to with ptrace(2) (ptrace(2) is used by utilities like strace(1), ltrace(1), and gdb(1)).

Write Mode (w)
> Allows the program to have write access to the resource. Files must have this permission if they are to be unlinked (removed).

Append Mode (a)
> Allows a program to write to the end of a file. In contrast to the w mode, the append mode does not include the ability to overwrite data, to rename, or to remove a file.

The append permission is typically used with applications who need to be able to write to log files, but which should not be able to manipulate any existing data in the log files. As the append permission is just a subset of the permissions associated with the write mode, the `w` and `a` permission flags cannot be used together and are mutually exclusive.

File Locking Mode (k)

The application can take file locks. Former versions of AppArmor allowed files to be locked if an application had access to them. By using a separate file locking mode, AppArmor makes sure locking is restricted only to those files which need file locking and tightens security as locking can be used in several denial of service attack scenarios.

Discrete Profile Execute Mode (px)

This mode requires that a discrete security profile is defined for a resource executed at an AppArmor domain transition. If there is no profile defined, the access is denied.

WARNING: Using the Discrete Profile Execute Mode

`px` does not scrub the environment of variables such as `LD_PRELOAD`. As a result, the calling domain may have an undue amount of influence over the called item.

Incompatible with `Ux`, `ux`, `Px`, and `ix`.

Discrete Profile Execute Mode (Px)—Clean Exec

`Px` allows the named program to run in `px` mode, but AppArmor invokes the Linux kernel's `unsafe_exec` routines to scrub the environment, similar to setuid programs. See `ld.so(8)` for some information about setuid and setgid environment scrubbing.

Incompatible with `Ux`, `ux`, `px`, and `ix`.

Unconstrained Execute Mode (ux)

Allows the program to execute the resource without any AppArmor profile applied to the executed resource. Requires listing execute mode as well.

This mode is useful when a confined program needs to be able to perform a privileged operation, such as rebooting the machine. By placing the privileged section in another executable and granting unconstrained execution rights, it is possible to

bypass the mandatory constraints imposed on all confined processes. For more information about what is constrained, see the `apparmor(7)` man page.

WARNING: Using Unconstrained Execute Mode (ux)

Use `ux` only in very special cases. It enables the designated child processes to be run without any AppArmor protection. `ux` does not scrub the environment of variables such as `LD_PRELOAD`. As a result, the calling domain may have an undue amount of influence over the called resource. Use this mode only if the child absolutely must be run unconfined and `LD_PRELOAD` must be used. Any profile using this mode provides negligible security. Use at your own risk.

This mode is incompatible with `Ux`, `px`, `Px`, and `ix`.

Unconstrained Execute Mode (Ux)—Clean Exec
> `Ux` allows the named program to run in `ux` mode, but AppArmor invokes the Linux kernel's `unsafe_exec` routines to scrub the environment, similar to setuid programs. See `ld.so(8)` for some information about setuid and setgid environment scrubbing.

WARNING: Using Unconstrained Execute Mode (Ux)

Use `Ux` only in very special cases. It enables the designated child processes to run without any AppArmor protection. Use this mode only if the child absolutely must be run unconfined. Use at your own risk.

Incompatible with `ux`, `px`, `Px`, and `ix`.

Inherit Execute Mode (ix)
> `ix` prevents the normal AppArmor domain transition on `execve(2)` when the profiled program executes the named program. Instead, the executed resource inherits the current profile.

> This mode is useful when a confined program needs to call another confined program without gaining the permissions of the target's profile or losing the permissions of the current profile. There is no version to scrub the environment because `ix` executions do not change privileges.

> Incompatible with `Ux`, `ux`, `Px`, and `px`. Implies `m`.

Allow Executable Mapping (m)

This mode allows a file to be mapped into memory using `mmap(2)`'s `PROT_EXEC` flag. This flag marks the pages executable. It is used on some architectures to provide nonexecutable data pages, which can complicate exploit attempts. AppArmor uses this mode to limit which files a well-behaved program (or all programs on architectures that enforce nonexecutable memory access controls) may use as libraries, to limit the effect of invalid `-L` flags given to `ld(1)` and `LD_PRELOAD`, `LD_LIBRARY_PATH`, given to `ld.so(8)`.

Link Mode

The link mode mediates access to hard links. When a link is created, the target file must have the same access permissions as the link created (with the exception that the destination does not need link access).

When choosing one of the Ux or Px file permission access modes, take into account that the following environment variables are removed from the environment before the child process inherits it. As a consequence, applications or processes relying on any of these variables do not work anymore if the profile applied to them carries Ux or Px flags:

- `GCONV_PATH`

- `GETCONF_DIR`

- `HOSTALIASES`

- `LD_AUDIT`

- `LD_DEBUG`

- `LD_DEBUG_OUTPUT`

- `LD_DYNAMIC_WEAK`

- `LD_LIBRARY_PATH`

- `LD_ORIGIN_PATH`

- `LD_PRELOAD`

- `LD_PROFILE`

- LD_SHOW_AUXV

- LD_USE_LOAD_BIAS

- LOCALDOMAIN

- LOCPATH

- MALLOC_TRACE

- NLSPATH

- RESOLV_HOST_CONF

- RES_OPTIONS

- TMPDIR

- TZDIR

2.2 `#include` Statements

`#include` statements are directives that pull in components of other Novell AppArmor profiles to simplify profiles. Include files fetch access permissions for programs. By using an include, you can give the program access to directory paths or files that are also required by other programs. Using includes can reduce the size of a profile.

By default, AppArmor adds `/etc/apparmor.d` to the path in the `#include` statement. AppArmor expects the include files to be located in `/etc/apparmor.d`. Unlike other profile statements (but similar to C programs), `#include` lines do not end with a comma.

To assist you in profiling your applications, Novell AppArmor provides two classes of `#include`s: abstractions and program chunks.

2.2.1 Abstractions

Abstractions are `#include`s that are grouped by common application tasks. These tasks include access to authentication mechanisms, access to name service routines, common graphics requirements, and system accounting. Files listed in these abstractions are specific to the named task. Programs that require one of these files usually require some of the other files listed in the abstraction file (depending on the local configuration as well as the specific requirements of the program). Find abstractions in `/etc/apparmor.d/abstractions`.

2.2.2 Program Chunks

The program-chunks directory (`/etc/apparmor.d/program-chunks`) contains some chunks of profiles that are specific to program suites and not generally useful outside of the suite, thus are never suggested for use in profiles by the profile wizards (aa-logprof and aa-genprof). Currently program chunks are only available for the postfix program suite.

2.3 Capability Entries (POSIX.1e)

Capabilities statements are simply the word `capability` followed by the name of the POSIX.1e capability as defined in the `capabilities(7)` man page.

2.4 Using the Local AppArmor Profile Repository

AppArmor ships a set of profiles enabled by default and created by the AppArmor developers and kept under the `/etc/apparmor.d`. In addition to these profiles, openSUSE ships profiles for individual applications together with the respective application. These profiles are not enabled by default and reside under another directory than the standard AppArmor profiles, `/etc/apparmor/profiles/extras`.

The AppArmor tools, both YaST and aa-genprof and aa-logprof, support the use of a local repository. Whenever you start to create a new profile from scratch and there al-

ready is one inactive profile in your local repository, you are asked whether you would like to use the existing inactive one from `/etc/apparmor/profiles/extras` and whether you want to base your efforts on it. If you decide to use this profile, it gets copied over to the directory of profiles enabled by default (`/etc/apparmor.d`) and loaded whenever AppArmor is started. Any further further adjustments will be done to the active profile under `/etc/apparmor.d`.

2.5 Using the External AppArmor Profile Repository

In addition to the profiles shipping with openSUSE, AppArmor supports the use of an external profile repository. This repository is maintained by Novell and allows you to download profiles generated by Novell and other AppArmor users as well as uploading your own. Find the profile repository at `http://apparmor.opensuse.org`.

NOTE: Using the AppArmor Profile Repository

When using the profile repository in your deployment, bear in mind that the profiles maintained in the repository are primarily targeted at profile developers and might probably need fine-tuning before they suit your particular needs. Please test the downloaded profiles extensively before deploying them to your live setup and adjust them if necessary.

The profile repository serves two main purposes:

1. Allow users to browse profiles created by other users and pull them from the server to use on their own systems.

2. Allow users to upload their profiles to be able to easily use them on different machines.

A valid login on the profile repository server is required for uploading profiles. Just downloading profiles from the server does not require a login.

2.5.1 Setting up Profile Repository Support

Once properly configured, both the YaST and the command line tools support the use of an external profile repository. The initial configuration takes place when you start the YaST Add Profile Wizard, the Update Profile Wizard, aa-genprof, or aa-logprof to create or update a profile that already exists on the repository server:

1 Determine whether to use or not to use the profile repository at all.

2 Enable the repository for profile downloads.

3 Once you have created or modified a profile, determine whether the tools should be able to upload your profile to the repository.

  If you chose to upload profiles to the repository, enter your credentials for the repository server.

The configuration of the repository is done by editing two configuration files, `/etc/apparmor/logprof.conf` and `/etc/apparmor/respository.conf`.

The `/etc/apparmor/logprof.conf` file contains a section called `[repository]`. `distro` determines the version of openSUSE used on your system for which the AppArmor tools should search profiles on the server. `url` holds the server URL and `preferred_user` tells the AppArmor tools to prefer profiles created by the `novell` user. Those profiles were created, tested and approved by members of the SUSE development team.

```
...
[repository]
  distro         = opensuse10.3
  url            = http://apparmor.opensuse.org/backend/api
  preferred_user = novell
...
```

The `/etc/apparmor/repository.conf` file is created during the configuration process with the AppArmor tools. It contains your authentication data and specifies which actions to enable with regards to the profile repository. If you opt for profile download and do not want to be able to upload your own profiles `enabled` is set to `yes` while `upload` is set to `no`.

```
[repository]
```

```
enabled = yes
upload = yes
user = tux
pass = XXXXX
```

Once initially configured through the AppArmor tools, the configuration can only be changed manually.

2.5.2 Downloading a Profile

While creating a profile from scratch or updating an existing profile by processing reject messages in the log, the AppArmor tools search the repository for a matching profile. If the search is successful, the profile or the list of profiles is displayed and you can view them and choose the one that best matches your setup. As soon as you have chosen a profile, it gets copied to the local machine (to the /etc/apparmor.d directory) and activated. Alternatively, you can choose to ignore the profile on the repository and create your own one from scratch.

2.5.3 Uploading Your own Profile

After a profile has been created or updated, the AppArmor tools that a profile also present in the repository has been changed or that a new one has been created. If your system is configured to upload profiles to the repository, you are prompted to provide a ChangeLog to document your changes before the changes are uploaded to the server. These changes are only synced to the repository, but not to the creator of the original profile.

2.6 Important Filenames and Directories

The following list contains the most important files and directories used by the AppArmor framework. If you intend to manage and troubleshoot your profiles manually, make sure that you know about these files and directories:

/sys/kernel/security/apparmor/profiles
    Virtualized file representing the currently loaded set of profiles.

`/etc/apparmor/`
Location of AppArmor configuration files.

`/etc/apparmor/profiles/extras/`
A local repository of profiles shipped with AppArmor, but not enabled by default.

`/etc/apparmor.d/`
Location of profiles, named with the convention of replacing the `/` in paths with `.` (not for the root `/`) so profiles are easier to manage. For example, the profile for the program `/usr/sbin/ntpd` is named `usr.sbin.ntpd`.

`/etc/apparmor.d/abstractions/`
Location of abstractions.

`/etc/apparmor.d/program-chunks/`
Location of program chunks.

`/proc/*/attr/current`
Check this file to review the confinement status of a process and the profile that is used to confine the process. The `ps auxZ` command retrieves this information automatically.

Building and Managing Profiles with YaST

3

YaST provides an easy way to build profiles and manage Novell® AppArmor. It provides two interfaces: a fully graphical one and a text-based one. The text-based interface consumes less resources and bandwidth, making it a better choice for remote administration or for times when a local graphical environment is inconvenient. Although the interfaces have differing appearances, they offer the same functionality in similar ways. Another alternative is to use AppArmor commands, which can control AppArmor from a terminal window or through remote connections. The command line tools are described in Chapter 4, *Building Profiles from the Command Line* (page 49).

Start YaST from the main menu and enter your `root` password when prompted for it. Alternatively, start YaST by opening a terminal window, logging in as `root`, and entering `yast2` for the graphical mode or `yast` for the text-based mode.

Figure 3.1   *YaST Controls for AppArmor*
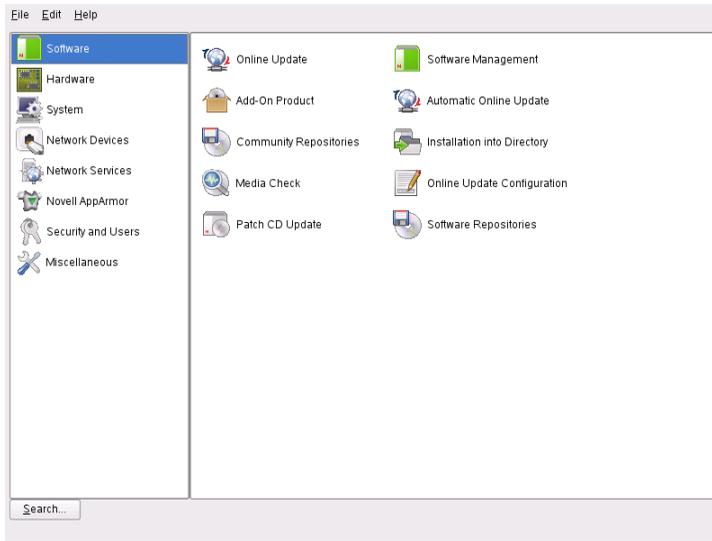
The right frame shows the AppArmor options:

Add Profile Wizard

For detailed steps, refer to Section 3.1, "Adding a Profile Using the Wizard" (page 29).

Manually Add Profile

Add a Novell AppArmor profile for an application on your system without the help of the wizard. For detailed steps, refer to Section 3.2, "Manually Adding a Profile" (page 37).

Edit Profile

Edits an existing Novell AppArmor profile on your system. For detailed steps, refer to Section 3.3, "Editing Profiles" (page 38).

Delete Profile

Deletes an existing Novell AppArmor profile from your system. For detailed steps, refer to Section 3.4, "Deleting a Profile" (page 43).

Update Profile Wizard

For detailed steps, refer to Section 3.5, "Updating Profiles from Log Entries" (page 44).

AppArmor Reports

For detailed steps, refer to Section 6.3, "Configuring Reports" (page 91).

AppArmor Control Panel

For detailed steps, refer to Section 3.6, "Managing Novell AppArmor and Security Event Status" (page 45).

3.1 Adding a Profile Using the Wizard

Add Profile Wizard is designed to set up Novell AppArmor profiles using the AppArmor profiling tools, aa-genprof (generate profile) and aa-logprof (update profiles from learning mode log file). For more information about these tools, refer to Section 4.6.3, "Summary of Profiling Tools" (page 56).

1 Stop the application before profiling it to ensure that application start-up is included in the profile. To do this, make sure that the application or daemon is not running.

For example, enter `rcPROGRAM stop` (or `/etc/init.d/PROGRAM stop`) in a terminal window while logged in as `root`, replacing *PROGRAM* with the name of the program to profile.

2 Start YaST and select *Novell AppArmor > Add Profile Wizard*.

This wizard will help you create a new AppArmor security
profile for an application, or you can use it to enhance
an existing profile by allowing AppArmor to learn new
application behavior.

Please enter the application name for which you would like
to create a profile, or selecte Browse to find the
application on your system.

Application to Profile

httpd2-prefork

Browse

Create   Abort

3 Enter the name of the application or browse to the location of the program.

4 Click *Create*. This runs an AppArmor tool named aa-autodep, which performs
a static analysis of the program to profile and loads an approximate profile
into the AppArmor module. For more information about aa-autodep, refer to
Section "aa-autodep—Creating Approximate Profiles" (page 56).

Depending on whether the profile you are about to create already exists either
in the local profile repository (see Section 2.4, "Using the Local AppArmor
Profile Repository" (page 22)) or in the external profile repository (see Sec-
tion 2.5, "Using the External AppArmor Profile Repository" (page 23)) or
whether it does not exist yet, proceed with one of the following options:

   • Determine whether you want to use or fine-tune an already existing profile
   from your local profile repository, as outlined in Step 5 (page 30).

   • Determine whether you want to use of fine-tune an already existing profile
   from the external profile repository, as outlined in Step 6 (page 31).

   • Create the profile from scratch and proceed with Step 7 (page 31) and
   beyond.

5 If the profile already exists in the local profile repository under /etc/
apparmor/profiles/extra, YaST informs you that there is an inactive

profile which you can either use as a base for your own efforts or which you can just accept as is.

Alternatively, you can choose not to use the local version at all and start creating the profile from scratch. In any case, proceed with Step 7 (page 31).

6 If the profile already exists in the external profile repository and this is the first time you tried to create a profile that already exists in the repository, configure your access to the server and determine how to use it:

6a Determine whether you want to enable access to the external repository or postpone this decision. In case you have selected *Enable Repository*, determine the access mode (download/upload) in a next step. In case you want to postpone the decision, select *Ask Me Later* and proceed directly to Step 7 (page 31).

6b Provide username and password for your account on the profile repository server and register at the server.

6c Select the profile to use and proceed to Step 7 (page 31).

7 Run the application to profile.

8 Perform as many of the application functions as possible so learning mode can log the files and directories to which the program requires access to function properly. Be sure to include restarting and stopping the program in the exercised functions. AppArmor needs to handle these events as well as any other program function.

9 Click *Scan system log for AppArmor events* to parse the learning mode log files. This generates a series of questions that you must answer to guide the wizard in generating the security profile.

If requests to add hats appear, proceed to Chapter 5, *Profiling Your Web Applications Using ChangeHat* (page 75).

The questions fall into two categories:

- A resource is requested by a profiled program that is not in the profile (see Figure 3.2, "Learning Mode Exception: Controlling Access to Specific Resources" (page 32)). Allow or deny access to a specific resource.

- A program is executed by the profiled program and the security domain transition has not been defined (see Figure 3.3, "Learning Mode Exception: Defining Execute Permissions for an Entry" (page 33)). Define execute permissions for an entry.

Each of these cases results in a series of questions that you must answer to add the resource to the profile or to add the program to the profile. For an example of each case, see Figure 3.2, "Learning Mode Exception: Controlling Access to Specific Resources" (page 32) and Figure 3.3, "Learning Mode Exception: Defining Execute Permissions for an Entry" (page 33). Subsequent steps describe your options in answering these questions.

NOTE: Varying Processing Options

Depending on the type of entry processed, the available options vary.

Figure 3.2   *Learning Mode Exception: Controlling Access to Specific Resources*

Figure 3.3   *Learning Mode Exception: Defining Execute Permissions for an Entry*

10 The *Add Profile Wizard* begins suggesting directory path entries that have been accessed by the application profiled (as seen in Figure 3.2, "Learning Mode Exception: Controlling Access to Specific Resources" (page 32)) or requires you to define execute permissions for entries (as seen in Figure 3.3, "Learning Mode Exception: Defining Execute Permissions for an Entry" (page 33)).

- For Figure 3.2: Learning Mode Exception: Controlling Access to Specific Resources: Select the option that satisfies the request for access, which could be a suggested include, a particular globbed version of the path, or the actual pathname. Depending on the situation, these options are available:

#include
   The section of a Novell AppArmor profile that refers to an include file. Include files give access permissions for programs. By using an include, you can give the program access to directory paths or files that are also required by other programs. Using includes can reduce the size of a profile. It is good practice to select includes when suggested.

Globbed Version

Accessed by clicking *Glob*. For information about globbing syntax, refer to Section 2.1.2, "Paths and Globbing" (page 15).

Actual Pathname

Literal path that the program needs to access to run properly.

After selecting a directory path, process it as an entry to the Novell App-Armor profile by clicking *Allow* or *Deny*. If you are not satisfied with the directory path entry as it is displayed, you can also *Glob* or *Edit* it.

The following options are available to process the learning mode entries and build the profile:

Allow

Grant the program access to the specified directory path entries. The *Add Profile Wizard* suggests file permission access. For more information about this, refer to Section 2.1.3, "File Permission Access Modes" (page 17).

Deny

Click *Deny* to prevent the program from accessing the specified paths.

Glob

Clicking this modifies the directory path (using wild cards) to include all files in the suggested directory. Double-clicking it grants access to all files and subdirectories beneath the one shown. For more information about globbing syntax, refer to Section 2.1.2, "Paths and Globbing" (page 15).

Glob w/Ext

Modify the original directory path while retaining the filename extension. A single click causes `/etc/apache2/file.ext` to become `/etc/apache2/*.ext`, adding the wild card (asterisk) in place of the filename. This allows the program to access all files in the suggested directories that end with the `.ext` extension. When you double-click it, access is granted to all files with the particular extension and subdirectories beneath the one shown.

Edit

Edit the highlighted line. The new edited line appears at the bottom of the list.

Abort

Abort aa-logprof, losing all rule changes entered so far and leaving all profiles unmodified.

Finish

Close aa-logprof, saving all rule changes entered so far and modifying all profiles.

Click *Allow* or *Deny* for each learning mode entry. These help build the Novell AppArmor profile.

NOTE

The number of learning mode entries corresponds to the complexity of the application.

- For Figure 3.3: Learning Mode Exception: Defining Execute Permissions for an Entry: From the following options, select the one that satisfies the request for access. For detailed information about the options available, refer to Section 2.1.3, "File Permission Access Modes" (page 17).

Inherit

Stay in the same security profile (parent's profile).

Profile

Require a separate profile to exist for the executed program. When selecting this option, also select whether AppArmor should sanitize the environment when switching profiles by removing certain environment variables that can modify the execution behavior of the child process. Unless these variables are absolutely required to properly execute the child process, always choose the more secure, sanitized option.

Unconfined

> Execute the program without a security profile. When prompted, have AppArmor sanitize the environment to avoid adding security risks by inheriting certain environment variables from the parent process.

WARNING: Risks of Running Unconfined

Unless absolutely necessary, do not run unconfined. Choosing the *Unconfined* option executes the new program without any protection from AppArmor.

Deny

> Click *Deny* to prevent the program from accessing the specified paths.

Abort

> Abort aa-logprof, losing all rule changes entered so far and leaving all profiles unmodified.

Finish

> Close aa-logprof, saving all rule changes entered so far and modifying all profiles.

11 Repeat the previous steps if you need to execute more functionality of the application.

When you are done, click *Finish*. Choose to apply your changes to the local profile set. If you have previously chosen to upload your profile to the external profile repository, provide a brief change log entry describing your work and upload the profile. If you had postponed the decision on whether to upload the profile or not, YaST asks you again and you can create an account the upload the profile now or not upload it at all.

As soon as you exit the *Profile Creation Wizard*, the profile is saved both locally and on the repository server, if you have chosen to upload it. The profile is then loaded into the AppArmor module.

3.2 Manually Adding a Profile

Novell AppArmor enables you to create a Novell AppArmor profile by manually adding entries into the profile. Select the application for which to create a profile then add entries.

1 Start YaST and select *Novell AppArmor > Manually Add Profile*.

2 Browse your system to find the application for which to create a profile.

3 When you find the application, select it and click *Open*. A basic, empty profile appears in the *AppArmor Profile Dialog* window.
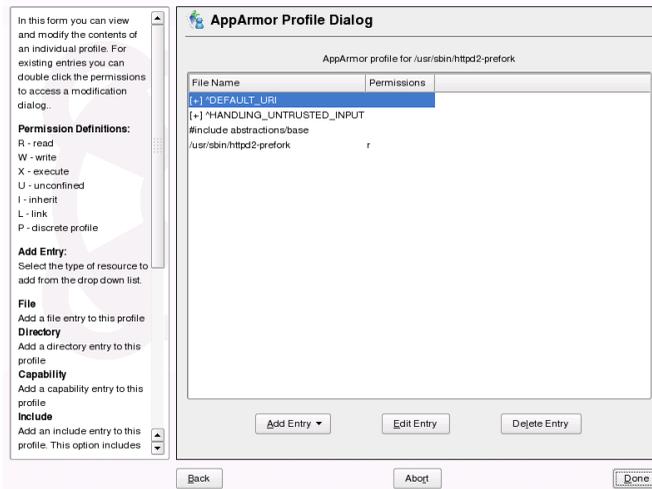
4 In *AppArmor Profile Dialog*, add, edit, or delete AppArmor profile entries by clicking the corresponding buttons and referring to Section 3.3.1, "Adding an Entry" (page 40), Section 3.3.2, "Editing an Entry" (page 42), or Section 3.3.3, "Deleting an Entry" (page 43).

5 When finished, click *Done*.

3.3 Editing Profiles

AppArmor enables you to edit Novell AppArmor profiles manually by adding, editing, or deleting entries. To edit a profile, proceed as follows:

1 Start YaST and select *Novell AppArmor > Edit Profile*.
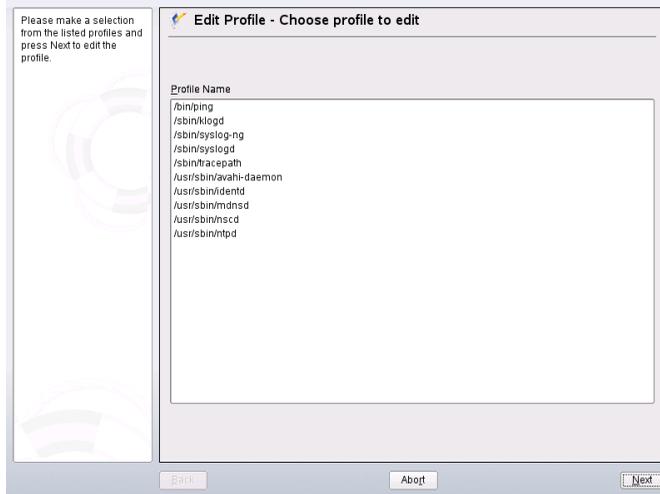
2 From the list of profiled applications, select the profile to edit.

3 Click *Next*. The *AppArmor Profile Dialog* window displays the profile.
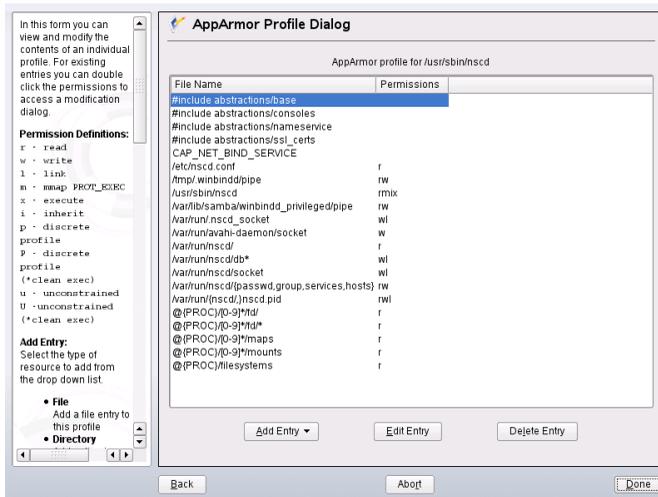
4 In the *AppArmor Profile Dialog* window, add, edit, or delete Novell AppArmor profile entries by clicking the corresponding buttons and referring to Section 3.3.1, "Adding an Entry" (page 40), Section 3.3.2, "Editing an Entry" (page 42), or Section 3.3.3, "Deleting an Entry" (page 43).

5 When you are finished, click *Done*.

6 In the pop-up that appears, click *Yes* to confirm your changes to the profile and reload the AppArmor profile set.

TIP: Syntax Checking in AppArmor

AppArmor contains a syntax check that notifies you of any syntax errors in profiles you are trying to process with the YaST AppArmor tools. If an error occurs, edit the profile manually as `root` and reload the profile set with `rcapparmor reload`.

3.3.1 Adding an Entry

The *Add Entry* option can be found in Section 3.2, "Manually Adding a Profile" (page 37) or Section 3.3, "Editing Profiles" (page 38). When you select *Add Entry*, a list shows the types of entries you can add to the Novell AppArmor profile.

From the list, select one of the following:

File

> In the pop-up window, specify the absolute path of a file, including the type of access permitted. When finished, click *OK*.

> You can use globbing if necessary. For globbing information, refer to Section 2.1.2, "Paths and Globbing" (page 15). For file access permission information, refer to Section 2.1.3, "File Permission Access Modes" (page 17).
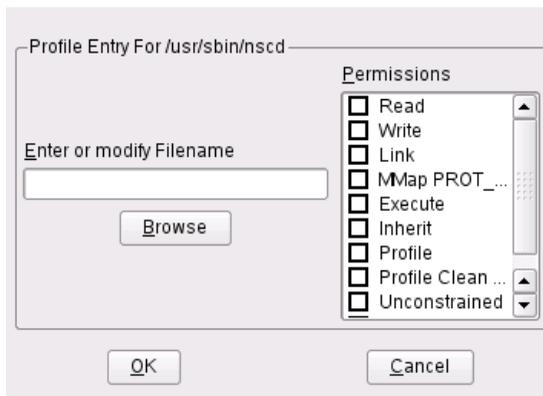
Directory

> In the pop-up window, specify the absolute path of a directory, including the type of access permitted. You can use globbing if necessary. When finished, click *OK*.

> For globbing information, refer to Section 2.1.2, "Paths and Globbing" (page 15). For file access permission information, refer to Section 2.1.3, "File Permission Access Modes" (page 17).
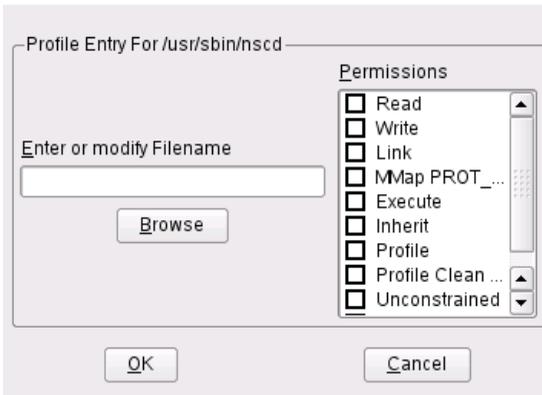
Capability

> In the pop-up window, select the appropriate capabilities. These are statements that enable each of the 32 POSIX.1e capabilities. Refer to Section 2.1, "Breaking a Novell AppArmor Profile into Its Parts" (page 12) for more information about capabilities. When finished making your selections, click *OK*.
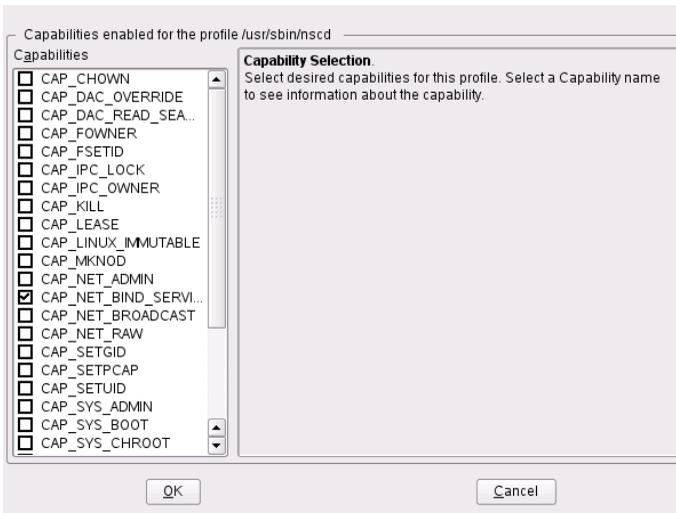
Include

> In the pop-up window, browse to the files to use as includes. Includes are directives that pull in components of other Novell AppArmor profiles to simplify profiles. For more information, refer to Section 2.2, "#include Statements" (page 21).
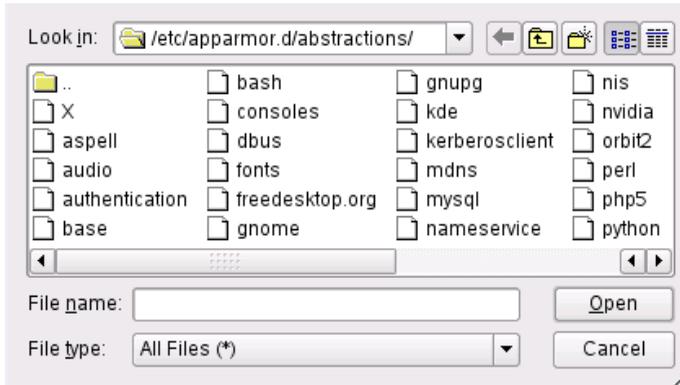
Hat

In the pop-up window, specify the name of the subprofile (*hat*) to add to your current profile and click *Create Hat*. For more information, refer to Chapter 5, *Profiling Your Web Applications Using ChangeHat* (page 75).

3.3.2 Editing an Entry

When you select *Edit Entry*, the file browser pop-up window opens. From here, edit the selected entry.

In the pop-up window, specify the absolute path of a file, including the type of access permitted. You can use globbing if necessary. When finished, click *OK*.
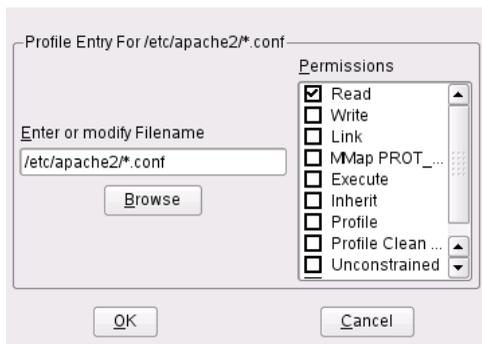
For globbing information, refer to Section 2.1.2, "Paths and Globbing" (page 15). For file access permission information, refer to Section 2.1.3, "File Permission Access Modes" (page 17).

3.3.3 Deleting an Entry

To delete an entry in a given profile, select *Delete Entry*. AppArmor removes the selected profile entry.

3.4 Deleting a Profile

AppArmor enables you to delete an AppArmor profile manually. Simply select the application for which to delete a profile then delete it as follows:

1 Start YaST and select *Novell AppArmor > Delete Profile*.

2 Select the profile to delete.

3 Click *Next*.

4 In the pop-up that opens, click *Yes* to delete the profile and reload the AppArmor profile set.

3.5 Updating Profiles from Log Entries

The Novell AppArmor profile wizard uses aa-logprof, the tool that scans log files and enables you to update profiles. aa-logprof tracks messages from the Novell AppArmor module that represent exceptions for all profiles running on your system. These exceptions represent the behavior of the profiled application that is outside of the profile definition for the program. You can add the new behavior to the relevant profile by selecting the suggested profile entry.

TIP: Support for the External Profile Repository

Similar to the *Add Profile Wizard*, the *Update Profile Wizard* also supports profile exchange with the external repository server. For background information on the use of the external AppArmor profile repository, refer to Section 2.5, "Using the External AppArmor Profile Repository" (page 23). For details on how to configure access and access mode to the server, check the procedure described under Section 3.1, "Adding a Profile Using the Wizard" (page 29).

---

1 Start YaST and select *Novell AppArmor > Update Profile Wizard*.

Running *Update Profile Wizard* (aa-logprof) parses the learning mode log files. This generates a series of questions that you must answer to guide aa-logprof to generate the security profile. The exact procedure is the same as with creating a

new profile. Refer to Step 9 (page 31) in Section 3.1, "Adding a Profile Using the Wizard" (page 29) for details.

2 When you are done, click *Finish*. In the following pop-up, click *Yes* to exit the *Add Profile Wizard*. The profile is saved and loaded into the Novell AppArmor module.

3.6 Managing Novell AppArmor and Security Event Status

You can change the status of AppArmor by enabling or disabling it. Enabling AppArmor protects your system from potential program exploitation. Disabling AppArmor, even if your profiles have been set up, removes protection from your system. You can determine how and when you are notified when system security events occur.

NOTE

For event notification to work, you must set up a mail server on your system that can send outgoing mail using the single mail transfer protocol (SMTP), such as postfix or exim.

To configure event notification or change the status of AppArmor, start YaST and select *Novell AppArmor > Novell AppArmor Control Panel*.
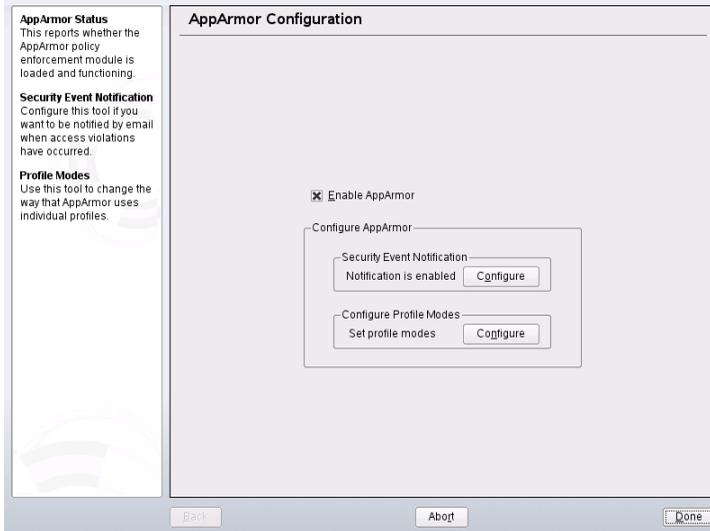
From the *AppArmor Configuration* screen, determine whether Novell AppArmor and security event notification are running by looking for a status message that reads *enabled* or configure the mode of individual profiles.

To change the status of Novell AppArmor, continue as described in Section 3.6.1, "Changing Novell AppArmor Status" (page 46). To change the mode of individual profiles, continue as described in Section 3.6.2, "Changing the Mode of Individual Profiles" (page 47). To configure security event notification, continue as described in Section 6.2, "Configuring Security Event Notification" (page 88).

3.6.1 Changing Novell AppArmor Status

When you change the status of AppArmor, set it to enabled or disabled. When AppArmor is enabled, it is installed, running, and enforcing the AppArmor security policies.

1 Start YaST and select *Novell AppArmor > AppArmor Control Panel*.

2 Enable AppArmor by checking *Enable AppArmor* or disable AppArmor by deselecting it.

3 Click *Done* in the *AppArmor Configuration* window.

4 Click *File > Quit* in the YaST Control Center.

3.6.2 Changing the Mode of Individual Profiles

AppArmor can apply profiles in two different modes. In *complain* or *learning* mode, violations of AppArmor profile rules, such as the profiled program accessing files not permitted by the profile, are detected. The violations are permitted, but also logged. This mode is convenient for developing profiles and is used by the AppArmor tools for generating profiles. Loading a profile in *enforce* mode enforces the policy defined in the profile and reports policy violation attempts to syslogd.

The *Profile Modes* dialog allows you to view and edit the mode of currently loaded AppArmor profiles. This feature is useful for determining the status of your system during profile development. During the course of systemic profiling (see), you can use this tool to adjust and monitor the scope of the profiles for which you are learning behavior.

To edit an application's profile mode, proceed as follows:

1 Start YaST and select *Novell AppArmor > AppArmor Control Panel*.

2 In the *Configure Profile Modes* section, select *Configure*.

3 Select the profile for which to change the mode.

4 Select *Toggle Mode* to set this profile to *complain* mode or to *enforce* mode.

5 Apply your settings and leave YaST with *Done*.

To change the mode of all profiles, use *Set All to Enforce* or *Set All to Complain*.

TIP: Listing the Profiles Available

By default, only active profiles are listed—any profile that has a matching application installed on your system. To set up a profile before installing the respective application, click *Show All Profiles* and select the profile to configure from the list that appears.

Building Profiles from the Command Line

4

Novell® AppArmor provides the ability to use a command line interface rather than a graphical interface to manage and configure your system security. Track the status of Novell AppArmor and create, delete, or modify AppArmor profiles using the AppArmor command line tools.

TIP: Background Information

Before starting to manage your profiles using the AppArmor command line tools, check out the general introduction to AppArmor given in Chapter 1, *Immunizing Programs* (page 1) and Chapter 2, *Profile Components and Syntax* (page 11).

4.1 Checking the AppArmor Module Status

An AppArmor module can be in any one of three states:

Unloaded
> The AppArmor module is not loaded into the kernel.

Running
> The AppArmor module is loaded into the kernel and is enforcing AppArmor program policies.

Stopped
    The AppArmor module is loaded into the kernel, but no policies are enforced.

Detect the state of the AppArmor module by inspecting `/sys/kernel/security/`
`apparmor/profiles`. If `cat`
`/sys/kernel/security/apparmor/profiles` reports a list of profiles,
AppArmor is running. If it is empty and returns nothing, AppArmor is stopped. If the
file does not exist, AppArmor is unloaded.

Manage AppArmor through the script `rcapparmor`, which can perform the following
operations:

`rcapparmor start`
    Behavior depends on the AppArmor module state. If it is unloaded, `start` loads
    the module and starts it, putting it in the running state. If it is stopped, `start`
    causes the module to rescan the AppArmor profiles usually found in `/etc/`
    `apparmor.d` and puts the module in the running state. If the module is already
    running, `start` reports a warning and takes no action.

`rcapparmor stop`
    Stops the AppArmor module if it is running by removing all profiles from kernel
    memory, effectively disabling all access controls, and putting the module into the
    stopped state. If the AppArmor module is unloaded or already stopped, `stop` tries
    to unload the profiles again, but nothing happens.

`rcapparmor restart`
    Causes the AppArmor module to rescan the profiles in `/etc/apparmor.d`
    without unconfining running processes. Freshly created profiles are enforced and
    recently deleted ones are removed from the `/etc/apparmor.d` directory.

`rcapparmor kill`
    Unconditionally removes the AppArmor module from the kernel. This is unsafe,
    because unloading modules from the Linux kernel is unsafe. This command is
    provided only for debugging and emergencies when the module might need to be
    removed.

WARNING

AppArmor is a powerful access control system and it is possible to lock yourself out of your own machine to the point where you must boot the machine from a rescue medium (such as the first medium of openSUSE) to regain control.

To prevent such a problem, always ensure that you have a running, unconfined, `root` login on the machine being configured when you restart the AppArmor module. If you damage your system to the point where logins are no longer possible (for example, by breaking the profile associated with the SSH daemon), you can repair the damage using your running `root` prompt then restart the AppArmor module.

4.2 Building AppArmor Profiles

The AppArmor module profile definitions are stored in the `/etc/apparmor.d` directory as plain text files. For a detailed description of the syntax of these files, refer to Chapter 2, *Profile Components and Syntax* (page 11).

All files in the `/etc/apparmor.d` directory are interpreted as profiles and are loaded as such. Renaming files in that directory is not an effective way of preventing profiles from being loaded. You must remove profiles from this directory to prevent them from being read and evaluated effectively.

You can use a text editor, such as vim, to access and make changes to these profiles. The following options contain detailed steps for building profiles:

Adding or Creating AppArmor Profiles
    Refer to Section 4.3, "Adding or Creating an AppArmor Profile" (page 52)

Editing AppArmor Profiles
    Refer to Section 4.4, "Editing an AppArmor Profile" (page 52)

Deleting AppArmor Profiles
    Refer to Section 4.5, "Deleting an AppArmor Profile" (page 52)

4.3 Adding or Creating an AppArmor Profile

To add or create an AppArmor profile for an application, you can use a systemic or stand-alone profiling method, depending on your needs. Learn more about these two approaches in Section 4.6, "Two Methods of Profiling" (page 53).

4.4 Editing an AppArmor Profile

The following steps describe the procedure for editing an AppArmor profile:

1 If you are not currently logged in as `root`, enter `su` in a terminal window.

2 Enter the `root` password when prompted.

3 Go to the profile directory with `cd /etc/apparmor.d/`.

4 Enter `ls` to view all profiles currently installed.

5 Open the profile to edit in a text editor, such as vim.

6 Make the necessary changes then save the profile.

7 Restart AppArmor by entering `rcapparmor restart` in a terminal window.

4.5 Deleting an AppArmor Profile

The following steps describe the procedure for deleting an AppArmor profile.

1 If you are not currently logged in as `root`, enter `su` in a terminal window.

2 Enter the `root` password when prompted.

3 Go to the AppArmor directory with `cd /etc/apparmor.d/`.

4 Enter `ls` to view all the AppArmor profiles that are currently installed.

5 Delete the profile with `rm` *profilename*.

6 Restart AppArmor by entering `rcapparmor restart` in a terminal window.

4.6 Two Methods of Profiling

Given the syntax for AppArmor profiles in Chapter 2, *Profile Components and Syntax* (page 11), you could create profiles without using the tools. However, the effort involved would be substantial. To avoid such a hassle, use the AppArmor tools to automate the creation and refinement of profiles.

There are two ways to approach AppArmor profile creation. Tools are available for both methods.

Stand-Alone Profiling
> A method suitable for profiling small applications that have a finite run time, such as user client applications like mail clients. For more information, refer to Section 4.6.1, "Stand-Alone Profiling" (page 54).

Systemic Profiling
> A method suitable for profiling large numbers of programs all at once and for profiling applications that may run for days, weeks, or continuously across reboots, such as network server applications like Web servers and mail servers. For more information, refer to Section 4.6.2, "Systemic Profiling" (page 54).

Automated profile development becomes more manageable with the AppArmor tools:

1 Decide which profiling method suits your needs.

2 Perform a static analysis. Run either aa-genprof or aa-autodep, depending on the profiling method chosen.

3 Enable dynamic learning. Activate learning mode for all profiled programs.

4.6.1 Stand-Alone Profiling

Stand-alone profile generation and improvement is managed by a program called aa-genprof. This method is easy because aa-genprof takes care of everything, but is limited because it requires aa-genprof to run for the entire duration of the test run of your program (you cannot reboot the machine while you are still developing your profile).

To use aa-genprof for the stand-alone method of profiling, refer to Section "aa-genprof—Generating Profiles" (page 59).

4.6.2 Systemic Profiling

This method is called *systemic profiling* because it updates all of the profiles on the system at once, rather than focusing on the one or few targeted by aa-genprof or stand-alone profiling. With systemic profiling, profile construction and improvement are somewhat less automated, but more flexible. This method is suitable for profiling long-running applications whose behavior continues after rebooting or a large number of programs all at once.

Build an AppArmor profile for a group of applications as follows:

1 Create profiles for the individual programs that make up your application.

Although this approach is systemic, AppArmor only monitors those programs with profiles and their children. To get AppArmor to consider a program, you must at least have aa-autodep create an approximate profile for it. To create this approximate profile, refer to Section "aa-autodep—Creating Approximate Profiles" (page 56).

2 Put relevant profiles into learning or complain mode.

Activate learning or complain mode for all profiled programs by entering `aa-complain /etc/apparmor.d/*` in a terminal window while logged in as `root`. This functionality is also available through the YaST Profile Mode module, described in Section 3.6.2, "Changing the Mode of Individual Profiles" (page 47).

When in learning mode, access requests are not blocked even if the profile dictates that they should be. This enables you to run through several tests (as shown in

Step 3 (page 55)) and learn the access needs of the program so it runs properly. With this information, you can decide how secure to make the profile.

Refer to Section "aa-complain—Entering Complain or Learning Mode" (page 57) for more detailed instructions for using learning or complain mode.

3 Exercise your application.

Run your application and exercise its functionality. How much to exercise the program is up to you, but you need the program to access each file representing its access needs. Because the execution is not being supervised by aa-genprof, this step can go on for days or weeks and can span complete system reboots.

4 Analyze the log.

In systemic profiling, run aa-logprof directly instead of letting aa-genprof run it (as in stand-alone profiling). The general form of aa-logprof is:

```
aa-logprof [ -d /path/to/profiles ] [ -f /path/to/logfile ]
```

Refer to Section "aa-logprof—Scanning the System Log" (page 67) for more information about using aa-logprof.

5 Repeat Step 3 (page 55) and Step 4 (page 55).

This generates optimum profiles. An iterative approach captures smaller data sets that can be trained and reloaded into the policy engine. Subsequent iterations generate fewer messages and run faster.

6 Edit the profiles.

You might want to review the profiles that have been generated. You can open and edit the profiles in /etc/apparmor.d/ using vim.

7 Return to enforce mode.

This is when the system goes back to enforcing the rules of the profiles, not just logging information. This can be done manually by removing the flags=(complain) text from the profiles or automatically by using the aa-enforce command, which works identically to the aa-complain command, except it sets the profiles to enforce mode. This functionality is also

available through the YaST Profile Mode module, described in Section 3.6.2, "Changing the Mode of Individual Profiles" (page 47).

To ensure that all profiles are taken out of complain mode and put into enforce mode, enter `aa-enforce /etc/apparmor.d/*`.

8 Rescan all profiles.

To have AppArmor rescan all of the profiles and change the enforcement mode in the kernel, enter `rcapparmor restart`.

4.6.3 Summary of Profiling Tools

All of the AppArmor profiling utilities are provided by the `apparmor-utils` RPM package and are stored in `/usr/sbin`. Each tool has a different purpose.

aa-autodep—Creating Approximate Profiles

This creates an approximate profile for the program or application selected. You can generate approximate profiles for binary executables and interpreted script programs. The resulting profile is called "approximate" because it does not necessarily contain all of the profile entries that the program needs to be properly confined by AppArmor. The minimum aa-autodep approximate profile has at least a base include directive, which contains basic profile entries needed by most programs. For certain types of programs, aa-autodep generates a more expanded profile. The profile is generated by recursively calling `ldd(1)` on the executables listed on the command line.

To generate an approximate profile, use the aa-autodep program. The program argument can be either the simple name of the program, which aa-autodep finds by searching your shell's path variable, or it can be a fully qualified path. The program itself can be of any type (ELF binary, shell script, Perl script, etc.). aa-autodep generates an approximate profile to improve through the dynamic profiling that follows.

The resulting approximate profile is written to the `/etc/apparmor.d` directory using the AppArmor profile naming convention of naming the profile after the absolute path of the program, replacing the forward slash (/) characters in the path with period (.) characters. The general form of aa-autodep is to enter the following in a terminal window when logged in as `root`:

```
aa-autodep [ -d /path/to/profiles ] [program1 program2...]
```

If you do not enter the program name or names, you are prompted for them.
/path/to/profiles overrides the default location of /etc/apparmor.d,
should you keep profiles in a location other than the default.

To begin profiling, you must create profiles for each main executable service that is
part of your application (anything that might start without being a child of another
program that already has a profile). Finding all such programs depends on the application
in question. Here are several strategies for finding such programs:

Directories

> If all the programs to profile are in one directory and there are no other programs
> in that directory, the simple command `aa-autodep`
> */path/to/your/programs/** creates basic profiles for all programs in that
> directory.

ps command

> You can run your application and use the standard Linux `ps` command to find all
> processes running. Then manually hunt down the location of these programs and
> run the `aa-autodep` for each one. If the programs are in your path, aa-autodep
> finds them for you. If they are not in your path, the standard Linux command `find`
> might be helpful in finding your programs. Execute `find / -name`
> '*my_application*' `-print` to determine an application's path
> (*my_application* being an example application). You may use wild cards if
> appropriate.

aa-complain—Entering Complain or Learning Mode

The complain or learning mode tool (aa-complain) detects violations of AppArmor
profile rules, such as the profiled program accessing files not permitted by the profile.
The violations are permitted, but also logged. To improve the profile, turn complain
mode on, run the program through a suite of tests to generate log events that characterize
the program's access needs, then postprocess the log with the AppArmor tools to
transform log events into improved profiles.

Manually activating complain mode (using the command line) adds a flag to the top of
the profile so that /bin/foo becomes /bin/foo flags=(complain). To use
complain mode, open a terminal window and enter one of the following lines as root:

- If the example program (*program1*) is in your path, use:

```
aa-complain [program1 program2 ...]
```

- If the program is not in your path, specify the entire path as follows:

```
aa-complain /sbin/program1
```

- If the profiles are not in /etc/apparmor.d, use the following to override the default location:

```
aa-complain /path/to/profiles/ program1
```

- Specify the profile for *program1* as follows:

```
aa-complain /etc/apparmor.d/sbin.program1
```

Each of the above commands activates the complain mode for the profiles or programs listed. If the program name does not include its entire path, aa-complain searches $PATH for the program. For instance, aa-complain /usr/sbin/* finds profiles associated with all of the programs in /usr/sbin and puts them into complain mode. aa-complain /etc/apparmor.d/* puts all of the profiles in /etc/apparmor .d into complain mode.

TIP: Toggling Profile Mode with YaST

YaST offers a graphical front-end for toggling complain and enforce mode. See for information.

---

aa-enforce—Entering Enforce Mode

The enforce mode detects violations of AppArmor profile rules, such as the profiled program accessing files not permitted by the profile. The violations are logged and not permitted. The default is for enforce mode to be enabled. To log the violations only, but still permit them, use complain mode. Enforce toggles with complain mode.

Manually activating enforce mode (using the command line) adds a flag to the top of the profile so that /bin/foo becomes /bin/foo flags=(enforce). To use enforce mode, open a terminal window and enter one of the following lines as root.

- If the example program (*program1*) is in your path, use:

  ```
  aa-enforce [program1 program2 ...]
  ```

- If the program is not in your path, specify the entire path, as follows:

  ```
  aa-enforce /sbin/program1
  ```

- If the profiles are not in */etc/apparmor.d*, use the following to override the default location:

  ```
  aa-enforce /path/to/profiles/program1
  ```

- Specify the profile for *program1* as follows:

  ```
  aa-enforce /etc/apparmor.d/sbin.program1
  ```

Each of the above commands activates the enforce mode for the profiles and programs listed.

If you do not enter the program or profile names, you are prompted to enter one. */path/to/profiles* overrides the default location of /etc/apparmor.d.

The argument can be either a list of programs or a list of profiles. If the program name does not include its entire path, aa-enforce searches $PATH for the program.

TIP: Toggling Profile Mode with YaST

YaST offers a graphical front-end for toggling complain and enforce mode. See Section 3.6.2, "Changing the Mode of Individual Profiles" (page 47) for information.

---

aa-genprof—Generating Profiles

aa-genprof is AppArmor's profile generating utility. It runs aa-autodep on the specified program, creating an approximate profile (if a profile does not already exist for it), sets it to complain mode, reloads it into AppArmor, marks the log, and prompts the user to execute the program and exercise its functionality. Its syntax is as follows:

```
aa-genprof [ -d /path/to/profiles ]  program
```

To create a profile for the the Apache Web server program httpd2-prefork, do the following as `root`:

1 Enter `rcapache2 stop`.

2 Next, enter `aa-genprof httpd2-prefork`.

Now aa-genprof does the following:

1. Resolves the full path of httpd2-prefork using your shell's path variables. You can also specify a full path. On openSUSE, the default full path is `/usr/sbin/httpd2-prefork`.

2. Checks to see if there is an existing profile for httpd2-prefork. If there is one, it updates it. If not, it creates one using the aa-autodep as described in Section 4.6.3, "Summary of Profiling Tools" (page 56).

3. Puts the profile for this program into learning or complain mode so that profile violations are logged but are permitted to proceed. A log event looks like this (see `/var/log/audit/audit.log`):

```
type=APPARMOR_ALLOWED msg=audit(1189682639.184:20816):
operation="file_mmap" requested_mask="r" denied_mask="r"
name="/srv/www/htdocs/index.html" pid=27471
profile="null-complain-profile"
```

If you are not running the audit daemon, the AppArmor events are logged to `/var/log/messages`:

```
Sep 13 13:20:30 K23 kernel: audit(1189682430.672:20810):
operation="file_mmap" requested_mask="r" denied_mask="r"
name="/srv/www/htdocs/phpsysinfo/templates/bulix/form.tpl" pid=30405
 profile="/usr/sbin/httpd2-prefork///phpsysinfo/"
```

They also can be viewed using the `dmesg` command:

```
audit(1189682430.672:20810): operation="file_mmap" requested_mask="r"
 denied_mask="r"
name="/srv/www/htdocs/phpsysinfo/templates/bulix/form.tpl" pid=30405
 profile="/usr/sbin/httpd2-prefork///phpsysinfo/"
```

4. Marks the log with a beginning marker of log events to consider. For example:

```
Sep 13 17:48:52 figwit root: GenProf:
e2ff78636296f16d0b5301209a04430d
```

3 When prompted by the tool, run the application to profile in another terminal window and perform as many of the application functions as possible. Thus, the learning mode can log the files and directories to which the program requires access in order to function properly. For example, in a new terminal window, enter `rcapache2 start`.

4 Select from the following options that are available in the aa-logprof terminal window after you have executed the program function:

- `S` runs aa-logprof on the system log from where it was marked when aa-genprof was started and reloads the profile. If system events exist in the log, AppArmor parses the learning mode log files. This generates a series of questions that you must answer to guide aa-genprof in generating the security profile.

- `F` exits the tool and returns to the main menu.

NOTE

If requests to add hats appear, proceed to Chapter 5, *Profiling Your Web Applications Using ChangeHat* (page 75).

---

5 Answer two types of questions:

- A resource is requested by a profiled program that is not in the profile (see Example 4.1, "Learning Mode Exception: Controlling Access to Specific Resources" (page 62)).

- A program is executed by the profiled program and the security domain transition has not been defined (see Example 4.2, "Learning Mode Exception: Defining Execute Permissions for an Entry" (page 63)).

Each of these categories results in a series of questions that you must answer to add the resource or program to the profile. Example 4.1, "Learning Mode Exception: Controlling Access to Specific Resources" (page 62) and Example 4.2, "Learning Mode Exception: Defining Execute Permissions for an Entry"

(page 63) provide examples of each one. Subsequent steps describe your options in answering these questions.

- Dealing with execute accesses is complex. You must decide how to proceed with this entry regarding which execute permission type to grant to this entry:

Example 4.1 *Learning Mode Exception: Controlling Access to Specific Resources*

```
Reading log entries from /var/log/audit/audit.log.
Updating AppArmor profiles in /etc/apparmor.d.

Profile:  /usr/sbin/xinetd
Program:  xinetd
Execute:  /usr/lib/cups/daemon/cups-lpd
Severity: unknown

[(I)nherit] / (P)rofile / (U)nconfined / (D)eny / Abo(r)t / (F)inish
```

Inherit (ix)
> The child inherits the parent's profile, running with the same access controls as the parent. This mode is useful when a confined program needs to call another confined program without gaining the permissions of the target's profile or losing the permissions of the current profile. This mode is often used when the child program is a *helper application*, such as the /usr/bin/mail client using less as a pager or the Mozilla* Web browser using Adobe Acrobat* to display PDF files.

Profile (px)
> The child runs using its own profile, which must be loaded into the kernel. If the profile is not present, attempts to execute the child fail with permission denied. This is most useful if the parent program is invoking a global service, such as DNS lookups or sending mail with your system's MTA.
>
> Choose the *profile with clean exec* (Px) option to scrub the environment of environment variables that could modify execution behavior when passed to the child process.

Unconfined (ux)
> The child runs completely unconfined without any AppArmor profile applied to the executed resource.

Choose the *unconfined with clean exec* (Ux) option to scrub the environment of environment variables that could modify execution behavior when passed to the child process. This option introduces a security vulnerability that could be used to exploit AppArmor. Only use it as a last resort.

mmap (m)

This permission denotes that the program running under the profile can access the resource using the mmap system call with the flag PROT_EXEC. This means that the data mapped in it can be executed. You are prompted to include this permission if it is requested during a profiling run.

Deny

Prevents the program from accessing the specified directory path entries. AppArmor then continues to the next event.

Abort

Aborts aa-logprof, losing all rule changes entered so far and leaving all profiles unmodified.

Finish

Closes aa-logprof, saving all rule changes entered so far and modifying all profiles.

• Example 4.2, "Learning Mode Exception: Defining Execute Permissions for an Entry" (page 63) shows AppArmor suggesting directory path entries that have been accessed by the application being profiled. It might also require you to define execute permissions for entries.

Example 4.2   *Learning Mode Exception: Defining Execute Permissions for an Entry*

```
Adding /bin/ps ix to profile.

Profile:  /usr/sbin/xinetd
Path:     /etc/hosts.allow
New Mode: r

 [1 - /etc/hosts.allow]

[(A)llow] / (D)eny / (N)ew / (G)lob / Glob w/(E)xt / Abo(r)t /
(F)inish
```

AppArmor provides one or more paths or includes. By entering the option number, select the desired options then proceed to the next step.

NOTE

All of these options are not always presented in the AppArmor menu.

`#include`
> This is the section of an AppArmor profile that refers to an include file, which procures access permissions for programs. By using an include, you can give the program access to directory paths or files that are also required by other programs. Using includes can reduce the size of a profile. It is good practice to select includes when suggested.

Globbed Version
> This is accessed by selecting *Glob* as described in the next step. For information about globbing syntax, refer to .

Actual Path
> This is the literal path to which the program needs access so that it can run properly.

After you select the path or include, process it as an entry into the AppArmor profile by selecting *Allow* or *Deny*. If you are not satisfied with the directory path entry as it is displayed, you can also *Glob* it.

The following options are available to process the learning mode entries and build the profile:

Select Enter
> Allows access to the selected directory path.

Allow
> Allows access to the specified directory path entries. AppArmor suggests file permission access. For more information, refer to .

Deny
> Prevents the program from accessing the specified directory path entries. AppArmor then continues to the next event.

New

> Prompts you to enter your own rule for this event, allowing you to specify a regular expression. If the expression does not actually satisfy the event that prompted the question in the first place, AppArmor asks for confirmation and lets you reenter the expression.

Glob

> Select a specific path or create a general rule using wild cards that match a broader set of paths. To select any of the offered paths, enter the number that is printed in front of the path then decide how to proceed with the selected item.
>
> For more information about globbing syntax, refer to Section 2.1.2, "Paths and Globbing" (page 15).

Glob w/Ext

> This modifies the original directory path while retaining the filename extension. For example, `/etc/apache2/file.ext` becomes `/etc/apache2/*.ext`, adding the wild card (asterisk) in place of the filename. This allows the program to access all files in the suggested directory that end with the `.ext` extension.

Abort

> Aborts aa-logprof, losing all rule changes entered so far and leaving all profiles unmodified.

Finish

> Closes aa-logprof, saving all rule changes entered so far and modifying all profiles.

6 To view and edit your profile using vim, enter `vim /etc/apparmor.d/profilename` in a terminal window.

7 Restart AppArmor and reload the profile set including the newly created one using the `rcapparmor restart` command.

Like the graphical front-end for building AppArmor profiles, the YaST Add Profile Wizard, aa-genprof also supports the use of the local profile repository under `/etc/apparmor/profiles/extras` and the remote AppArmor profile repository.

To use a profile from the local repository, proceed as follows:

1 Start aa-genprof as described above.

If aa-genprof finds an inactive local profile, the following lines appear on your terminal window:

```
Profile: /usr/bin/opera

 [1 - Inactive local profile for /usr/bin/opera]

[(V)iew Profile] / (U)se Profile / (C)reate New Profile / Abo(r)t /
(F)inish
```

2 If you want to just use this profile, hit U (*Use Profile*) and follow the profile generation procedure outlined above.

If you want to examine the profile before activating it, hit V (*View Profile*).

If you want to ignore the existing profile, hit C (*Create New Profile*) and follow the profile generation procedure outlined above to create the profile from scratch.

3 Leave aa-genprof by hitting F (*Finish*) when you are done and save your changes.

To use the remote AppArmor profile repository with aa-genprof, proceed as follows:

1 Start aa-genprof as described above.

If aa-genprof detects a suitable profile on the repository server, the following lines appear on your terminal window:

```
Repository: http://apparmor.opensuse.org/backend/api

Would you like to enable access to the profile repository?

(E)nable Repository / (D)isable Repository / Ask Me (L)ater
```

2 Hit E (*Enable Repository*) to enable the repository.

3 Determine whether you want to aa-genprof to upload any profiles to the repository server:

```
Would you like to upload newly created and changed profiles to
     the profile repository?
```

```
(Y)es / (N)o / Ask Me (L)ater
```

Hit Y (*Yes*), if you want to enable profile upload or select N (*No*), if you want aa-genprof to just pull profiles from the repository, but not to upload any.

4 Create a new user on the profile repository server to be able to upload profiles. Provide username and password.

5 Determine whether you want to use the profile downloaded from the server or whether you would just like to review it:

```
Profile: /usr/bin/opera

 [1 - novell]

[(V)iew Profile] / (U)se Profile / (C)reate New Profile / Abo(r)t /
(F)inish
```

If you want to just use this profile, hit U (*Use Profile*) and follow the profile generation procedure outlined above.

If you want to examine the profile before activating it, hit V (*View Profile*).

If you want to ignore the existing profile, hit C (*Create New Profile*) and follow the profile generation procedure outlined above to create the profile from scratch.

6 Leave aa-genprof by hitting F (*Finish*) when you are done and save the profile.

If you opted for uploading your profile, provide a short change log and push it to the repository.

aa-logprof—Scanning the System Log

aa-logprof is an interactive tool used to review the learning or complain mode output found in the log entries in `/var/log/audit/audit.log` or `/var/log/messages` (if auditd is not running) and generate new entries in AppArmor security profiles.

When you run aa-logprof, it begins to scan the log files produced in learning or complain mode and, if there are new security events that are not covered by the existing profile set, it gives suggestions for modifying the profile. The learning or complain mode traces

program behavior and enters it in the log. aa-logprof uses this information to observe program behavior.

If a confined program forks and executes another program, aa-logprof sees this and asks the user which execution mode should be used when launching the child process. The execution modes *ix*, *px*, *Px*, *ux*, and *Ux* are options for starting the child process. If a separate profile exists for the child process, the default selection is *px*. If one does not exist, the profile defaults to *ix*. Child processes with separate profiles have aa-autodep run on them and are loaded into AppArmor, if it is running.

When aa-logprof exits, profiles are updated with the changes. If the AppArmor module is running, the updated profiles are reloaded and, if any processes that generated security events are still running in the null-complain-profile, those processes are set to run under their proper profiles.

TIP: Support for the External Profile Repository

Similar to the aa-genprof, aa-logprof also supports profile exchange with the external repository server. For background information on the use of the external AppArmor profile repository, refer to Section 2.5, "Using the External AppArmor Profile Repository" (page 23). For details on how to configure access and access mode to the server, check the procedure described under Section "aa-genprof—Generating Profiles" (page 59).

To run aa-logprof, enter `aa-logprof` into a terminal window while logged in as `root`. The following options can be used for aa-logprof:

`aa-logprof -d` */path/to/profile/directory/*
   Specifies the full path to the location of the profiles if the profiles are not located in the standard directory, `/etc/apparmor.d/`.

`aa-logprof -f` */path/to/logfile/*
   Specifies the full path to the location of the log file if the log file is not located in the default directory, `/var/log/audit/audit.log` or `/var/log/messages` (if auditd is not running).

`aa-logprof -m "string marker in logfile"`
   Marks the starting point for aa-logprof to look in the system log. aa-logprof ignores all events in the system log before the specified mark. If the mark contains spaces, it must be surrounded by quotes to work correctly. For example:

```
aa-logprof -m"17:04:21"
```

or

```
logprof -m e2ff78636296f16d0b5301209a04430d
```

aa-logprof scans the log, asking you how to handle each logged event. Each question presents a numbered list of AppArmor rules that can be added by pressing the number of the item on the list.

By default, aa-logprof looks for profiles in `/etc/apparmor.d/` and scans the log in `/var/log/messages`. In many cases, running `aa-logprof` as `root` is enough to create the profile.

However, there might be times when you need to search archived log files, such as if the program exercise period exceeds the log rotation window (when the log file is archived and a new log file is started). If this is the case, you can enter `zcat -f \`ls -1tr /var/log/messages*\` | aa-logprof -f -`.

aa-logprof Example 1

The following is an example of how aa-logprof addresses httpd2-prefork accessing the file `/etc/group`. `[]` indicates the default option.

In this example, the access to `/etc/group` is part of httpd2-prefork accessing name services. The appropriate response is `1`, which includes a predefined set of AppArmor rules. Selecting `1` to `#include` the name service package resolves all of the future questions pertaining to DNS lookups and also makes the profile less brittle in that any changes to DNS configuration and the associated name service profile package can be made just once, rather than needing to revise many profiles.

```
Profile:  /usr/sbin/httpd2-prefork
Path:     /etc/group
New Mode: r

[1 - #include <abstractions/nameservice>]
 2 - /etc/group
[(A)llow] / (D)eny / (N)ew / (G)lob / Glob w/(E)xt / Abo(r)t / (F)inish
```

Select one of the following responses:

Select Enter

> Triggers the default action, which is, in this example, allowing access to the specified directory path entry.

Allow

> Allows access to the specified directory path entries. AppArmor suggests file permission access. For more information about this, refer to Section 2.1.3, "File Permission Access Modes" (page 17).

Deny

> Prevents the program from accessing the specified directory path entries. AppArmor then continues to the next event.

New

> Prompts you to enter your own rule for this event, allowing you to specify whatever form of regular expression you want. If the expression entered does not actually satisfy the event that prompted the question in the first place, AppArmor asks for confirmation and lets you reenter the expression.

Glob

> Select either a specific path or create a general rule using wild cards that matches on a broader set of paths. To select any of the offered paths, enter the number that is printed in front of the paths then decide how to proceed with the selected item.

> For more information about globbing syntax, refer to Section 2.1.2, "Paths and Globbing" (page 15).

Glob w/Ext

> This modifies the original directory path while retaining the filename extension. For example, `/etc/apache2/file.ext` becomes `/etc/apache2/*.ext`, adding the wild card (asterisk) in place of the filename. This allows the program to access all files in the suggested directory that end with the `.ext` extension.

Abort

> Aborts aa-logprof, losing all rule changes entered so far and leaving all profiles unmodified.

Finish

> Closes aa-logprof, saving all rule changes entered so far and modifying all profiles.

aa-logprof Example 2

For example, when profiling vsftpd, see this question:

```
Profile:  /usr/sbin/vsftpd
Path:     /y2k.jpg
New Mode: r

[1 - /y2k.jpg]

(A)llow / [(D)eny] / (N)ew / (G)lob / Glob w/(E)xt / Abo(r)t / (F)inish
```

Several items of interest appear in this question. First, note that vsftpd is asking for a path entry at the top of the tree, even though vsftpd on openSUSE serves FTP files from `/srv/ftp` by default. This is because httpd2-prefork uses chroot and, for the portion of the code inside the chroot jail, AppArmor sees file accesses in terms of the chroot environment rather than the global absolute path.

The second item of interest is that you might want to grant FTP read access to all JPEG files in the directory, so you could use *Glob w/Ext* and use the suggested path of `/*.jpg`. Doing so collapses all previous rules granting access to individual `.jpg` files and forestalls any future questions pertaining to access to `.jpg` files.

Finally, you might want to grant more general access to FTP files. If you select *Glob* in the last entry, aa-logprof replaces the suggested path of `/y2k.jpg` with `/*`. Alternatively, you might want to grant even more access to the entire directory tree, in which case you could use the *New* path option and enter `/**.jpg` (which would grant access to all `.jpg` files in the entire directory tree) or `/**` (which would grant access to all files in the directory tree).

These items deal with read accesses. Write accesses are similar, except that it is good policy to be more conservative in your use of regular expressions for write accesses. Dealing with execute accesses is more complex. Find an example in Example 4.1, "Learning Mode Exception: Controlling Access to Specific Resources" (page 62).

In the following example, the `/usr/bin/mail` mail client is being profiled and aa-logprof has discovered that `/usr/bin/mail` executes `/usr/bin/less` as a helper application to "page" long mail messages. Consequently, it presents this prompt:

```
/usr/bin/nail -> /usr/bin/less
(I)nherit / (P)rofile / (U)nconfined / (D)eny
```

> **TIP**
>
> The actual executable file for `/usr/bin/mail` turns out to be `/usr/bin/nail`, which is not a typographical error.

The program `/usr/bin/less` appears to be a simple one for scrolling through text that is more than one screen long and that is in fact what `/usr/bin/mail` is using it for. However, less is actually a large and powerful program that makes use of many other helper applications, such as tar and rpm.

> **TIP**
>
> Run `less` on a tar file or an RPM file and it shows you the inventory of these containers.

You do not want to run `rpm` automatically when reading mail messages (that leads directly to a Microsoft* Outlook–style virus attack, because rpm has the power to install and modify system programs), so, in this case, the best choice is to use *Inherit*. This results in the less program executed from this context running under the profile for `/usr/bin/mail`. This has two consequences:

- You need to add all of the basic file accesses for `/usr/bin/less` to the profile for `/usr/bin/mail`.

- You can avoid adding the helper applications, such as tar and rpm, to the `/usr/bin/mail` profile so that when `/usr/bin/mail` runs `/usr/bin/less` in this context, the less program is far less dangerous than it would be without App-Armor protection.

In other circumstances, you might instead want to use the *Profile* option. This has two effects on aa-logprof:

- The rule written into the profile uses px, which forces the transition to the child's own profile.

- aa-logprof constructs a profile for the child and starts building it, in the same way that it built the parent profile, by assigning events for the child process to the child's profile and asking the aa-logprof user questions.

If a confined program forks and executes another program, aa-logprof sees this and asks the user which execution mode should be used when launching the child process. The execution modes of inherit, profile, unconfined or an option to deny the execution are presented.

If a separate profile exists for the child process, the default selection is profile. If a profile does not exist, the default is inherit. The inherit option, or `ix`, is described in

The profile option indicates that the child program should run in its own profile—a secondary question asks whether to sanitize the environment that the child program inherits from the parent. If you choose to sanitize the environment, this places the execution modifier `Px` in your AppArmor profile. If you select not to sanitize, `px` is placed in the profile and no environment sanitizing occurs. The default for the execution mode is `px` if you select profile execution mode.

The unconfined execution mode is not recommended and should only be used in cases where there is no other option to generate a profile for a program reliably. Selecting unconfined opens a warning dialog asking for confirmation of the choice. If you are sure and choose *Yes*, a second dialog ask whether to sanitize the environment. Choosing *Yes* uses the execution mode `Ux` in your profile. Choosing *No* uses the execution mode `ux` for your profile. The default value selected is `Ux` for unconfined execution mode.

IMPORTANT: Running Unconfined

Choosing `ux` is very dangerous and provides no enforcement of policy from a security perspective of resulting execution behavior of the child program.

---

aa-unconfined—Identifying Unprotected Processes

The `aa-unconfined` command examines open network ports on your system, compares that to the set of profiles loaded on your system, and reports network services that do not have AppArmor profiles. It requires `root` privileges and that it not be confined by an AppArmor profile.

aa-unconfined must be run as `root` to retrieve the process executable link from the `/proc` file system. This program is susceptible to the following race conditions:

• An unlinked executable is mishandled

• A process that dies between `netstat(8)` and further checks is mishandled

NOTE

This program lists processes using TCP and UDP only. In short, this program is unsuitable for forensics use and is provided only as an aid to profiling all network-accessible processes in the lab.

Profiling Your Web Applications Using ChangeHat

5

A Novell® AppArmor profile represents the security policy for an individual program instance or process. It applies to an executable program, but if a portion of the program needs different access permissions than other portions, the program can "change hats" to use a different security context, distinctive from the access of the main program. This is known as a *hat* or *subprofile*.

ChangeHat enables programs to change to or from a *hat* within a Novell AppArmor profile. It enables you to define security at a finer level than the process. This feature requires that each application be made "ChangeHat aware" meaning that it is modified to make a request to the Novell AppArmor module to switch security domains at arbitrary times during the application execution. Two examples for ChangeHat-aware applications are the Apache Web server and Tomcat.

A profile can have an arbitrary number of subprofiles, but there are only two levels: a subprofile cannot have further sub-subprofiles. A subprofile is written as a separate profile and named as the containing profile followed by the subprofile name, separated by a ^. Subprofiles must be stored in the same file as the parent profile.

Note that the security of hats is considerably weaker than that of full profiles. That is to say, if an attacker can find just the right kind of bug in a program, they may be able to escape from a hat into the containing profile. This is because the security of hats is determined by a secret key handled by the containing process, and the code running in the hat must not have access to the key. Thus change_hat is most useful in conjunction with application servers, where a language interpreter (such as PERL, PHP, or Java) is isolating pieces of code such that they do not have direct access to the memory of the containing process.

The rest of this chapter describes using change_hat in conjunction with Apache, to contain web server components run using mod_perl and mod_php. Similar approaches can be used with any application server by providing an application module similar to the mod_apparmor described next in Section 5.2.2, "Location and Directory Directives" (page 84).

NOTE: For More Information

For more information, see the change_hat man page.

5.1 Apache ChangeHat

Novell AppArmor provides a mod_apparmor module (package apache2-mod_apparmor) for the Apache program. This module makes the Apache Web server ChangeHat aware. Install it along with Apache.

When Apache is ChangeHat aware, it checks for the following customized Novell AppArmor security profiles in the order given for every URI request that it receives.

- URI-specific hat (for example, ^phpsysinfo/templates/classic/images/bar_left.gif)

- DEFAULT_URI

- HANDLING_UNTRUSTED_INPUT

NOTE: Apache Configuration

If you install apache2-mod_apparmor without Novell AppArmor, make sure that the Apache load module has a command in the configuration file that loads the mod_apparmor module by adding apparmor to the list of modules to load in /etc/sysconfig/apache2:

```
APACHE_MODULES="apparmor"
```

Alternatively, add the following line to your Apache configuration file:

```
LoadModule mod_apparmor modules/mod_apparmor.so
```

5.1.1 Managing ChangeHat-Aware Applications

As with most of the Novell AppArmor tools, you can use two methods for managing ChangeHat, YaST or the command line interface. Managing ChangeHat-aware applications from the command line is much more flexible, but the process is also more complicated. Both methods allow you to manage the hats for your application and populate them with profile entries.

The following steps are a demonstration that adds hats to an Apache profile using YaST. In the *Add Profile Wizard*, the Novell AppArmor profiling utilities prompt you to create new hats for distinct URI requests. Choosing to create a new hat allows you to create individual profiles for each URI. You can create very tight rules for each request.

If the URI that is processed does not represent significant processing or otherwise does not represent a significant security risk, safely select *Use Default Hat* to process this URI in the default hat, which is the default security profile.

This example creates a new hat for the URI `phpsysinfo` and its subsequent accesses. Using the profiling utilities, delegate what to add to this new hat. The resulting hat becomes a tight-security container that encompasses all the processing on the server that occurs when the `phpsysinfo` URI is passed to the Apache Web server.

The URI runs the application phpsysinfo (refer to http://phpsysinfo.sourceforge.net for more information). The `phpsysinfo` package is assumed to be installed in `/srv/www/htdocs/phpsysinfo` in a clean (new) installation of openSUSE and AppArmor.

1 Once `phpsysinfo` is installed, you are ready to add hats to the Apache profile. From the Novell AppArmor GUI, select *Add Profile Wizard*.

This wizard will help you create a new AppArmor security
profile for an application, or you can use it to enhance
an existing profile by allowing AppArmor to learn new
application behavior.

Please enter the application name for which you would like
to create a profile, or selecte Browse to find the
application on your system.

Application to Profile

httpd2-prefork

Browse

Create   Abort

2 In *Application to Profile*, enter `httpd2-prefork`.

3 Click *Create Profile*.

AppArmor Profiling Wizard
This wizard presents entries
generated by the AppArmor
access control module. You
can generate highly
optimized and robust
security profiles by using the
suggestions made by
AppArmor. AppArmor
suggests that you allow or
deny access to specific
resources or define execute
permission for entries.
These questions that display
were logged during the
normal application execution
test previously performed.
The following help text
describes the detail of the
security profile syntax used
by AppArmor.

At any stage, you may
customize the profile entry
by changing the suggested
response. This overview will
assist you in your options.
Refer to the Admin Guide for
step-by-step instructions.

Access Modes

AppArmor Profile Wizard

Profiling /usr/sbin/httpd2-prefork

Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the 'Scan' option below in
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

Scan system log for AppArmor events

Back   Abort   Finish

4 Restart Apache by entering `rcapache2 restart` in a terminal window.

Restart any program you are profiling at this point.

5 Open `http://localhost/phpsysinfo/` in a Web browser window. The browser window should display network usage and system information.

> **NOTE: Data Caching**
>
> To ensure that this request is processed by the server and you do not review cached data in your browser, refresh the page. To do this, click the browser *Refresh* button to make sure that Apache processes the request for the `phpsysinfo` URI.

6 Click *Scan System Log for Entries to Add to Profiles*. Novell AppArmor launches the `aa-logprof` tool, which scans the information learned in the previous step. It begins to prompt you with profile questions.

7 `aa-logprof` first prompts with *Add Requested Hat* or *Use Default Hat* because it noticed that the `phpsysinfo` URI was accessed. Select *Add Requested Hat*.
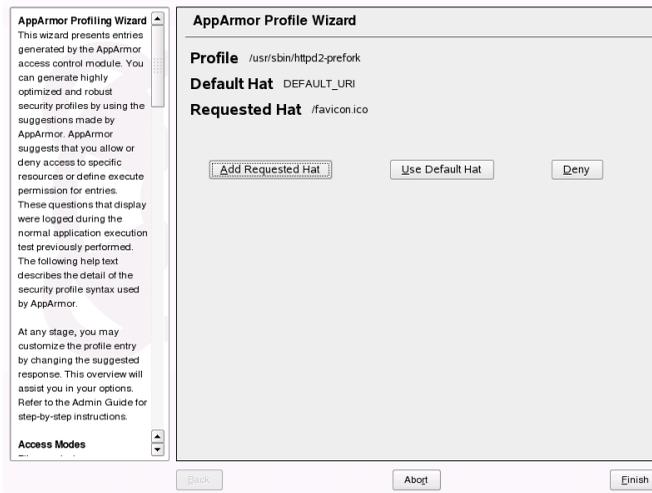
8 Click *Allow*.

Choosing *Add Requested Hat* in the previous step creates a new hat in the profile and specifies that the results of subsequent questions about the script's actions are added to the newly created hat rather than the default hat for this application.

In the next screen, Novell AppArmor displays an external program that the script executed. You can specify that the program should run confined by the phpsysinfo hat (choose *Inherit*), confined by a separate profile (choose *Profile*), or that it should run unconfined or without any security profile (choose *Unconfined*). For the case of the *Profile* option, a new profile is created for the program if one does not already exist.

NOTE: Security Considerations

Selecting *Unconfined* can create a significant security hole and should be done with caution.
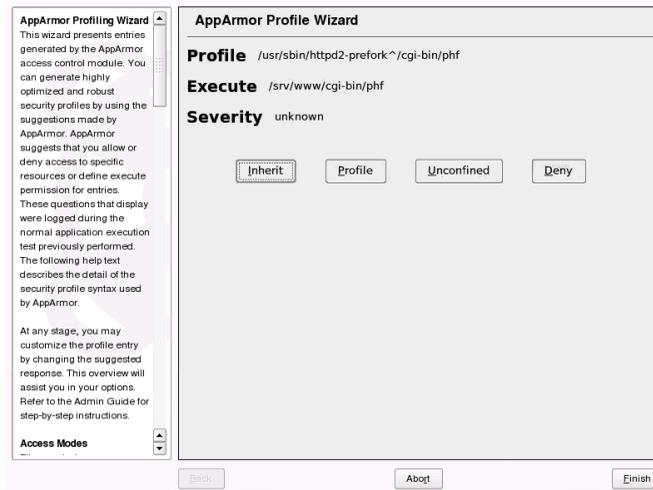
8a Select *Inherit* for the /bin/bash path. This adds /bin/bash (accessed by Apache) to the phpsysinfo hat profile with the necessary permissions.

8b Click *Allow*.

9 The remaining questions prompt you to generate new hats and add entries to your profile and its hats. The process of adding entries to profiles is covered in detail in the Section 3.1, "Adding a Profile Using the Wizard" (page 29).

When all profiling questions are answered, click *Finish* to save your changes and exit the wizard.

The following is an example phpsysinfo hat.

Example 5.1   *Example phpsysinfo Hat*

```
/usr/sbin/httpd2-prefork {
  ...
  ^phpsysinfo {
    #include <abstractions/bash>
    #include <abstractions/nameservice>

    /bin/basename                   ixr,
    /bin/bash                       ixr,
    /bin/df                         ixr,
    /bin/grep                       ixr,
    /bin/mount                      Ux,
    /bin/sed                        ixr,
    /dev/bus/usb/                   r,
    /dev/bus/usb/**                 r,
    /dev/null                       w,
    /dev/tty                        rw,
    /dev/urandom                    r,
    /etc/SuSE-release               r,
    /etc/ld.so.cache                r,
    /etc/lsb-release                r,
    /etc/lsb-release.d/             r,
    /lib/ld-2.6.1.so                ixr,
    /proc/**                        r,
    /sbin/lspci                     ixr,
    /srv/www/htdocs/phpsysinfo/**   r,
    /sys/bus/pci/**                 r,
    /sys/bus/scsi/devices/          r,
    /sys/devices/**                 r,
    /usr/bin/cut                    ixr,
    /usr/bin/getopt                 ixr,
    /usr/bin/head                   ixr,
    /usr/bin/lsb_release            ixr,
    /usr/bin/lsscsi                 ixr,
    /usr/bin/tr                     ixr,
    /usr/bin/who                    ixr,
    /usr/lib/lib*so*                mr,
    /usr/lib/locale/**              r,
    /usr/sbin/lsusb                 ixr,
    /usr/share/locale/**            r,
    /usr/share/pci.ids              r,
    /usr/share/usb.ids              r,
    /var/log/apache2/access_log     w,
    /var/run/utmp                   kr,
  }
}
```

NOTE: Hat and Parent Profile Relationship

The profile `^phpsysinfo` is only valid in the context of a process running under the parent profile `httpd2-prefork`.

5.1.2 Adding Hats and Entries to Hats

When you use the *Edit Profile* dialog (for instructions, refer to Section 3.3, "Editing Profiles" (page 38)) or when you add a new profile using *Manually Add Profile* (for instructions, refer to Section 3.2, "Manually Adding a Profile" (page 37)), you are given the option of adding hats (subprofiles) to your Novell AppArmor profiles. Add a ChangeHat subprofile from the *AppArmor Profile Dialog* window as in the following.
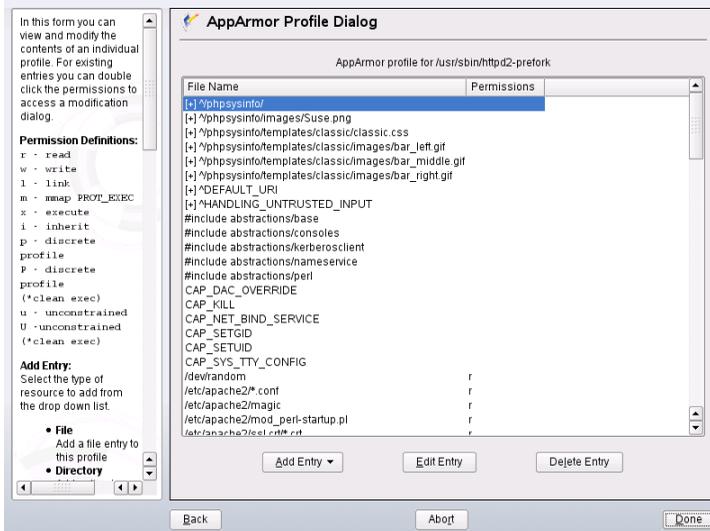
1 From the *AppArmor Profile Dialog* window, click *Add Entry* then select *Hat*. The *Enter Hat Name* dialog box opens:

Please enter the name of the Hat that you would like
to add to the profile /usr/sbin/httpd2-prefork.

Hat name to add

phpsysinfo

Create Hat   Abort

2 Enter the name of the hat to add to the Novell AppArmor profile. The name is the URI that, when accessed, receives the permissions set in the hat.

3 Click *Create Hat*. You are returned to the *AppArmor Profile Dialog* screen.

4 After adding the new hat, click *Done*.

NOTE: For More Information

For an example of an Novell AppArmor profile, refer to Example 5.1, "Example phpsysinfo Hat" (page 81).

5.2 Configuring Apache for mod_apparmor

Apache is configured by placing directives in plain text configuration files. The main configuration file is usually `httpd.conf`. When you compile Apache, you can indicate the location of this file. Directives can be placed in any of these configuration files to alter the way Apache behaves. When you make changes to the main configuration files, you need to start or restart Apache so the changes are recognized.

5.2.1 Virtual Host Directives

Virtual host directives control whether requests that contain trailing pathname information following an actual filename or that refer to a nonexistent file in an existing directory are accepted or rejected. For Apache documentation on virtual host directives, refer

to `http://httpd.apache.org/docs-2.2/mod/core.html#virtualhost`.

The ChangeHat-specific configuration keyword is `AADefaultHatName`. It is used similarly to `AAHatName`, for example, `AADefaultHatName My_Funky_Default_Hat`.

The configuration option is actually based on a server directive, which enables you to use the keyword outside of other options, setting it for the default server. Virtual hosts are considered internally within Apache to be separate "servers," so you can set a default hat name for the default server as well as one for each virtual host, if desired.

When a request comes in, the following steps reflect the sequence in which `mod_apparmor` attempts to apply hats.

1. A location or directory hat as specified by the `AAHatName` keyword

2. A hat named by the entire URI path

3. A default server hat as specified by the `AADefaultHatName` keyword

4. `DEFAULT_URI` (if none of those exist, it goes back to the "parent" Apache hat)

5.2.2 Location and Directory Directives

Location and directory directives specify hat names in the program configuration file so the program calls the hat regarding its security. For Apache, you can find documentation about the location and directory directives at `http://httpd.apache.org/docs-2.0/sections.html`.

The location directive example below specifies that, for a given location, `mod_apparmor` should use a specific hat:

```
<Location /foo/> AAHatName MY_HAT_NAME </Location>
```

This tries to use `MY_HAT_NAME` for any URI beginning with `/foo/` (`/foo/`, `/foo/bar`, `/foo/cgi/path/blah_blah/blah`, etc.).

The directory directive works similarly to the location directive, except it refers to a path in the file system as in the following example:

```
<Directory "/srv/www/www.immunix.com/docs">
  # Note lack of trailing slash
  AAHatName immunix.com
</Directory>
```

Example: The program phpsysinfo is used to illustrate a location directive in the following example. The tarball can be downloaded from http://phpsysinfo .sourceforge.com.

 1 After downloading the tarball, install it into `/srv/www/htdocs/ phpsysinfo`.

 2 Create `/etc/apache2/conf.d/phpsysinfo.conf` and add the following text to it:

```
<Location "/phpsysinfo">
  AAHatName phpsysinfo
</Location>
```

The following hat should then work for phpsysinfo:

```
/usr/sbin/httpd2-prefork {
  ...
  ^phpsysinfo {
    #include <abstractions/bash>
    #include <abstractions/nameservice>

    /bin/basename                  ixr,
    /bin/bash                      ixr,
    /bin/df                        ixr,
    /bin/grep                      ixr,
    /bin/mount                     Ux,
    /bin/sed                       ixr,
    /dev/bus/usb/                  r,
    /dev/bus/usb/**                r,
    /dev/null                      w,
    /dev/tty                       rw,
    /dev/urandom                   r,
    /etc/SuSE-release              r,
    /etc/ld.so.cache               r,
    /etc/lsb-release               r,
    /etc/lsb-release.d/            r,
    /lib/ld-2.6.1.so               ixr,
    /proc/**                       r,
    /sbin/lspci                    ixr,
    /srv/www/htdocs/phpsysinfo/**  r,
    /sys/bus/pci/**                r,
    /sys/bus/scsi/devices/         r,
```

```
        /sys/devices/**                        r,
        /usr/bin/cut                           ixr,
        /usr/bin/getopt                        ixr,
        /usr/bin/head                          ixr,
        /usr/bin/lsb_release                   ixr,
        /usr/bin/lsscsi                        ixr,
        /usr/bin/tr                            ixr,
        /usr/bin/who                           ixr,
        /usr/lib/lib*so*                       mr,
        /usr/lib/locale/**                     r,
        /usr/sbin/lsusb                        ixr,
        /usr/share/locale/**                   r,
        /usr/share/pci.ids                     r,
        /usr/share/usb.ids                     r,
        /var/log/apache2/access_log            w,
        /var/run/utmp                          kr,
    }
}
```

3 Reload Novell AppArmor profiles by entering `rcapparmor restart` at a terminal window as `root`.

4 Restart Apache by entering `rcapache2 restart` at a terminal window as `root`.

5 Enter `http://hostname/phpsysinfo/` into a browser to receive the system information that phpsysinfo delivers.

6 Locate configuration errors by going to `/var/log/audit/audit.log` or running `dmesg` and looking for any rejections in the output.

Managing Profiled Applications

6

After creating profiles and immunizing your applications, openSUSE® becomes more efficient and better protected if you perform Novell® AppArmor profile maintenance, which involves analyzing log files and refining your profiles as well as backing up your set of profiles and keeping it up-to-date. You can deal with these issues before they become a problem by setting up event notification by e-mail, running periodic reports, updating profiles from system log entries by running the aa-logprof tool through YaST, and dealing with maintenance issues.

6.1 Monitoring Your Secured Applications

Applications that are confined by Novell AppArmor security profiles generate messages when applications execute in unexpected ways or outside of their specified profile. These messages can be monitored by event notification, periodic report generation, or integration into a third-party reporting mechanism.

For reporting and alerting, AppArmor uses a userspace daemon (`/usr/sbin/aa-eventd`). This daemon monitors log traffic, sends out notifications, and runs scheduled reports. It does not require any end user configuration and it is started automatically as part of the security event notification through the YaST AppArmor Control Panel or by the configuration of scheduled reports in the YaST AppArmor Reports module.

Apart from transparently enabling and disabling aa-eventd with the YaST modules, you can manually toggle its status with the `rcaaeventd` init script. The AppArmor event daemon is not required for proper functioning of the profiling process (such as enforcement or learning). It is just required for reporting.

Find more details on security event notification in Section 6.2, "Configuring Security Event Notification" (page 88) and on scheduled reports in Section 6.3, "Configuring Reports" (page 91).

If you prefer a simple way of being notified of any AppArmor reject events that does not require you to check your e-mails or any log files, use the AppArmor Desktop Monitor applet that integrates into the GNOME desktop. Refer to Section 6.4, "Configuring and Using the AppArmor Desktop Monitor Applet" (page 111) for details.

6.2 Configuring Security Event Notification

Security event notification is a Novell AppArmor feature that informs you when systemic Novell AppArmor activity occurs. Activate it by selecting a notification frequency (receiving daily notification, for example). Enter an e-mail address, so you can be notified by e-mail when Novell AppArmor security events occur. Select one of the following notification types:

Terse

Terse notification summarizes the total number of system events without providing details. For example:

```
jupiter.example.com has had 41 security events since Mon Sep 10 14:53:16
 2007.
```

Summary Notification

Summary notification displays the logged Novell AppArmor security events and lists the number of individual occurrences, including the date of the last occurrence. For example:

```
AppArmor: PERMITTING access to capability 'setgid' (httpd2-prefork(6347)
 profile /usr/sbin/httpd2-prefork active /usr/sbin/httpd2-prefork) 2 times,
 the latest at Sat Oct  9 16:05:54 2004.
```

Verbose Notification

Verbose notification displays unmodified, logged Novell AppArmor security events. It tells you every time an event occurs and writes a new line in the verbose log. These security events include the date and time the event occurred, when the application profile permits and rejects access, and the type of file permission access that is permitted or rejected. Verbose notification also reports several messages that the aa-logprof tool (see Section "aa-logprof—Scanning the System Log" (page 67)) uses to interpret profiles. For example:

```
type=APPARMOR_DENIED msg=audit(1189428793.218:2880):
operation="file_permission" requested_mask="w" denied_mask="w"
name="/var/log/apache2/error_log" pid=22969
profile="/usr/sbin/httpd2-prefork"
```

NOTE

You must set up a mail server that can send outgoing mail using the SMTP protocol (for example, postfix or exim) for event notification to work.

---

1 In the *Enable Security Event Notification* section of the *AppArmor Configuration* window, click *Configure*.
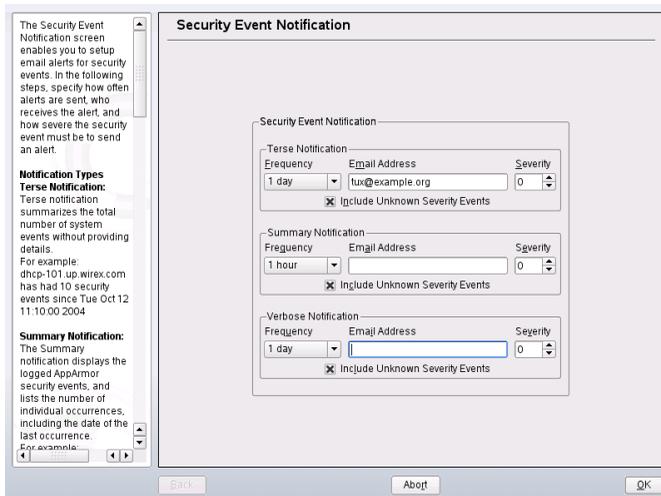
2 In the *Security Event Notification* window, enable *Terse*, *Summary*, or *Verbose* event notification.

2a In each applicable notification type section, enter the e-mail addresses of those who should receive notification in the field provided. If notification is enabled, you must enter an e-mail address. Separate multiple e-mail addresses with commas.

2b For each notification type enabled, select the frequency of notification.

Select a notification frequency from the following options:

- Disabled

- 1 minute

- 5 minutes

- 10 minutes

- 15 minutes

- 30 minutes

- 1 hour

- 1 day

- 1 week

2c For each selected notification type, select the lowest severity level for which a notification should be sent. Security events are logged and the notifications are sent at the time indicated by the interval when events are equal to or greater than the selected severity level. If the interval is *1 day*, the notification is sent daily, if security events occur.

> **NOTE: Severity Levels**
>
> Novell AppArmor sends out event messages for things that are in the severity database and above the level selected. Severity levels are numbered 1 through 10, with 10 being the most severe security incident. The `/etc/severity.db` file defines the severity level of potential security events. The severity levels are determined by the

> importance of different security events, such as certain resources
> accessed or services denied.

3 Click *OK*.

4 Click *Done* in the *Novell AppArmor Configuration* window.

5 Click *File > Quit* in the YaST Control Center.

After configuring security event notification, read the reports and determine whether events require follow up. Follow up may include the procedures outlined in Section 6.5, "Reacting to Security Event Rejections" (page 112).

6.3 Configuring Reports

Novell AppArmor's reporting feature adds flexibility by enhancing the way users can view security event data. The reporting tool performs the following:

- Creates on-demand reports

- Exports reports

- Schedules periodic reports for archiving

- E-mails periodic reports

- Filters report data by date

- Filters report data by other options, such as program name

Using reports, you can read important Novell AppArmor security events reported in the log files without manually sifting through the messages only useful to the aa-logprof tool. Narrow down the size of the report by filtering by date range or program name. You can also export an `html` or `csv` file.

The following are the three types of reports available in Novell AppArmor:

Executive Security Summary

A combined report, consisting of one or more security incident reports from one or more machines. This report can provide a single view of security events on multiple machines. For more details, refer to Section "Executive Security Summary" (page 101).

Application Audit Report

An auditing tool that reports which application servers are running and whether the applications are confined by AppArmor. Application servers are applications that accept incoming network connections. For more details, refer to Section "Application Audit Report" (page 97).

Security Incident Report

A report that displays application security for a single host. It reports policy violations for locally confined applications during a specific time period. You can edit and customize this report or add new versions. For more details, refer to Section "Security Incident Report" (page 99).

To use the Novell AppArmor reporting features, proceed with the following steps:

1 Open *YaST > Novell AppArmor*.

2 In *Novell AppArmor*, click *AppArmor Reports*. The *AppArmor Security Event Reports* window appears. From the *Reports* window, select an option and proceed to the respective section for instructions:
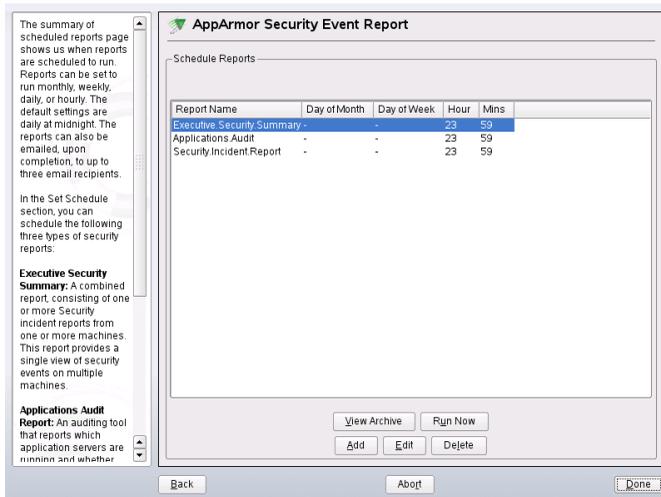
View Archive

Displays all reports that have been run and stored in `/var/log/apparmor/reports-archived/`. Select the report you want to see in detail and click *View*. For *View Archive* instructions, proceed to Section 6.3.1, "Viewing Archived Reports" (page 94).

Run Now

Produces an instant version of the selected report type. If you select a security incident report, it can be further filtered in various ways. For *Run Now* instructions, proceed to Section 6.3.2, "Run Now: Running On-Demand Reports" (page 102).

Add

Creates a scheduled security incident report. For *Add* instructions, proceed to Section 6.3.3, "Adding New Reports" (page 105).

Edit

Edits a scheduled security incident report.

Delete

Deletes a scheduled security incident report. All stock or canned reports cannot be deleted.

Back
> Returns you to the Novell AppArmor main screen.

Abort
> Returns you to the Novell AppArmor main screen.

Next
> Performs the same function as the *Run Now* button.

6.3.1 Viewing Archived Reports

View Reports enables you to specify the location of a collection of reports from one or more systems, including the ability to filter by date or names of programs accessed and display them all together in one report.

1 From the *AppArmor Security Event Report* window, select *View Archive*.
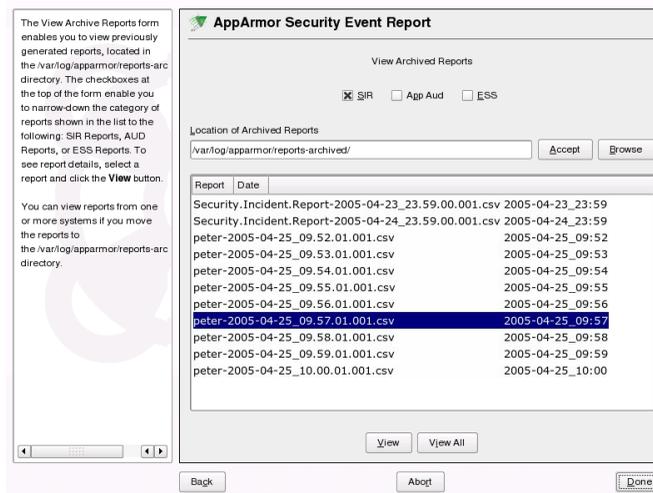
2 Select the report type to view. Toggle between the different types: *SIR* (Security Incident Report), *App Aud* (Application Audit), and *ESS* (Executive Security Summary).

3 You can alter the directory location of the archived reports in *Location of Archived Reports*. Select *Accept* to use the current directory or select *Browse* to find a new report location. The default directory is `/var/log/apparmor/reports-archived`.

4 To view all the reports in the archive, select *View All*. To view a specific report, select a report file listed in the *Report* field then select *View*.

5 For *Application Audit* and *Executive Security Summary* reports, proceed to Step 9 (page 97).

6 The *Report Configuration Dialog* opens for *Security Incident* reports.

7 The *Report Configuration* dialog enables you to filter the reports selected in the previous screen. Enter the desired filter details. The fields are:

Date Range
> To display reports for a certain time period, select *Filter By Date Range*. Enter the start and end dates that define the scope of the report.

Program Name

> When you enter a program name or pattern that matches the name of the binary executable of the program of interest, the report displays security events that have occurred for a specific program.

Profile Name

> When you enter the name of the profile, the report displays the security events that are generated for the specified profile. You can use this to see what is being confined by a specific profile.

PID Number

> *PID number* is a number that uniquely identifies one specific process or running program (this number is valid only during the lifetime of that process).

Severity

> Select the lowest severity level for security events to include in the report. The selected severity level and above are then included in the reports.

Detail

> A source to which the profile has denied access. This includes capabilities and files. You can use this field to report the resources to which profiles prevent access.

Access Type

> The access type describes what is actually happening with the security event. The options are PERMITTING, REJECTING, or AUDITING.

Mode

> The *Mode* is the permission that the profile grants to the program or process to which it is applied. The options are all (all modes without filtering), r (read), w (write), l (link), x (execute), and m (mmap).

Export Type

> Enables you to export a CSV (comma separated values) or HTML file. The CSV file separates pieces of data in the log entries with commas using a standard data format for importing into table-oriented applications. You can enter a path for your exported report by typing the full path in the field provided.

Location to Store Log

Enables you to change the location at which to store the exported report. The default location is `/var/log/apparmor/reports-exported`. When you change this location, select *Accept*. Select *Browse* to browse the file system.

8 To see the report, filtered as desired, select *Next*. One of the three reports displays.

9 Refer the following sections for detailed information about each type of report.

- For the application audit report, refer to Section "Application Audit Report" (page 97).

- For the security incident report, refer to Section "Security Incident Report" (page 99).

- For the executive summary report, refer to Section "Executive Security Summary" (page 101).

Application Audit Report

An application audit report is an auditing tool that reports which application servers are running and whether they are confined by AppArmor.

AppArmor On-Demand Report

Applications Audit Report

Host	Date	Program	Profile	PID	State
K23	Tue Sep 11 11:17:36 2007	/sbin/dhcpcd	-	8732	not-confine
K23	Tue Sep 11 11:17:36 2007	/sbin/portmap	-	8952	not-confine
K23	Tue Sep 11 11:17:36 2007	/sbin/portmap	-	8952	not-confine
K23	Tue Sep 11 11:17:36 2007	/usr/sbin/cupsd	-	8972	not-confine
K23	Tue Sep 11 11:17:36 2007	/usr/sbin/cupsd	-	8972	not-confine
K23	Tue Sep 11 11:17:36 2007	/usr/sbin/cupsd	-	8972	not-confine
K23	Tue Sep 11 11:17:36 2007	/usr/lib/postfix/master	-	9017	not-confine
K23	Tue Sep 11 11:17:36 2007	/usr/lib/postfix/master	-	9017	not-confine
K23	Tue Sep 11 11:17:36 2007	/usr/sbin/sshd	-	9133	not-confine
K23	Tue Sep 11 11:17:36 2007	/usr/sbin/httpd2-prefork	/usr/sbin/httpd2-prefork	27468	confined
K23	Tue Sep 11 11:17:36 2007	/usr/bin/Xnest	-	30097	not-confine
K23	Tue Sep 11 11:17:36 2007	/usr/bin/Xnest	-	30097	not-confine
K23	Tue Sep 11 11:17:36 2007	-	-	[?1034h	-

First Page   Previous   Sort   Forward   Last Page   Go to Page

Back   Abort   Done

The following fields are provided in an application audit report:

Host

The machine protected by AppArmor for which the security events are reported.

Date

The date during which security events occurred.

Program

The name and path of the executing process.

Profile

The absolute name of the security profile that is applied to the process.

PID

A number that uniquely identifies one specific process or running program (this number is valid only during the lifetime of that process).

State

This field reveals whether the program listed in the program field is confined. If it is not confined, you might consider creating a profile for it.

Type
This field reveals the type of confinement the security event represents. It says either complain or enforce. If the application is not confined (state), no type of confinement is reported.

Security Incident Report

A security incident report displays security events of interest to an administrator. The SIR reports policy violations for locally confined applications during the specified time period. It also reports policy exceptions and policy engine state changes. These two types of security events are defined as follows:

Policy Exceptions
When an application requests a resource that is not defined within its profile, a security event is triggered. A report is generated that displays security events of interest to an administrator. The SIR reports policy violations for locally confined applications during the specified time period. The SIR reports policy exceptions and policy engine state changes.

Policy Engine State Changes
Enforces policy for applications and maintains its own state, including when engines start or stop, when a policy is reloaded, and when global security feature are enabled or disabled.
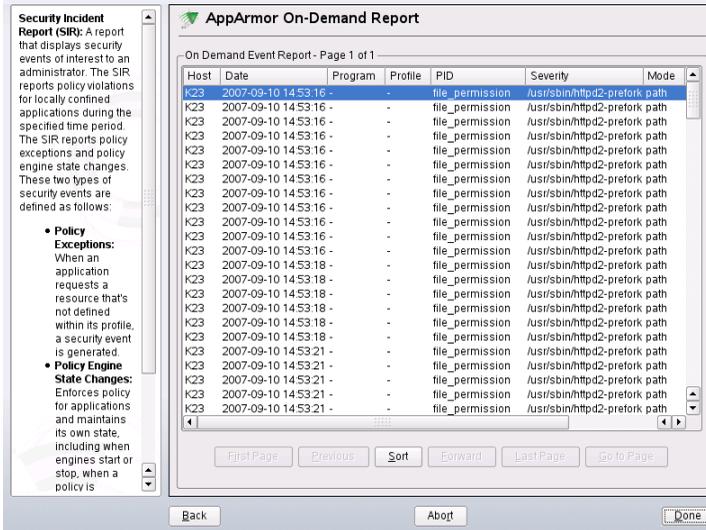
Security Incident Report (SIR): A report that displays security events of interest to an administrator. The SIR reports policy violations for locally confined applications during the specified time period. The SIR reports policy exceptions and policy engine state changes. These two types of security events are defined as follows:

- **Policy Exceptions:** When an application requests a resource that's not defined within its profile, a security event is generated.
- **Policy Engine State Changes:** Enforces policy for applications and maintains its own state, including when engines start or stop, when a policy is

AppArmor On-Demand Report

On Demand Event Report - Page 1 of 1

Host	Date	Program	Profile	PID	Severity	Mode
K23	2007-09-10 14:53:16	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:16	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:16	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:16	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:16	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:16	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:16	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:16	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:16	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:16	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:18	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:18	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:18	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:18	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:18	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:21	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:21	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:21	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:21	-	-	file_permission	/usr/sbin/httpd2-prefork	path
K23	2007-09-10 14:53:21	-	-	file_permission	/usr/sbin/httpd2-prefork	path

First Page   Previous   Sort   Forward   Last Page   Go to Page

Back   Abort   Done

The fields in the SIR report have the following meanings:

Host

The machine protected by AppArmor for which the security events are reported.

Date

The date during which security events occurred.

Program

The name of the executing process.

Profile

The absolute name of the security profile that is applied to the process.

PID

A number that uniquely identifies one specific process or running program (this number is valid only during the lifetime of that process).

Severity

Severity levels of events are reported from the severity database. The severity database defines the importance of potential security events and numbers them 1 through 10, 10 being the most severe security incident. The severity levels are de-

termined by the threat or importance of different security events, such as certain resources accessed or services denied.

Mode
> The mode is the permission that the profile grants to the program or process to which it is applied. The options are `r` (read), `w` (write), `l` (link), and `x` (execute).
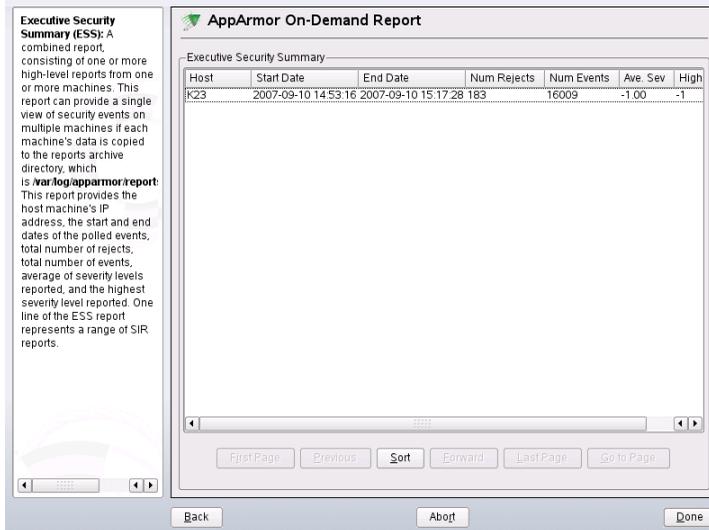
Detail
> A source to which the profile has denied access. This includes capabilities and files. You can use this field to report the resources to which the profile prevents access.

Access Type
> The access type describes what is actually happening with the security event. The options are `PERMITTING`, `REJECTING`, or `AUDITING`.

Executive Security Summary

A combined report consisting of one or more high-level reports from one or more machines. This report can provide a single view of security events on multiple machines if each machine's data is copied to the report archive directory, which is `/var/log/apparmor/reports-archived`. One line of the ESS report represents a range of SIR reports.

The following fields are provided in an executive security summary:

Host

The machine protected by AppArmor for which the security events are reported.

Start Date

The first date in a range of dates during which security events are reported.

End Date

The last date in a range of dates during which security events are reported.

Num Rejects

In the date range given, the total number of security events that are rejected access attempts.

Num Events

In the date range given, the total number of security events.

Ave. Sev

This is the average of the severity levels reported in the date range given. Unknown severities are disregarded in this figure.

High Sev

This is the severity of the highest severity event reported in the date range given.

6.3.2 Run Now: Running On-Demand Reports

The *Run Now* report feature enables you to instantly extract report information from the Novell AppArmor event logs without waiting for scheduled events. If you need help navigating to the main report screen, see . Perform the following steps to run a report from the list of reports:

   1 Select the report to run instantly from the list of reports in the *Schedule Reports* window.

   2 Select *Run Now* or *Next*. The next screen depends on which report you selected in the previous step. As an example, select a security incident report.

3 The *Report Configuration Dialog* opens for security incident reports.
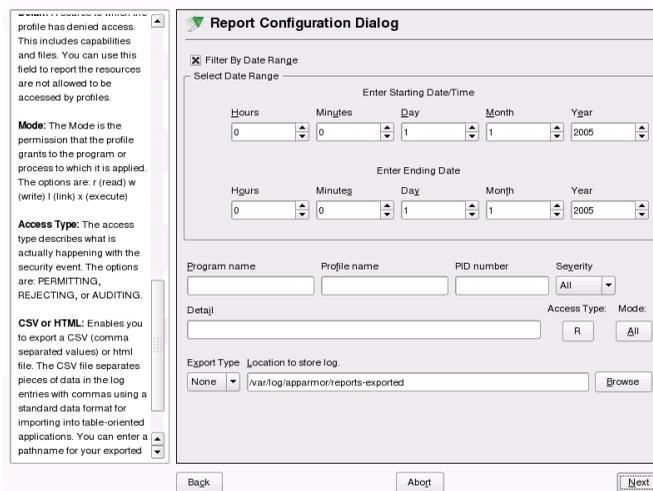
4 The *Report Configuration Dialog* enables you to filter the reports selected in the previous screen. Enter the desired filter details. The following filter options are available:

Date Range
>   To limit reports to a certain time period, select *Filter By Date Range*. Enter the start and end dates that determine the scope of the report.

Program Name
>   When you enter a program name or pattern that matches the name of the binary executable for the program of interest, the report displays security events that have occurred for the specified program only.

Profile Name
>   When you enter the name of the profile, the report displays the security events that are generated for the specified profile. You can use this to see what is confined by a specific profile.

PID Number
>   A number that uniquely identifies one specific process or running program (this number is valid only during the lifetime of that process).

Severity

    Select the lowest severity level for security events to include in the report. The selected severity level and above are included in the reports.

Detail

    A source to which the profile has denied access. This includes capabilities and files. You can use this field to report the resources to which profiles prevent access.

Access Type

    The access type describes what is actually happening with the security event. The options are PERMITTING, REJECTING, or AUDITING.

Mode

    The mode is the permission that the profile grants to the program or process to which it is applied. The options are r (read), w (write), l (link), and x (execute).

Export Type

    Enables you to export a CSV (comma separated values) or HTML file. The CSV file separates pieces of data in the log entries with commas using a standard data format for importing into table-oriented applications. Enter a path for your exported report by typing in the full path in the field provided.

Location to Store Log

    Enables you to change the location that the exported report is stored. The default location is /var/log/apparmor/reports-exported. When you change this location, select *Accept*. Select *Browse* to browse the file system.

5 To see the report, filtered as desired, select *Next*. One of the three reports displays.

Refer the following sections for detailed information about each type of report.

- For the application audit report, refer to Section "Application Audit Report" (page 97).

- For the security incident report, refer to Section "Security Incident Report" (page 99).

• For the executive summary report, refer to Section "Executive Security Summary" (page 101).

6.3.3 Adding New Reports

Adding new reports enables you to create a scheduled security incident report that displays Novell AppArmor security events according to your preset filters. When a report is set up in *Schedule Reports*, it periodically launches a report of Novell AppArmor security events that have occurred on the system.

You can configure a daily, weekly, monthly, or hourly report to run for a specified period. You can set the report to display rejections for certain severity levels or to filter by program name, profile name, severity level, or denied resources. This report can be exported to an HTML (Hypertext Markup Language) or CSV (Comma Separated Values) file format.

NOTE

Return to the beginning of this section if you need help navigating to the main report screen (see Section 6.3, "Configuring Reports" (page 91)).

To add a new scheduled security incident report, proceed as follows:

1 Click *Add* to create a new security incident report. The first page of *Add Scheduled SIR* opens.

2 Fill in the fields with the following filtering information, as necessary:

Report Name
> Specify the name of the report. Use names that easily distinguish different reports.

Day of Month
> Select any day of the month to activate monthly filtering in reports. If you select `All`, monthly filtering is not performed.

Day of Week
> Select the day of the week on which to schedule weekly reports, if desired. If you select `ALL`, weekly filtering is not performed. If monthly reporting is selected, this field defaults to `ALL`.

Hour and Minute
> Select the time. This specifies the hour and minute that you would like the reports to run. If you do not change the time, selected reports runs at midnight. If neither month nor day of week are selected, the report runs daily at the specified time.

E-Mail Target
> You have the ability to send the scheduled security incident report via e-mail to up to three recipients. Just enter the e-mail addresses for those who require the security incident information.

Export Type
> This option enables you to export a CSV (comma separated values) or HTML file. The CSV file separates pieces of data in the log entries with commas using a standard data format for importing into table-oriented applications. Enter a path for your exported report by typing in the full path in the field provided.

Location to Store Log
> Enables you to change the location that the exported report is stored. The default location is `/var/log/apparmor/reports-exported`. When you change this location, select *Accept*. Select *Browse* to browse the file system.

3 Click *Next* to proceed to the second page of *Add Scheduled SIR*.

4 Fill in the fields with the following filtering information, as necessary:

Program Name

> You can specify a program name or pattern that matches the name of the
> binary executable for the program of interest. The report displays security
> events that have occurred for the specified program only.

Profile Name

> You can specify the name of the profile for which the report should display
> security events. You can use this to see what is being confined by a specific
> profile.

PID Number

> A number that uniquely identifies one specific process or running program
> (this number is valid only during the lifetime of that process).

Detail

> A source to which the profile has denied access. This includes capabilities
> and files. You can use this field to create a report of resources to which
> profiles prevent access.

Severity

> Select the lowest severity level of security events to include in the report.
> The selected severity level and above are included in the reports.

Access Type

The access type describes what is actually happening with the security event. The options are `PERMITTING`, `REJECTING`, or `AUDITING`.

Mode

The mode is the permission that the profile grants to the program or process to which it is applied. The options are `r` (read), `w` (write), `l` (link), and `x` (execute).

5 Click *Save* to save this report. Novell AppArmor returns to the *Scheduled Reports* main window where the newly scheduled report appears in the list of reports.

6.3.4 Editing Reports

From the AppArmor *Reports* screen, you can select and edit a report. The three pre-configured reports (*stock reports*) cannot be edited or deleted.

NOTE

Return to the beginning of this section if you need help navigating to the main report screen (see Section 6.3, "Configuring Reports" (page 91)).

Perform the following steps to modify a report from the list of reports:

1 From the list of reports in the *Schedule Reports* window, select the report to edit. This example assumes that you have selected a security incident report.

2 Click *Edit* to edit the security incident report. The first page of the *Edit Scheduled SIR* displays.

Edit Report Schedule for   Security.Incident.Report

Day of Month	Day of Week	Hour		Minute	
All	Sun	23		59	

Email Target 1	Email Target 2	Email Target 3
root@localhost		

Export Type	Location to store log.		
Both	/var/log/apparmor/reports-exported	Accept	Browse

Cancel   Next

3 Modify the following filtering information, as necessary:

Day of Month
> Select any day of the month to activate monthly filtering in reports. If you select `All`, monthly filtering is not performed.

Day of Week
> Select the day of the week on which to schedule the weekly reports. If you select `All`, weekly filtering is not performed. If monthly reporting is selected, this defaults to `All`.

Hour and Minute
> Select the time. This specifies the hour and minute that you would like the reports to run. If you do not change the time, the selected report runs at midnight. If neither the day of the month nor day of the week is selected, the report runs daily at the specified time.

E-Mail Target
> You have the ability to send the scheduled security incident report via e-mail to up to three recipients. Just enter the e-mail addresses for those who require the security incident information.

Export Type
> This option enables you to export a CSV (comma separated values) or HTML file. The CSV file separates pieces of data in the log entries with commas using a standard data format for importing into table-oriented applications. Enter a path for your exported report by typing the full path in the field provided.

Location to Store Log
> Enables you to change the location where the exported report is stored. The default location is `/var/log/apparmor/reports-exported`. When you change this location, select *Accept*. Select *Browse* to browse the file system.

4 Click *Next* to proceed to the next *Edit Scheduled SIR* page. The second page of *Edit Scheduled Reports* opens.

Program name Profile name
httpd2-prefork

PID number Detail

Severity Access Type: Mode:
All ▼ R All

Cancel Save

5 Modify the fields with the following filtering information, as necessary:

Program Name

You can specify a program name or pattern that matches the name of the binary executable for the program of interest. The report displays security events that have occurred for the specified program only.

Profile Name

You can specify the name of the profile for which to display security events. You can use this to see what is being confined by a specific profile.

PID Number

Process ID number is a number that uniquely identifies one specific process or running program (this number is valid only during the lifetime of that process).

Detail

A source to which the profile has denied access. This includes capabilities and files. You can use this field to create a report of resources to which profiles prevent access.

Severity

Select the lowest severity level for security events to include in the report. The selected severity level and above are included in the reports.

Access Type
: The access type describes what is actually happening with the security event. The options are PERMITTING, REJECTING, or AUDITING.

Mode
: The mode is the permission that the profile grants to the program or process to which it is applied. The options are `r` (read), `w` (write), `l` (link), and `x` (execute).

6 Select *Save* to save the changes to this report. Novell AppArmor returns to the *Scheduled Reports* main window where the scheduled report appears in the list of reports.

6.3.5 Deleting Reports

Delete a Report enables you to permanently remove a report from the list of Novell AppArmor scheduled reports. To delete a report, follow these instructions:

1 To remove a report from the list of reports, highlight the report and click *Delete*.

2 From the confirmation pop-up, select *Cancel* if you do not want to delete the selected report. If you are sure you want to remove the report permanently from the list of reports, select *Delete*.

6.4 Configuring and Using the AppArmor Desktop Monitor Applet

The Linux audit framework contains a dispatcher that can send AppArmor events to any consumer application via dbus. The GNOME AppArmor Desktop Monitor applet is one example of an application that gathers AppArmor events via dbus. To configure audit to use the dbus dispatcher, just set the dispatcher in your audit configuration in `/etc/audit/auditd.conf` to `apparmor-dbus` and restart auditd:

```
dispatcher=/usr/bin/apparmor-dbus
```

Once the dbus dispatcher is configured correctly, add the AppArmor Desktop Monitor to the GNOME panel by right-clicking the panel and selecting *Add to Panel > AppArmor Desktop Monitor*. As soon as a `REJECT` event is logged, the applet's panel icon changes appearance and you can click the applet to see the number of reject events per confined application. To view the exact log messages, refer to the audit log under `/var/log/audit/audit.log`. React to any `REJECT` events as described in Section 6.5, "Reacting to Security Event Rejections" (page 112).

6.5 Reacting to Security Event Rejections

When you receive a security event rejection, examine the access violation and determine if that event indicated a threat or was part of normal application behavior. Application-specific knowledge is required to make the determination. If the rejected action is part of normal application behavior, run `aa-logprof` at the command line or the *Update Profile Wizard* in Novell AppArmor to update your profile.

If the rejected action is not part of normal application behavior, this access should be considered a possible intrusion attempt (that was prevented) and this notification should be passed to the person responsible for security within your organization.

6.6 Maintaining Your Security Profiles

In a production environment, you should plan on maintaining profiles for all of the deployed applications. The security policies are an integral part of your deployment. You should plan on taking steps to back up and restore security policy files, plan for software changes, and allow any needed modification of security policies that your environment dictates.

6.6.1 Backing Up Your Security Profiles

Because you take the time to make profiles, it makes sense to back them up. Backing up profiles might save you from having to reprofile all your programs after a disk crash. Also, if profiles are changed, you can easily restore previous settings by using the backed up files.

Back up profiles by copying the profile files to a specified directory.

1 You should first archive the files into one file. To do this, open a terminal window and enter the following as `root`:

```
tar zclpf profiles.tgz /etc/apparmor.d
```

The simplest method to ensure that your security policy files are regularly backed up is to include the directory `/etc/apparmor.d` in the list of directories that your backup system archives.

2 You can also use `scp` or a file manager like Konqueror or Nautilus to store the files on some kind of storage media, the network, or another computer.

6.6.2 Changing Your Security Profiles

Maintenance of security profiles includes changing them if you decide that your system requires more or less security for its applications. To change your profiles in Novell AppArmor, refer to Section 3.3, "Editing Profiles" (page 38).

6.6.3 Introducing New Software into Your Environment

When you add a new application version or patch to your system, you should always update the profile to fit your needs. You have several options that depend on your company's software deployment strategy. You can deploy your patches and upgrades into a test or production environment. The following explains how to do this with each method.

If you intend to deploy a patch or upgrade in a test environment, the best method for updating your profiles is one of the following:

• Run the profiling wizard by selecting *Add Profile Wizard* in YaST. This creates a new profile for the added or patched application. For step-by-step instructions, refer to Section 3.1, "Adding a Profile Using the Wizard" (page 29).

• Run aa-genprof by typing `aa-genprof` in a terminal while logged in as `root`. For detailed instructions, refer to Section "aa-genprof—Generating Profiles" (page 59).

If you intend to deploy a patch or upgrade directly into a production environment, the best method for updating your profiles is one of the following:

- Monitor the system frequently to determine if any new rejections should be added to the profile and update as needed using aa-logprof. For detailed instructions, refer to Section "aa-logprof—Scanning the System Log" (page 67).

- Run the YaST *Update Profile Wizard* to learn the new behavior (high security risk as all accesses are allowed and logged, not rejected). For step-by-step instructions, refer to Section 3.5, "Updating Profiles from Log Entries" (page 44).

Support

7

This chapter outlines maintenance-related tasks. Learn how to update Novell® App-Armor and get a list of available man pages providing basic help for using the command line tools provided by Novell AppArmor. Use the troubleshooting section to learn about some common problems encountered with Novell AppArmor and their solutions. Report defects or enhancement requests for Novell AppArmor following the instructions in this chapter.

7.1 Updating Novell AppArmor Online

Updates for Novell AppArmor packages are provided in the same way as any other update for openSUSE. Retrieve and apply them exactly like for any other package that ships as part of openSUSE.

7.2 Using the Man Pages

There are man pages available for your use. In a terminal, enter `man apparmor` to open the apparmor man page. Man pages are distributed in sections numbered 1 through 8. Each section is specific to a category of documentation:

Table 7.1   *Man Pages: Sections and Categories*

Section	Category
1	User commands
2	System calls
3	Library functions
4	Device driver information
5	Configuration file formats
6	Games
7	High level concepts
8	Administrator commands

The section numbers are used to distinguish man pages from each other. For example, `exit(2)` describes the exit system call, while `exit(3)` describes the exit C library function.

The Novell AppArmor man pages are:

- unconfined(8)

- autodep(1)

- complain(1)

- enforce(1)

- genprof(1)

- logprof(1)

- change_hat(2)

- logprof.conf(5)

- `apparmor.conf(5)`

- `apparmor.d(5)`

- `apparmor.vim(5)`

- `apparmor(7)`

- `apparmor_parser(8)`

7.3 For More Information

Find more information about the AppArmor product on the Novell AppArmor product page at Novell: `http://www.novell.com/products/apparmor/`. Find the product documentation for Novell AppArmor, including this document, at `http://www.novell.com/documentation/apparmor/` or in the installed system in `/usr/share/doc/manual`.

There are specific mailing lists for AppArmor that users can post to or join to communicate with developers.

apparmor-general@forge.novell.com [`mailto:apparmor-general@forge.novell.com`]
> This is a mailing list for end users of AppArmor. It is a good place for questions about how to use AppArmor to protect your applications.

apparmor-dev@forge.novell.com [`mailto:apparmor-dev@forge.novell.com`]
> This is a developer mailing list for AppArmor developers and community members. This list is for questions about development of core AppArmor features—the kernel module and the profiling tools. If you are interested in reviewing the code for AppArmor and contributing reviews or patches, this would be the list for you.

apparmor-announce@forge.novell.com [`mailto:apparmor-announce@forge.novell.com`]
> This is a low traffic list announcing the availability of new releases or features.

7.4 Troubleshooting

This section lists the most common problems and error messages that may occur using Novell AppArmor.

7.4.1 How to React to odd Application Behavior?

If you notice odd application behavior or any other type of application problem, you should first check the reject messages in the log files to see if AppArmor is too closely constricting your application. To check reject messages, start *YaST > Novell AppArmor* and go to *AppArmor Reports*. Select *View Archive* and *App Aud* for the application audit report. You can filter dates and times to narrow down the specific periods when the unexpected application behavior occurred.

If you detect reject messages that indicate that your application or service is too closely restricted by AppArmor, update your profile to properly handle your use case of the application. Do this with the *Update Profile Profile Wizard* in YaST, as described in Section 3.5, "Updating Profiles from Log Entries" (page 44).

If you decide to run your application or service without AppArmor protection, remove the application's profile from /etc/apparmor.d or move it to another location.

7.4.2 My Profiles do not Seem to Work Anymore ...

If you have been using previous versions of AppArmor and have updated your system, but kept your old set of profiles, you might notice some applications behaving strangely or not working at all which seemed to work perfectly before you updated.

This version of AppArmor introduces a set of new features to the profile syntax and the AppArmor tools that might cause trouble with older versions of the AppArmor profiles. The features concerned are:

- File Locking

- Network Access Control

- The SYS_PTRACE Capability

- Directory Path Access

The current version of AppArmor mediates file locking and introduces a new permission mode (k) for this. Applications requesting file locking permission might misbehave or fail altogether if confined by older profiles which do not explicitly contain permissions to lock files. If you suspect this being the case, check the log file under /var/log/audit/audit.log for entries like the following:

```
type=APPARMOR_DENIED msg=audit(1188913493.299:9304): operation="file_lock"
requested_mask="k" denied_mask="k" name="/home/tux/.qt/.qtrc.lock" pid=25736
 profile="/usr/bin/opera"
```

Update the profile using the YaST Update Profile Wizard or the aa-logprof command as outlined below.

The new network access control syntax based on the network family and type specification, described in Section 2.1.1, "Network Access Control" (page 14), might cause application misbehavior or even stop applications from working. If you notice a network-related application behaving strangely, check the log file under /var/log/audit/audit.log for entries like the following:

```
type=APPARMOR_DENIED msg=audit(1188894313.206:9123): operation="socket_create"
 family="inet" sock_type="raw" protocol=1 pid=23810 profile="/bin/ping"
```

This log entry means that our example application, /bin/ping in this case, failed to get AppArmor's permission to open a network connection. This permission has to be explicitly stated to make sure that an application has network access. To update the profile to the new syntax, use the YaST Update Profile Wizard or the aa-logprof command as outlined below.

The current kernel requires the SYS_PTRACE capability, if a process tries to access files in /proc/*pid*/fd/*. New profiles need an entry for the file and the capability where old profiles only needed the file entry. For example:

```
/proc/*/fd/**  rw,
```

in the old syntax would translate to the following rules in the new syntax:

```
capability SYS_PTRACE,
/proc/*/fd/**  rw,
```

To update the profile to the new syntax, use the YaST Update Profile Wizard or the `aa-logprof` command as outlined below.

With this version of AppArmor, a few changes have been made to the profile rule syntax to better distinguish directory from file access. Therefore, some rules matching both file and directory paths in the previous version might now just match a file path. This could lead to AppArmor not being able to access a crucial directory at all and thus trigger misbehavior of your application and various log messages. The following examples highlight the most important changes to the path syntax.

Using the old syntax, the following rule would allow access to files and directories in `/proc/net`. It would allow directory access only to read the entries in the directory, but not give access to files or directories under the directory, e.g. `/proc/net/dir/foo` would be matched by the asterisk (*), but as `foo` is a file or directory under `dir`, it cannot be accessed.

```
/proc/net/*  r,
```

To get the same behavior using the new syntax, you need two rules instead of one. The first allows access to file under `/proc/net` and the second allows access to directories under `/proc/net`. Directory access can only be used for listing the contents, not to actually access files or directories underneath the directory.

```
/proc/net/*  r,
/proc/net/*/  r,
```

The following rule works similarly both under the old and the new syntax and allows access to both files and directories under `/proc/net`:

```
/proc/net/**  r,
```

To distinguish file from directory access using the above expression in the new syntax, use the following two rules. The first one only allows to recursively access directories under `/proc/net` while the second one explicitly allows for recursive file access only.

```
/proc/net/**/  r,
/proc/net/**[^/]  r,
```

The following rule works similarly both under the old and the new syntax and allows access to both files and directories beginning with `foo` under `/proc/net`:

```
/proc/net/foo**  r,
```

To distinguish file from directory access in the new syntax and use the `**` globbing pattern, use the following two rules. The first one would have matched both files and directories in the old syntax, but only matches files in the new syntax due to the missing trailing slash. The second rule matched neither file nor directory in the old syntax, but matches directories only in the new syntax:

```
/proc/net/**foo   r,
/proc/net/**foo/  r,
```

The following rules illustrate how the use of the `?` globbing pattern has changed. In the old syntax, the first rule would have matched both files and directories (four characters, last character could be any but a slash). In the new syntax, it matches only files (trailing slash is missing). The second rule would match nothing in the old profile syntax, but matches directories only in the new syntax. The last rule matches explicitly matches a file called `bar` under `/proc/net/foo?`. Using the old syntax, this rule would have applied to both files and directories:

```
/proc/net/foo?  r,
/proc/net/foo?/  r,
/proc/net/foo?/bar  r,
```

To find and resolve issues related to syntax changes, take some time after the update to check the profiles you want to keep and proceed as follows for each application you kept the profile for:

1 Make sure that AppArmor is running and that the application's profile is loaded.

2 Start the YaST AppArmor Control Panel and put the application's profile into complain mode. Log entries are made for any actions violating the current profile, but the profile is not enforced and the application's behavior not restricted.

3 Run the application covering all the tasks you need this application to be able to perform.

4 Start the YaST Update Profile Wizard to update the application's profile according to the log entries generated while running the application.

5 Once the profile is updated, put it back into enforce mode via the YaST AppArmor Control Panel.

Using the AppArmor command line tools, you would proceed as follows:

1 Put the application's profile into complain mode:

```
aa-complain /path/to/application
```

2 Run the application.

3 Update the profile according to the log entries made while running the application:

```
aa-logprof /path/to/application
```

4 Put the resulting profile back into enforce mode:

```
aa-enforce /path/to/application
```

7.4.3 How to Confine KDE Applications with AppArmor?

Currently, it is not possible to confine KDE applications to the same extent as any other application due to the way KDE manages its processes.

If you want to confine KDE applications, choose one of the following approaches, but note that none of them is really suited for a standard setup:

Create a Single Profile for the Entire KDE Desktop
As all KDE processes are children of one parent process and AppArmor cannot distinguish an individual application's process from the rest, create one huge profile to confine the entire desktop all at once. This approach is only feasible if your setup is a very limited (kiosk-type) one. Maintaining such a profile for a standard KDE desktop including all of its applications would be close to impossible.

Modify KDE's process handling
Use `KDE_EXEC_SLAVES=1` and `KDE_IS_PRELINKED=1` variables force KDE to manage its processes in a way that AppArmor can distinguish individual applications from each other and apply profiles to them. This approach might slow down your desktop considerably, as it turns off a crucial optimization for speed. Note that the above mentioned environment variables have to be set before KDM/XDM/GDM or startx are started. One way to achieve this would be to add them to `/etc/security/pam_env.conf`.

7.4.4 How to Resolve Issues with Apache?

Apache is not starting properly or it is not serving Web pages and you just installed a new module or made a configuration change. When you install additional Apache modules (like `apache2-mod_apparmor`) or make configuration changes to Apache, you should profile Apache again to catch any additional rules that need to be added to the profile.

7.4.5 Why are the Reports not Sent by E-Mail?

When the reporting feature generates an HTML or CSV file that exceeds the default size, the file is not sent. Mail servers have a default, hard limit for e-mail size. This limitation can impede AppArmor's ability to send e-mails that are generated for reporting purposes. If your mail is not arriving, this could be why. Consider the mail size limits and check the archives if e-mails have not been received.

7.4.6 How to Exclude Certain Profiles from the List of Profiles Used?

AppArmor always loads and applies all profiles that are available in its profile directory (`/etc/apparmor.d/`). If you decide not to apply a profile to a certain application, delete the appropriate profile or move it to another location where AppArmor would not check for it.

7.4.7 Can I Manage Profiles for Applications not Installed on my System?

Managing profiles with AppArmor requires you to have access to a the system's log the application is running on. So you do not need to run the application on your profile build host as long as you have access to the machine that runs the application. You can run the application on one system, transfer the logs (`/var/log/audit.log` or, if `audit` is not installed, `/var/log/messages`) to your profile build host and run `aa-logprof -f` *path_to_logfile*.

7.4.8 How to Spot and fix AppArmor Syntax Errors?

Manually editing Novell AppArmor profiles can introduce syntax errors. If you attempt to start or restart AppArmor with syntax errors in your profiles, error results are shown. This example shows the syntax of the entire parser error.

```
localhost:~ # rcapparmor start
Loading AppArmor profiles
AppArmor parser error, line 2: Found unexpected character: 'h'
Profile /etc/apparmor.d/usr.sbin.squid failed to load
failed
```

Using the AppArmor YaST tools, a graphical error message indicates which profile contained the error and requests you to fix it.
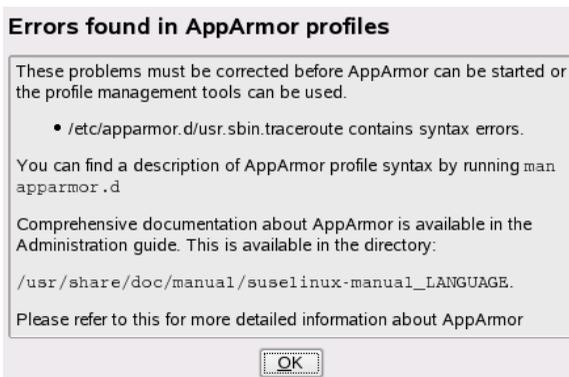
To fix a syntax error, log in to a terminal window as `root`, open the profile, and correct the syntax. Reload the profile set with `rcapparmor reload`.

7.5 Reporting Bugs for AppArmor

The developers of AppArmor are eager to deliver products of the highest quality. Your feedback and your bug reports help us keep the quality high. Whenever you encounter a bug in AppArmor, file a bug report against this product:

1 Use your Web browser to go to `https://bugzilla.novell.com/index.cgi`.

2 Enter the account data of your Novell account and click *Login*

   or

   Create a new Novell account as follows:

   2a Click *Create New Account* on the *Login to Continue* page.

   2b Provide a username and password and additional address data and click *Create Login* to immediately proceed with the login creation.

      or

      Provide data on which other Novell accounts you maintain to sync all these to one account.

3 Check whether a problem similar to yours has already been reported by clicking *Search Reports*. Use a quick search against a given product and keyword or use the *Advanced Search*.

4 If your problem has already been reported, check this bug report and add extra information to it, if necessary.

5 If your problem has not been reported yet, select *New* from the top navigation bar and proceed to the *Enter Bug* page.

6 Select the product against which to file the bug. In your case, this would be your product's release. Click *Submit*.

7 Select the product version, component (AppArmor in this case), hardware platform, and severity.

8 Enter a brief headline describing your problem and add a more elaborate description including log files. You may create attachments to your bug report for screen shots, log files, or test cases.

9 Click *Submit* after you have entered all the details to send your report to the developers.

Background Information on AppArmor Profiling

A

For more information about the science and security of Novell AppArmor, refer to the following papers:

SubDomain: Parsimonious Server Security by Crispin Cowan, Steve Beattie, Greg Kroah-Hartman, Calton Pu, Perry Wagle, and Virgil Gligor
> Describes the initial design and implementation of Novell AppArmor. Published in the proceedings of the USENIX LISA Conference, December 2000, New Orleans, LA. This paper is now out of date, describing syntax and features that are different from the current Novell AppArmor product. This paper should be used only for scientific background and not for technical documentation.

Defcon Capture the Flag: Defending Vulnerable Code from Intense Attack by Crispin Cowan, Seth Arnold, Steve Beattie, Chris Wright, and John Viega
> A good guide to strategic and tactical use of Novell AppArmor to solve severe security problems in a very short period of time. Published in the Proceedings of the DARPA Information Survivability Conference and Expo (DISCEX III), April 2003, Washington, DC.

AppArmor for Geeks by Seth Arnold
> This document tries to convey a better understanding of the technical details of AppArmor. It is available at http://en.opensuse.org/AppArmor_Geeks.

AppArmor Technical Documentation by Andreas Gruenbacher and Seth Arnold
> This document discusses the concept and design of AppArmor from a very technical point of view. It is available at http://forgeftp.novell.com//apparmor/LKML_Submission-June-07/techdoc.html.

B

GNU Licenses

This appendix contains the GNU General Public License and the GNU Free Documentation License.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does. Copyright (C) yyyy name of author

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.


This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.


You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place – Suite 330, Boston, MA 02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.


signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License [http://www.fsf.org/licenses/lgpl.html] instead of this License.

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
    Copyright (c) YEAR YOUR NAME.
```

```
    Permission is granted to copy, distribute and/or modify this document
    under the terms of the GNU Free Documentation License, Version 1.2
    only as published by the Free Software Foundation;
    with the Invariant Section being this copyright notice and license.
    A copy of the license is included in the section entitled "GNU
    Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
    with the Invariant Sections being LIST THEIR TITLES, with the
    Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Glossary

Apache

Apache is a freely available UNIX-based Web server. It is currently the most commonly used Web server on the Internet. Find more information about Apache at the Apache Web site at http://www.apache.org.

application firewalling

Novell AppArmor contains applications and limits the actions they are permitted to take. It uses privilege confinement to prevent attackers from using malicious programs on the protected server and even using trusted applications in unintended ways.

attack signature

Pattern in system or network activity that signals a possible virus or hacker attack. Intrusion detection systems might use attack signatures to distinguish between legitimate and potentially malicious activity.

By not relying on attack signatures, Novell AppArmor provides "proactive" instead of "reactive" defense from attacks. This is better because there is no window of vulnerability where the attack signature must be defined for Novell AppArmor as it does for products using attack signatures to secure their networks.

GUI

Graphical user interface. Refers to a software front-end meant to provide an attractive and easy-to-use interface between a computer user and application. Its elements include such things as windows, icons, buttons, cursors, and scroll bars.

globbing

Filename substitution.

HIP

Host intrusion prevention. Works with the operating system kernel to block abnormal application behavior in the expectation that the abnormal behavior represents an unknown attack. Blocks malicious packets on the host at the network level before they can "hurt" the application they target.

mandatory access control

A means of restricting access to objects that is based on fixed security attributes assigned to users, files, and other objects. The controls are mandatory in the sense that they cannot be modified by users or their programs.

profile foundation classes

Profile building blocks needed for common application activities, such as DNS lookup and user authentication.

RPM

The RPM Package Manager. An open packaging system available for anyone to use. It works on Red Hat Linux, openSUSE, and other Linux and UNIX systems. It is capable of installing, uninstalling, verifying, querying, and updating computer software packages. See http://www.rpm.org/ for more information.

SSH

Secure Shell. A service that allows you to access your server from a remote computer and issue text commands through a secure connection.

streamlined access control

Novell AppArmor provides streamlined access control for network services by specifying which files each program is allowed to read, write, and execute. This ensures that each program does what it is supposed to do and nothing else.

URI

Universal resource identifier. The generic term for all types of names and addresses that refer to objects on the World Wide Web. A URL is one kind of URI.

URL

Uniform Resource Locator. The global address of documents and other resources on the World Wide Web.

The first part of the address indicates what protocol to use and the second part specifies the IP address or the domain name where the resource is located.

For example, in http://www.novell.com, http is the protocol to use.

vulnerabilities

An aspect of a system or network that leaves it open to attack. Characteristics of computer systems that allow an individual to keep it from correctly operating or that allows unauthorized users to take control of the system. Design, administrative,

or implementation weaknesses or flaws in hardware, firmware, or software. If exploited, a vulnerability could lead to an unacceptable impact in the form of unauthorized access to information or disruption of critical processing.