Novell AppArmor Powered by Immunix Administration Guide

www.novell.com

1.2 | 09/29/2005

Novell AppArmor Powered by Immunix 1.2 Administration Guide

Legal Notices

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

Introduction to Novell AppArmor

Novell® AppArmor Powered by Immunix 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 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.
- The Novell AppArmor–loadable kernel module and associated control scripts to enforce AppArmor policies on your SUSE® Linux system.

NOTE

Some distributions of SUSE Linux include a version of AppArmor that enforce policies for a limited set of programs. These policies can be modified to suit your particular environment using the included AppArmor tool set. To create AppArmor profiles for additional programs, an upgrade to the full version of AppArmor is required.

---

1 Documentation Conventions

The following typographical conventions are used in this manual:

Menu Items, Field Names, and Screen Titles in GUIs
: When using GUIs, field names, menu and screen titles, and field values are shown as *File*.

Keys
: Key names are listed as they appear on your keyboard, as in Enter and Esc.

Command
: Linux commands (and other operating system commands, when used) are represented `this way`. This style should indicate to you that you can type the word or phrase on the command line and press Enter to run the command.

Example 1 *Command Environment*

To use `ls` to view the contents in the current directory, enter `ls` in a terminal window.

Filename
: Filenames, directory names, paths, and RPM package names are represented `this way`. This style should indicate that a particular file or directory exists by that name on your Linux system.

Placeholders
: Replace *`placeholder`* with the actual value that matches your setup.

Examples, Notes, and Warnings
: Examples use *Example:* when appropriate. Notes and pertinent information are shown with a *Note* or *Warning* flag, as in:

NOTE

Notes highlight information that might help better understand previous paragraphs. Warnings provide important information that might seriously affect the integrity of the product or your data.

Computer Output
: When you see text `in this style`, it indicates text displayed by the computer on the command line. You see responses to typed commands, error messages, and interactive prompts for your input during scripts or programs shown this way.

Example 2 *Computer Output*

Use the `ls` command to display the contents of a directory:

```
$ ls
Desktop  about.html       logs
Mail     backupfiles      mail
```

Trademarks
: A trademark symbol (®, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

2 Understanding This Guide

Immunizing Programs
: Describes operation of Novell AppArmor Powered by Immunix.

Selecting Programs to Immunize
: Describes the types of programs that should have Novell AppArmor profiles created for them.

Building Novell AppArmor Profiles
: Describes how to use the Novell AppArmor tools to immunize your own programs and third-party programs that you may have installed on your SUSE Linux system. It also helps you to add, edit, or delete profiles that have been created for your applications.

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

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

Support
Indicates support options for this product.

Glossary
Provides a list of terms and their definitions.

3 Getting Started with Novell AppArmor

Novell AppArmor Powered by Immunix (Novell AppArmor) provides you with technologies to protect your applications from their own vulnerabilities by creating Novell AppArmor profiles for applications on your SUSE Linux system.

3.1 Launching Novell AppArmor through the YaST GUI

SUSE Linux offers the utility YaST. Using YaST, you can launch the Novell AppArmor interface. This is the recommended method for a novice Linux user. For the other available methods, refer to Section 3.2, “Building and Managing Novell AppArmor Profiles” (page 24).

- To start YaST, select *System* → *Control Center (YaST)* from the SUSE menu.

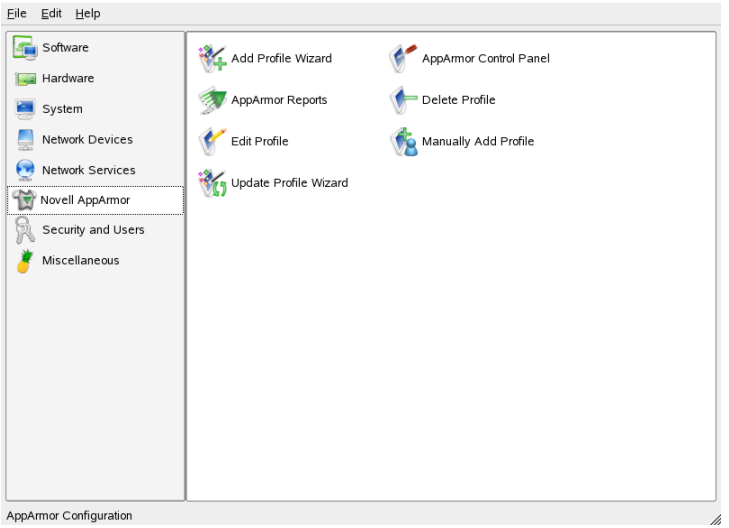
  YaST is launched as shown in Section 3.2, “Novell AppArmor Basics” (page x), below. You can refer to this section to navigate in Novell AppArmor.

NOTE

Alternately, you can launch the YaST GUI by opening a terminal window then entering `yast2` while logged in as root.

3.2 Novell AppArmor Basics

Novell AppArmor enables you to manage profiles through a simple user interface.



In the YaST Control Center, click *Novell AppArmor* in the left pane. The right from then shows the different Novell AppArmor configuration option. Select the appropriate Novell AppArmor configuration option by clicking the corresponding icon.

Depending on the configuration option you select, refer to one of the following locations in this guide:

Add Profile Wizard
: For detailed steps, refer to Section 3.3.1, “Adding a Profile Using the Wizard” (page 27).

AppArmor Reports
: For detailed steps, refer to Section 4.3, “Reports” (page 81).

Edit Profile
: Edit an existing Novell AppArmor profile on your system. For detailed steps, refer to Section 3.3.3, “Editing a Profile” (page 39).

Update Profile Wizard
: For detailed steps, refer to Section 3.3.5, “Updating Profiles from Syslog Entries” (page 42).

AppArmor Control Panel

For detailed steps, refer to Section 3.3.6, “Managing Novell AppArmor and Security Event Status” (page 47).

Delete Profile

Delete an existing Novell AppArmor profile from your system. For detailed steps, refer to Section 3.3.4, “Deleting a Profile” (page 41).

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 3.3.2, “Manually Adding a Profile” (page 34).

1 Immunizing Programs

Novell® AppArmor provides immunization technologies that protect SUSE Linux 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 you to the philosophy of immunizing programs. Proceed to Chapter 3, *Building Novell AppArmor Profiles* (page 21) 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. This ensures that each program does what it is supposed to do and nothing else.

Novell AppArmor is host intrusion prevention, or a mandatory access control scheme, that is optimized for servers. 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. Novell AppArmor controls the access given to network services and other programs to prevent weaknesses from being exploited.

2 Selecting Programs to Immunize

Novell® AppArmor quarantines programs to protect the rest of the system from being damaged by a compromised process. You should inspect your ports to see which programs should be profiled (refer to Section 2.2, “Inspect Open Ports to Immunize Programs” (page 16)) and profile all programs that grant privilege (Section 2.1, “Immunize Programs That Grant Privilege” (page 15)).

2.1 Immunize Programs That Grant Privilege

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/bin/updatedb` daily to keep the locate database up to date with sufficient privilege to read the name of every file in the system. For instructions for finding these types of programs, refer to Section 2.2.1, “Immunizing Cron Jobs” (page 18).

Web Applications
: Programs that can be invoked through a Web browser, including CGI Perl scripts, PHP pages, and more complex Web applications. For instructions on finding these

types of programs, refer to Section 2.2.2, “Immunizing Web Applications” (page 18).

Network Agents
: Programs (servers and clients) that have open network ports. User clients such as mail clients and Web browsers, surprisingly, mediate privilege. These programs run with the privilege to write to the user's home directories and they process input from potentially hostile remote sources, such as hostile Web sites and e-mailed malicious code. For instructions on finding these types of programs, refer to Section 2.2.3, “Immunizing Network Agents” (page 20).

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.

2.2 Inspect Open Ports to Immunize Programs

An automated method for finding network server daemons that should be profiled is to use the unconfined tool. You can also simply view a report of this information in the YaST GUI (refer to Section “Application Audit Report” (page 88) for instructions).

The 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. 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 for unconfined to detect and report the new profiled state.

Below is a sample 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

Unconfined requires root privileges and should not be run from a shell that is confined by an AppArmor profile.

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 unconfined tool detects and reports network ports opened by client applications, but only those client applications that are running at the time the 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, and Novell AppArmor is intended for servers rather than workstations. Therefore, we leave profiling of user network client applications as an exercise for the user.

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

Additional profiles can be traded with other users and with the Novell® security development team on the user mailing list at http://mail.wirex.com/mailman/listinfo/immunix-users.

2.2.1 Immunizing Cron Jobs

To find programs that are run by cron, you need to 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, you can 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 3.3.1, “Adding a Profile Using the Wizard” (page 27).

2.2.2 Immunizing Web Applications

To find Web applications, you should 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. SUSE Linux, 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 3.3.1, “Adding a Profile Using the Wizard” (page 27).

CGI Programs and Subprocess Confinement in Web Applications

Because CGI programs are executed by the Apache Web server, the profile for Apache itself `usr.sbin.httpd2-prefork` (for Apache2 on SUSE Linux) 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 ded-

icated 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 the 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 (`mod_change_hat`) 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 106).

Profiling Web applications that use `mod_perl` and `mod_php` require 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()` naming 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 syslog when Novell AppArmor complains about access violations. See Chapter 4, *Managing Profiled Applications* (page 77).

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/sbin/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:

```
/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,
```

If you want 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.

2.2.3 Immunizing Network Agents

To find network server daemons 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 netstat. Then inspect the machine to determine which programs are answering on the discovered open ports.

3 Building Novell AppArmor Profiles

This chapter explains how to build and manage Novell® AppArmor profiles. You are ready to build Novell AppArmor profiles after you select the programs to profile. For help with this, refer to Chapter 2, *Selecting Programs to Immunize* (page 15).

3.1 Profile Components and Syntax

This section details the syntax or makeup of Novell AppArmor profiles. An example illustrating this syntax is presented in Section 3.1.1, "Breaking a Novell AppArmor Profile into Its Parts" (page 21).

3.1.1 Breaking a Novell AppArmor Profile into Its Parts

Novell AppArmor profile components are called Novell AppArmor rules. Currently there are two main types of Novell AppArmor rules, path entries and capability 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. Consider, for example, the following profile for the program `/sbin/klogd`:

```
# profile to confine klogd❶
/sbin/klogd ❷
{❸
#include <abstractions/base>❹
  capability sys_admin,❺
  /boot/* r❻,
  /proc/kmsg r,
  /sbin/klogd r,
  /var/run/klogd.pid lw,
}
```

❶ A comment naming the program that is confined by this profile. Always precede comments like this with the # sign.

❷ The absolute path to the program that is confined.

❸ The curly braces { } serve as a container for include statements of other profiles as well as for path and capability entries.

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

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

❻ 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 white space of any kind (spaces or tabs) can precede pathnames or separate the pathname from the access modes. White space between the access mode and the trailing comma is optional.

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 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 syslog. 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.

3.1.2 #include

`#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, the `#include` statement appends `/etc/subdomain.d/`, which is where it expects to find the include file, to the beginning of the pathname. 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.

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). Abstractions can be found in `/etc/subdomain.d/abstractions/`.

Program Chunks

Program chunks are access controls for specific programs that a system administrator might want to control based on local site policy. Each chunk is used by a single program. These are provided to ease local-site modifications to policy and updates to policy provided by Novell AppArmor. Administrators can modify policy in these files to suit their own needs and leave the program profiles unmodified, simplifying the task of merging policy updates from Novell AppArmor into enforced policy at each site.

The access restrictions in the program chunks are typically very liberal and are designed to allow your users access to their files in the least intrusive way possible while still allowing system resources to be protected. An exception to this rule is the `postfix*` series of program chunks. These profiles are used to help abstract the location of the postfix binaries. You probably do not want to reduce the permissions in the `postfix*` series. Program chunks can be found in `/etc/subdomain.d/program-chunks/` .

3.1.3 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.

3.2 Building and Managing Novell AppArmor Profiles

There are three ways you can build and manage Novell AppArmor profiles, depending on the type of computer environment you prefer. You can use the graphical YaST interface (YaST GUI), the text-based YaST ncurses mode (YaST ncurses), or the command line interface. All three options are effective for creating and maintaining profiles while offering need-based options for users.

The command line interface requires knowledge of Linux commands and using terminal windows. All three methods use specialized Novell AppArmor tools for creating the profiles so you do not need to do it manually, which would be quite time consuming.

3.2.1 Using the YaST GUI

To use the YaST GUI for building and managing Novell AppArmor profiles, refer to Section 3.3, “Building Novell AppArmor Profiles with the YaST GUI” (page 26).

3.2.2 Using YaST ncurses

YaST ncurses can be used for building and managing Novell AppArmor profiles and is better suited for users with limited bandwidth connections to their server. Access YaST ncurses by typing `yast` while logged in to a terminal window or console as root. YaST ncurses has the same features as the YaST GUI.

Refer to the instructions in Section 3.3, “Building Novell AppArmor Profiles with the YaST GUI” (page 26) to build and manage Novell AppArmor profiles in YaST ncurses, but be aware that the screens look different but function similarly.

3.2.3 Using the Command Line Interface

The command line interface requires knowledge of Linux commands and using terminal windows. To use the command line interface for building and managing Novell AppArmor profiles, refer to Section 3.4, “Building Novell AppArmor Profiles Using the Command Line Interface” (page 49).

The command line interface offers access to a few tools that are not available using the other Novell AppArmor managing methods:

complain
: Sets profiles into complain mode. Set it back to enforce mode when you want the system to begin enforcing the rules of the profiles, not just logging information. For more information about this tool, refer to Section “Complain or Learning Mode” (page 58).

enforce
: Sets profiles back to enforce mode and the system begins enforcing the rules of the profiles, not just logging information. For more information about this tool, refer to Section “Enforce Mode” (page 59).

unconfined
: Performs a server audit to find processes that are running and listening for network connections then reports whether they are profiled.

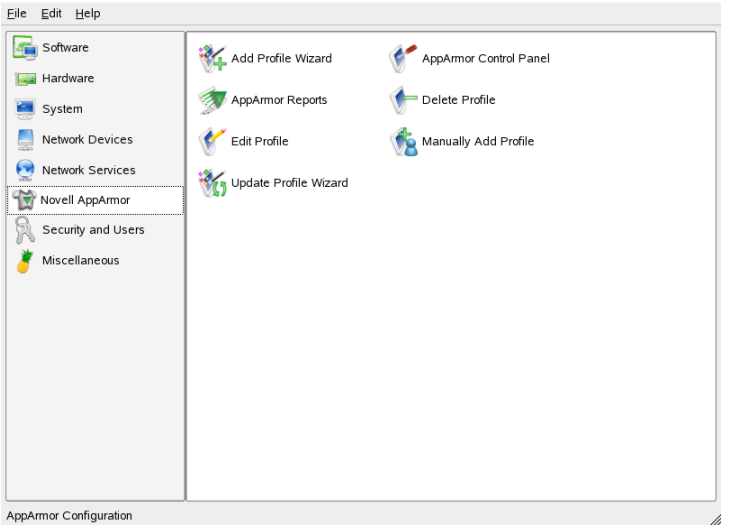
autodep
: Generates a profile skeleton for a program and loads it into the Novell AppArmor module in complain mode.

3.3 Building Novell AppArmor Profiles with the YaST GUI

Open the YaST GUI displays from the SUSE menu with *System* → *YaST* → *Novell AppArmor*. Novell AppArmor opens in the YaST interface as shown below:

NOTE

You can also access the YaST GUI by opening a terminal window, logging in as root, and entering `yast2`.



In the right frame, you see several Novell AppArmor option icons. If Novell AppArmor does not display in the left frame of the YaST window or if the Novell AppArmor icons do not display, you might want to reinstall Novell AppArmor. The following actions are available from Novell AppArmor.

Click one of the following Novell AppArmor icons and proceed to the section referenced below:

Add Profile Wizard
: For detailed steps, refer to Section 3.3.1, “Adding a Profile Using the Wizard” (page 27).

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 3.3.2, “Manually Adding a Profile” (page 34).

Edit Profile
: Edits an existing Novell AppArmor profile on your system. For detailed steps, refer to Section 3.3.3, “Editing a Profile” (page 39).

Delete Profile
: Deletes an existing Novell AppArmor profile from your system. For detailed steps, refer to Section 3.3.4, “Deleting a Profile” (page 41).

Update Profile Wizard
: For detailed steps, refer to Section 3.3.5, “Updating Profiles from Syslog Entries” (page 42).

AppArmor Reports
: For detailed steps, refer to Section 4.3, “Reports” (page 81).

AppArmor Control Panel
: For detailed steps, refer to Section 3.3.6, “Managing Novell AppArmor and Security Event Status” (page 47).

3.3.1 Adding a Profile Using the Wizard

The *Add Profile Wizard* is designed to set up Novell AppArmor profiles using the Novell AppArmor profiling tools, genprof (Generate Profile) and logprof (Update Profiles From Learning Mode Log File). For more information about these tools, refer to Section 3.5.3, “Summary of Profiling Tools” (page 56).

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

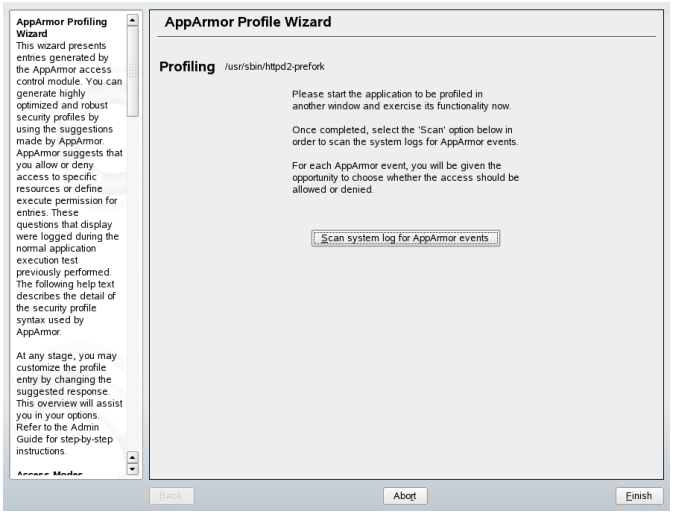
For example, enter `/etc/init.d/PROGRAM stop` in a terminal window while logged in as root, replacing *PROGRAM* is the name of the program to profile.

2 If you have not done so already, in the YaST GUI, click *Novell AppArmor* → *Add Profile Wizard*.

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

4 Click *Create*. This runs a Novell AppArmor tool named autodep, which performs a static analysis of the program to profile and loads an approximate profile into Novell AppArmor module. For more information about autodep, refer to Section “autodep” (page 57).

The *AppArmor Profiling Wizard* window opens.



In the background, Novell AppArmor also sets the profile to learning mode. For more information about learning mode, refer to Section “Complain or Learning Mode” (page 58).

5 Run the application that is being profiled.

6 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.

7 Click *Scan System Log for Entries to Add to Profile* 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.

NOTE

If requests to add hats appear, proceed to Chapter 5, *Profiling Your Web Applications Using ChangeHat Apache* (page 105).

The questions fall into two categories:

- A resource is requested by a profiled program that is not in the profile (see Figure 3.1, “Learning Mode Exception: Controlling Access to Specific Resources” (page 30)). The learning mode exception requires you to 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 3.2, “Learning Mode Exception: Defining Execute Permissions for an Entry” (page 31)). The learning mode exception requires you to 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 into the profile. The following two figures show an example of each case. Subsequent steps describe your options in answering these questions.

The *AppArmor Profiling Wizard* window opens.

Figure 3.1 *Learning Mode Exception: Controlling Access to Specific Resources*

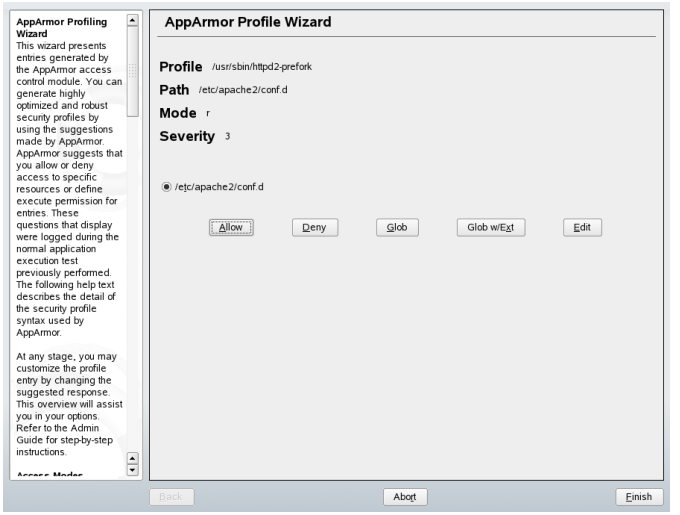
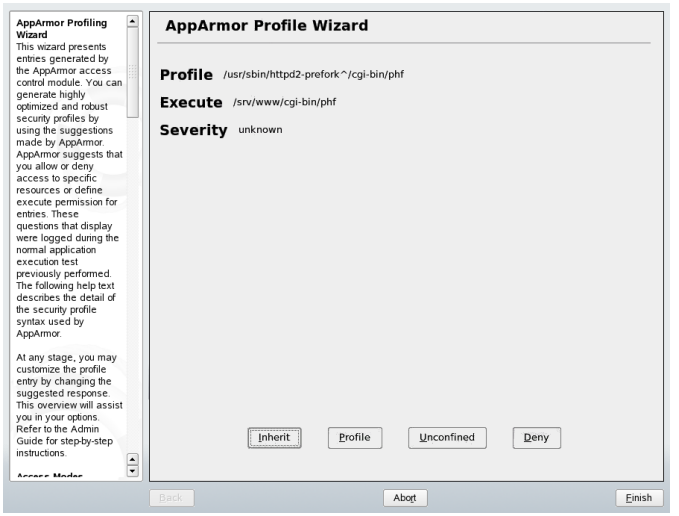


Figure 3.2 *Learning Mode Exception: Defining Execute Permissions for an Entry*



8 The *Add Profile Wizard* begins suggesting directory path entries that have been accessed by the application you are profiling (as seen in Figure 3.1, “Learning Mode Exception: Controlling Access to Specific Resources” (page 30)) or requiring you to define execute permissions for entries (as seen in Figure 3.2, “Learning Mode Exception: Defining Execute Permissions for an Entry” (page 31)).

a For Figure 3.1, “Learning Mode Exception: Controlling Access to Specific Resources”: From the following options, select the one that satisfies the request for access, which could be a suggested include, a particular globbed version of the path, or the actual pathname. Note that all of these options are not always available.

`#include`

The section of a Novell AppArmor profile that refers to an include file. Include files procure 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* as described in the next step. For information about globbing syntax, refer to Section 3.6, “Pathnames and Globbing” (page 73).

Actual Pathname
: Literal path that the program needs access to so that it can run properly.

b For Figure 3.2, “Learning Mode Exception: Defining Execute Permissions for an Entry”: From the following options, select the one that satisfies the request for access.

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

Profile
: Requires that a separate profile exists for the executed program.

Unconfined
: Executes the program without a security profile.

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

9 After you select a directory path, you need to process it as an entry into the Novell AppArmor 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 to build the profile:

Allow
: Grants 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 3.7, “File Permission Access Modes” (page 74).

Deny
: Click *Deny* to prevent the program from accessing the specified directory path entries.

Glob
: Clicking this modifies the directory path (by using wild cards) to include all files in the suggested entry directory. Double-clicking it grants access to all files and subdirectories beneath the one shown.

  For more information about globbing syntax, refer to Section 3.6, "Pathnames and Globbing" (page 73).

Glob w/Ext
: Modifies 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 file name. 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
: Enables editing of the highlighted line. The new (edited) line appears at the bottom of the list.

Abort
: Aborts logprof, dumping all rule changes entered so far and leaving all profiles unmodified.

Finish
: Closes 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.

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

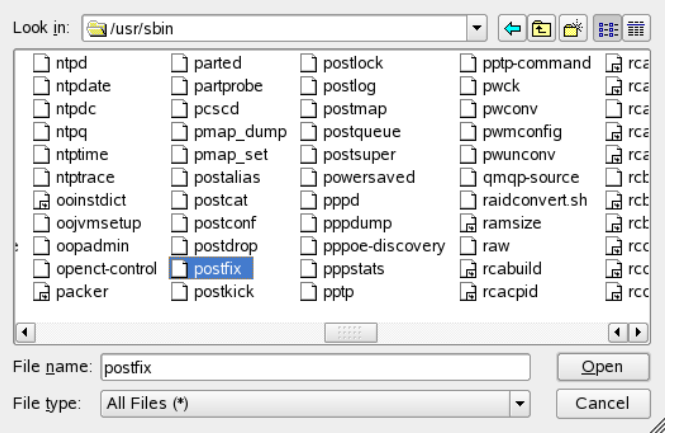
When you are done, click *Finish*. In the following pop-up, click *Yes* to exit the *Profile Creation Wizard*. The profile is saved and loaded into the Novell AppArmor module.

3.3.2 Manually Adding a Profile

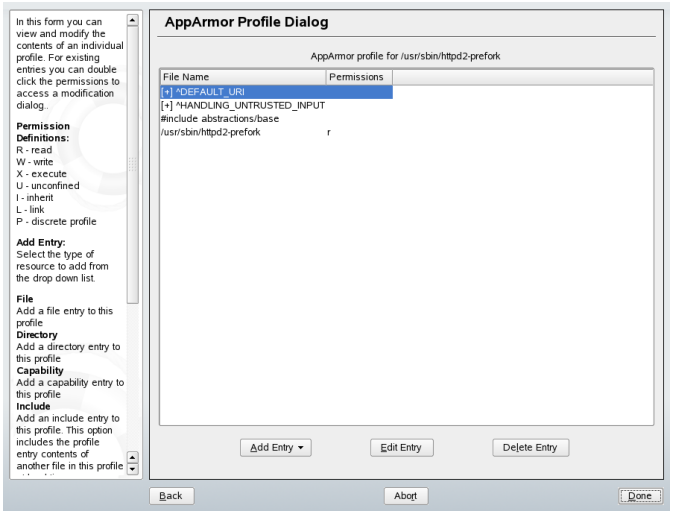
Novell AppArmor enables you to create a Novell AppArmor profile by manually adding entries into the profile. You simply need to select the application for which to create a profile, then add entries.

1. To add a profile, open *YaST* → *Novell AppArmor*. The Novell AppArmor interface opens.
2. In *Novell AppArmor*, click *Manually Add Profile* (see Figure 3.3, "Manually Adding a Profile: Select Application" (page 34)).

Figure 3.3 *Manually Adding a Profile: Select Application*



3. Browse your system to find the application for which to create a profile.
4. When you find the profile, select it and click *Open*. A basic, empty profile appears in the *Novell AppArmor Profile Dialog* window.



5 In the *AppArmor Profile Dialog* window, you can add, edit, or delete Novell AppArmor profile entries by clicking the corresponding buttons and referring to the following sections: Section “Adding an Entry” (page 35), Section “Editing an Entry” (page 38), or Section “Editing an Entry” (page 38).

6 When you are finished, click *Done*.

Adding an Entry

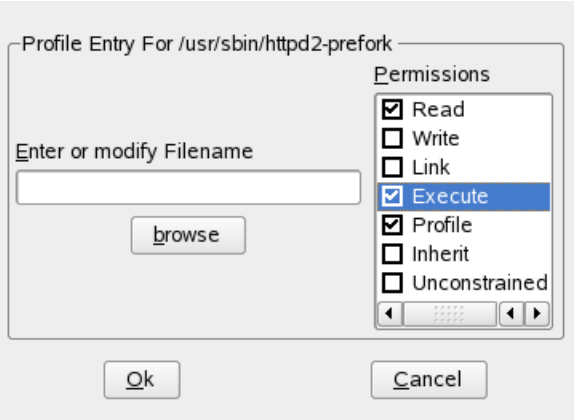
This section explains the *Add Entry* option that can be found in Section 3.3.2, “Manually Adding a Profile” (page 34) or Section 3.3.3, “Editing a Profile” (page 39). When you select *Add Entry*, a drop-down list displays the types of entries you can add to the Novell AppArmor profile.

- 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*.

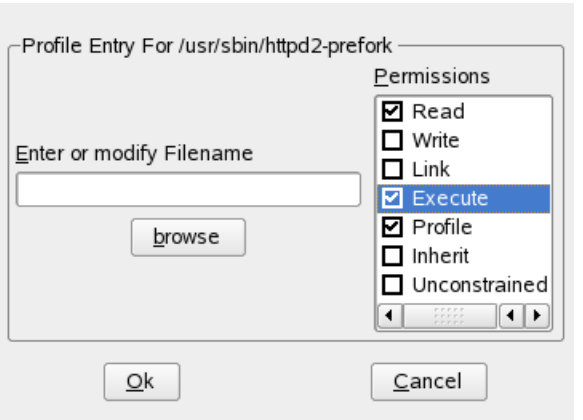
You can use globbing if necessary. For globbing information, refer to Section 3.6, “Pathnames and Globbing” (page 73). For file access permission information, refer to Section 3.7, “File Permission Access Modes” (page 74).



Directory

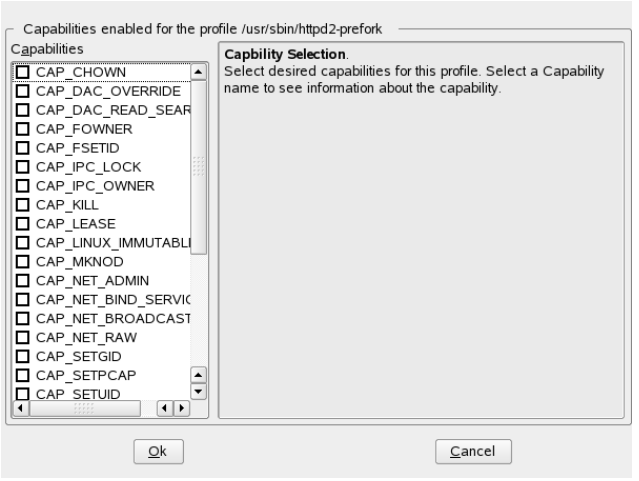
In the pop-up window, specify the absolute path of a directory, including the type of access permitted. You can use globbing if necessary. When finished, click *OK*.

For globbing information, refer to Section 3.6, “Pathnames and Globbing” (page 73). For file access permission information, refer to Section 3.7, “File Permission Access Modes” (page 74).



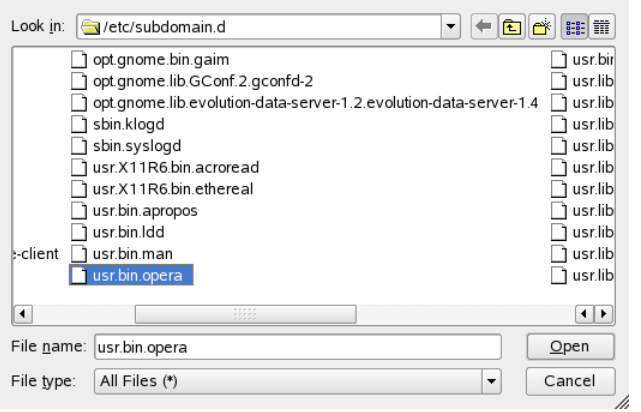
Capability

In the pop-up window, select the appropriate capabilities. These are statements that enable each of the 32 POSIX.1e capabilities. Refer to Section 3.1.1, “Breaking a Novell AppArmor Profile into Its Parts” (page 21) 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 3.1.2, “`#include`” (page 23).

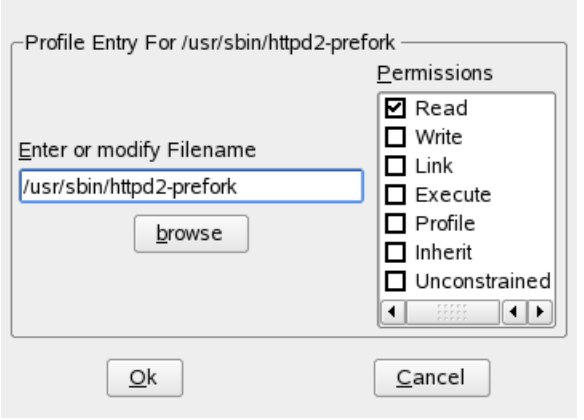


Editing an Entry

This section explains the *Edit Entry* option that can be found in Section 3.3.2, “Manually Adding a Profile” (page 34) or Section 3.3.3, “Editing a Profile” (page 39). When you select *Edit Entry*, the file browser pop-up window opens. From here, you can 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 3.6, “Pathnames and Globbing” (page 73). For file access permission information, refer to Section 3.7, “File Permission Access Modes” (page 74).



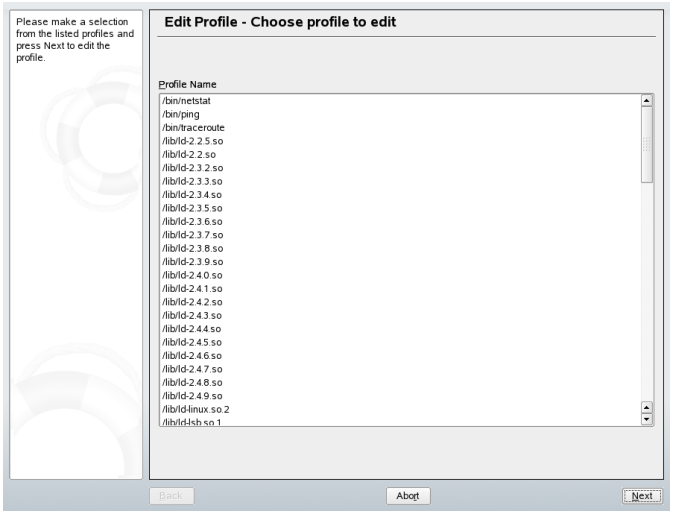
Deleting an Entry

This section explains the *Delete Entry* option that can be found in the Section 3.3.2, “Manually Adding a Profile” (page 34) or Section 3.3.3, “Editing a Profile” (page 39). When you select an entry then select *Delete Entry*, Novell AppArmor removes the profile entry that you have selected.

3.3.3 Editing a Profile

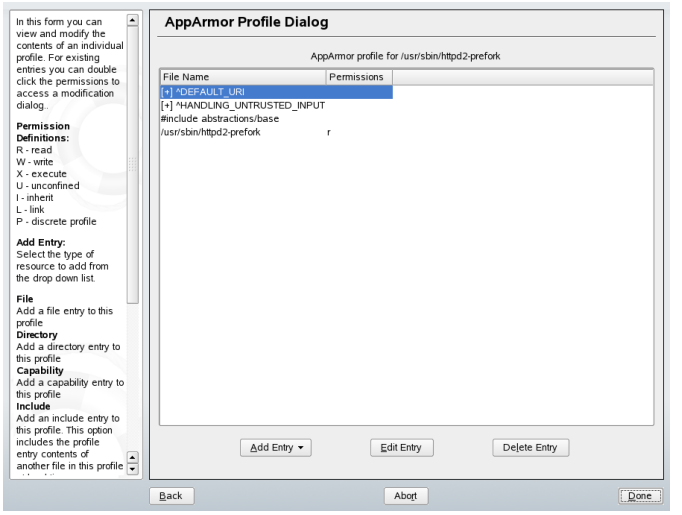
Novell AppArmor enables you to manually edit Novell AppArmor profiles by adding, editing, or deleting entries. You simply need to select the profile then add, edit, or delete entries. To edit a profile, follow these steps:

1. Open *YaST* → *Novell AppArmor*.
2. In *Novell AppArmor*, click *Edit Profile*. The *Edit Profile—Choose Profile to Edit* window opens.



3 From the list of profiled programs, select the profile to edit.

4 Click *Next*. The *AppArmor Profile Dialog* window displays the profile.



5 In the *AppArmor Profile Dialog* window, you can add, edit, or delete Novell AppArmor profile entries by clicking the corresponding buttons and referring to the following sections: Section “Adding an Entry” (page 35), Section “Editing an Entry” (page 38), or Section “Deleting an Entry” (page 39).

6 When you are finished, click *Done*.

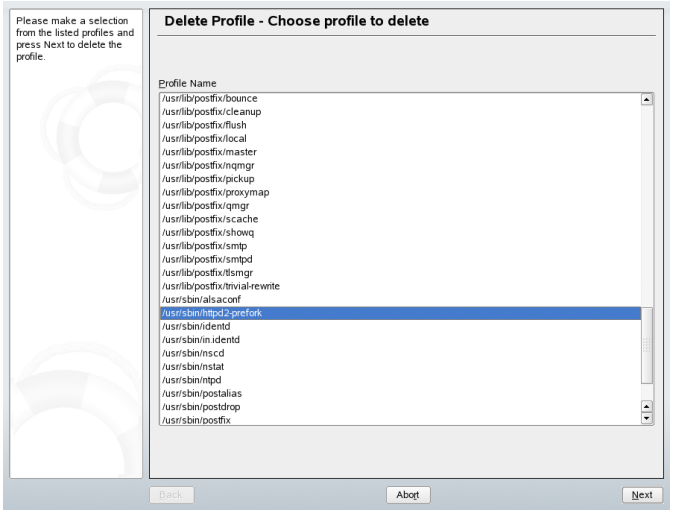
7 In the pop-up that appears, click *Yes* to confirm your changes to the profile.

3.3.4 Deleting a Profile

Novell AppArmor enables you to delete a Novell AppArmor profile manually. You simply need to select the application for which to delete a profile then delete it as follows:

1 Open the *YaST* → *Novell AppArmor*. The Novell AppArmor interface displays.

2 In *Novell AppArmor*, click *Delete Profile* icon. The *Delete Profile—Choose Profile to Delete* window opens.



3 Select the profile to delete.

4 Click *Next*.

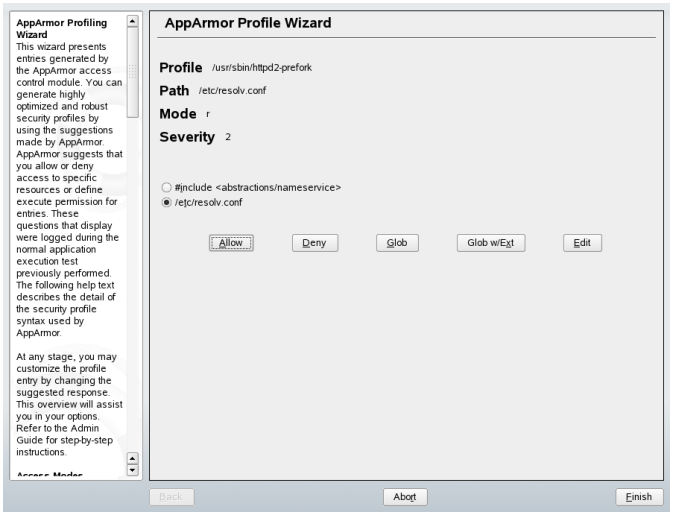
5 In the pop-up that opens, click *Yes* to delete the profile.

3.3.5 Updating Profiles from Syslog Entries

The Novell AppArmor Profile wizard uses logprof, the tool that scans log files and enables you to update profiles. 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.

1 Open *YaST → Novell AppArmor*. The Novell AppArmor interface displays.

2 In *Novell AppArmor*, click *Update Profile Wizard*. The *AppArmor Profile Wizard* window displays.



Running the *Update Profile Wizard* (logprof) parses the learning mode log files. This generates a series of questions that you must answer to guide logprof to generate the security profile.

The questions fall into two categories:

- A resource is requested by a profiled program that is not in the profile (see Figure 3.4, “Learning Mode Exception: Controlling Access to Specific Resources” (page 43)).

- A program is executed by the profiled program and the security domain transition has not been defined (see Figure 3.5, “Learning Mode Exception: Defining Execute Permissions for an Entry” (page 44)).

Each of these cases results in a question that you must answer that enables you to add the resource or program into the profile. The following two figures show an example of each case. Subsequent steps describe your options in answering these questions.

Figure 3.4 *Learning Mode Exception: Controlling Access to Specific Resources*

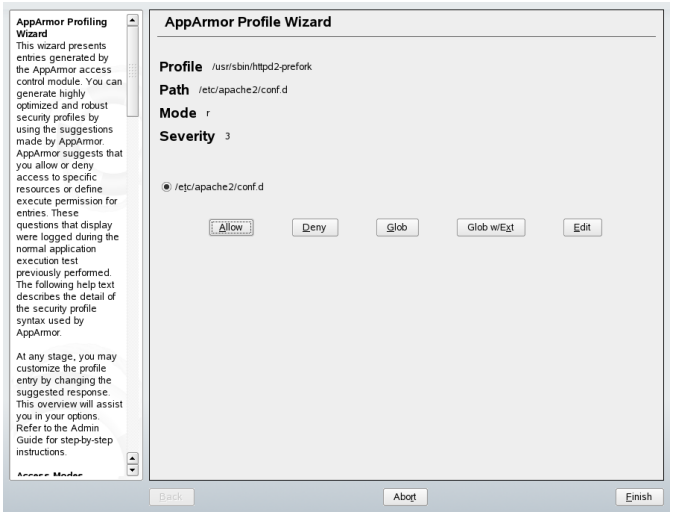
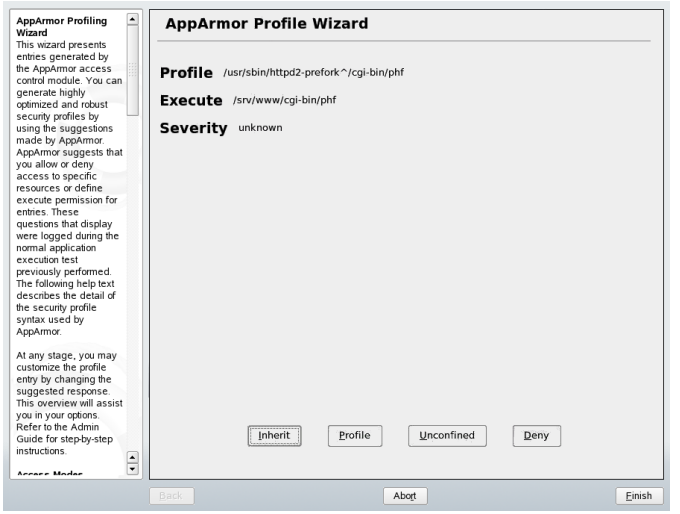


Figure 3.5 *Learning Mode Exception: Defining Execute Permissions for an Entry*



3 logprof begins suggesting directory path entries that have been accessed by the application you are profiling (as seen in Figure 3.4, “Learning Mode Exception: Controlling Access to Specific Resources” (page 43)) or requiring you to define execute permissions for entries (as seen in Figure 3.5, “Learning Mode Exception: Defining Execute Permissions for an Entry” (page 44)).

   a For Figure 3.4, “Learning Mode Exception: Controlling Access to Specific Resources” (page 43): From the following options, select the one that satisfies the request for access, which could be a suggested include, a particular globbed version of the path, or the actual pathname. Note that all of these options are not always available.

      `#include`

      The section of a Novell AppArmor profile that refers to an include file. 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. It is good practice to select includes when suggested.

Globbed Version
: Accessed by clicking *Glob* as described in the next step. For information about globbing syntax, refer to Section 3.6, “Pathnames and Globbing” (page 73).

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

b For Figure 3.5, “Learning Mode Exception: Defining Execute Permissions for an Entry” (page 44): Select the one that satisfies the request for access by choosing one of the following:

Inherit
: stay in the same security profile (parent's profile)

Profile
: requires that a separate profile exists for the executed program

Unconfined
: program executed without a security profile

WARNING

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

---

4 After you select a directory path, you need to process it as an entry into the Novell AppArmor 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 to build the profile:

Allow
: Grant the program access to the specified directory path entries. The *Profile Creation Wizard* suggests file permission access. For more information about this, refer to Section 3.7, “File Permission Access Modes” (page 74).

Deny
: Click *Deny* to prevent the program from accessing the specified directory path entries.

Glob
: Clicking this modifies the directory path (by using wild cards) to include all files in the suggested entry directory. Double-clicking it grants access to all files and subdirectories beneath the one shown.

  For more information about globbing syntax, refer to Section 3.6, “Pathnames and Globbing” (page 73).

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
: Enable editing of the highlighted line. The new (edited) line appears at the bottom of the list.

Abort
: Abort logprof, dumping all rule changes entered so far and leaving all profiles unmodified.

Finish
: Close 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.

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

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

3.3.6 Managing Novell AppArmor and Security Event Status

Novell AppArmor enables you to change the status of Novell AppArmor and configure event notification.

Changing Novell AppArmor Status
: You can change the status of Novell AppArmor by enabling or disabling it. Enabling Novell AppArmor protects your system from potential program exploitation. Disabling Novell AppArmor, even if your profiles have been set up, removes protection from your system.

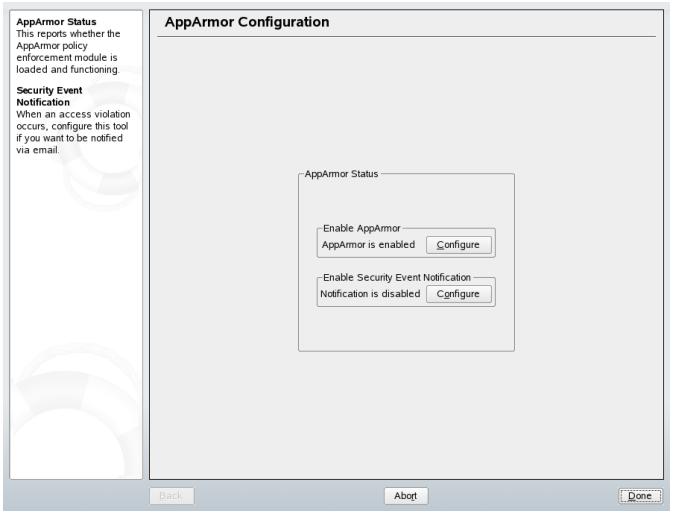
Configuring Event Notification
: You can determine how and when you are notified when system security events occur.

NOTE

You must set up a mail server on your SUSE Linux server that can send outgoing mail using the single mail transfer protocol (smtp). For example, postfix or exim, in order for event notification to work.

To either configure event notification or change the status of Novell AppArmor, perform the following steps:

1 When you click *Novell AppArmor Control Panel*, the *Novell AppArmor Configuration* window appears as shown below:



2 From the *AppArmor Configuration* screen, determine whether Novell AppArmor and security event notification are running by looking for a status message that reads *enabled*.

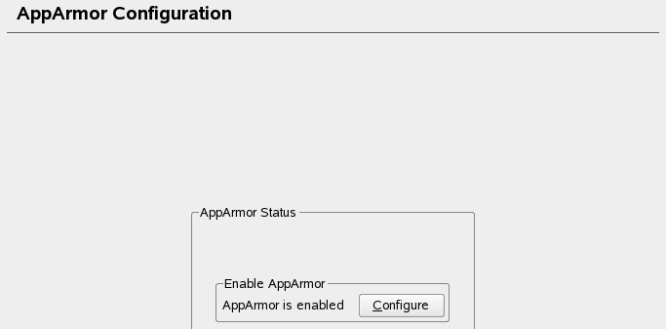
- To change the status of Novell AppArmor, continue as described in Section "Changing Novell AppArmor Status" (page 48).
- To configure security event notification, continue as described in Section 4.2.2, "Configuring Security Event Notification" (page 79).

Changing Novell AppArmor Status

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

1 To enable Novell AppArmor, open *YaST* → *Novell AppArmor*. The Novell AppArmor main menu opens.

2 In the *Novell AppArmor* main menu, click *AppArmor Control Panel*. The *AppArmor Configuration* window appears.



3 In the *Enable Novell AppArmor* section of the window, click *Configure*. The *Enable Novell AppArmor* dialog box opens.

4 Enable Novell AppArmor by selecting *Enable* or disable Novell AppArmor by selecting *Disable*. Then click *OK*.

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

6 Click *File* → *Quit* in the YaST Control Center.

3.4 Building Novell AppArmor Profiles Using the Command Line Interface

Novell AppArmor provides the ability to use a command line interface rather than the GUI to manage and configure your system security.

3.4.1 Checking the SubDomain Module Status

The SubDomain module can be in any one of three states:

Unloaded
: The SubDomain module is not loaded into the kernel.

Running
: The SubDomain module is loaded into the kernel and is enforcing Novell AppArmor program policies.

Stopped
: The SubDomain module is loaded into the kernel, but there are no policies being enforced.

You can detect which of the three states that the SubDomain module is in by inspecting `/subdomain/profiles`. If `cat /subdomain/profiles` reports a list of profiles, Novell AppArmor is running. If it is empty and returns nothing, SubDomain is stopped. If the file does not exist, SubDomain is unloaded.

The SubDomain module can be loaded and unloaded with the standard Linux module commands such as `modprobe`, `insmod`, `lsmod`, and `rmmod`, but this approach is not recommended. Instead, it is recommended to manage Novell AppArmor through the script `rcsubdomain`, which can perform the following operations:

`rcsubdomain start`
: Has different behaviors depending on the SubDomain module state. If it was unloaded, `start` loads the module and starts it, putting it in the running state. If it was stopped, then `start` causes the module to rescan the Novell AppArmor profiles usually found in `/etc/subdomain.d` and puts the module in the running state. If the module was already running, `start` reports a warning and takes no action.

`rcsubdomain stop`
: Stops SubDomain module (if it was running) by removing all profiles from kernel memory, effectively disabling all access controls, putting the module into the stopped state. If the SubDomain module was either unloaded or already stopped, `stop` tries to unload the profiles again, but nothing happens.

`rcsubdomain restart`

Causes SubDomain module to rescan the profiles usually found in `/etc/subdomain.d` without unconfining running processes, adding new profiles, and removing any profiles that had been deleted from `/etc/subdomain.d`.

`rcsubdomain kill`

Unconditionally removes the SubDomain 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 have to be removed.

NOTE

Novell AppArmor is a powerful access control system and it is possible to lock yourself out of your own machine to the point where you have to boot the machine from rescue media (such as CD 1 of SUSE Linux) 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 SubDomain 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 and restarting the SubDomain module.

3.4.2 Building Novell AppArmor Profiles

The SubDomain module profile definitions are stored in the directory `/etc/subdomain.d/` as plain text files.

WARNING

All files in the `/etc/subdomain.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 manage them 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 Novell AppArmor Profiles
: Refer to Section 3.4.3, “Adding or Creating a Novell AppArmor Profile” (page 52)

Editing Novell AppArmor Profiles
: Refer to Section 3.4.4, “Editing a Novell AppArmor Profile” (page 53)

Deleting Novell AppArmor Profiles
: Refer to Section 3.4.5, “Deleting a Novell AppArmor Profile” (page 53)

Use vim to view and edit your profile by typing vim at a terminal window. To enable syntax coloring when you edit a Novell AppArmor profile in vim, use the commands `:syntax on` then `:set syntax=subdomain`. For more information about vim and syntax coloring, refer to Section “Subdomain.vim” (page 71).

NOTE

After making changes to a profile, use the `rcsubdomain restart` command, described in the previous section. This command causes the Novell AppArmor to reread the profiles. For a detailed description of the syntax of these files, refer to Chapter 3, *Building Novell AppArmor Profiles* (page 21).

3.4.3 Adding or Creating a Novell AppArmor Profile

To add or ceate a Novell AppArmor profile for an application, you can use a systemic or stand-alone profiling method, depending on your needs.

Stand-Alone Profiling
: Suitable for profiling small applications that have a finite run time, such as user client applications like mail clients. Refer to Section 3.5.1, “Stand-Alone Profiling” (page 54).

Systemic Profiling
: Suitable for profiling large numbers of programs all at once and for profiling applications that might run for days, weeks, or continuously across reboots, such as

network server applications like Web servers and mail servers. Section 3.5.2, “Systemic Profiling” (page 55).

3.4.4 Editing a Novell AppArmor Profile

The following steps describe the procedure for editing a Novell AppArmor profile. To better understand what makes up a profile, refer to Section 3.1, “Profile Components and Syntax” (page 21).

1. If you are not currently signed in as root, type `su` in a terminal window.
2. Enter the root password when prompted.
3. To go to the directory, enter `cd /etc/subdomain.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 Novell AppArmor by entering `rcsubdomain restart` in a terminal window.

3.4.5 Deleting a Novell AppArmor Profile

The following steps describe the procedure for deleting a Novell AppArmor profile.

1. If you are not currently signed in as root, enter `su` in a terminal window.
2. Enter the root password when prompted.
3. To go to the Novell AppArmor directory, enter `cd /etc/subdomain.d/`.
4. Enter `ls` to view all the Novell AppArmor profiles that are currently installed.
5. Delete the profile exiting profile with `rm` *`profilename`*.

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

3.5 Two Methods of Profiling

Given the syntax for Novell AppArmor profiles in Section 3.1, "Profile Components and Syntax" (page 21), you could create profiles without using the tools. However, the effort involved would be substantial. To avoid such a hassle, use the Novell AppArmor tools to automate the creation and refinement of profiles.

There are two ways to approach creating Novell AppArmor profiles, along with tools to support 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 3.5.1, "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 3.5.2, "Systemic Profiling" (page 55).

Automated profile development becomes more manageable with the Novell AppArmor tools:

1 Decide which profiling method suits your needs.

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

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

3.5.1 Stand-Alone Profiling

Stand-alone profile generation and improvement is managed by a program called genprof. This method is easy because genprof takes care of everything, but is limited because

it requires 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 genprof for the stand-alone method of profiling, refer to Section “genprof” (page 60).

3.5.2 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 being targeted by genprof or *standalone profiling*.

With systemic profiling, building and improving profiles are somewhat less automated, but more flexible. This method is suitable for profiling long-running applications whose behavior continues after rebooting or a large numbers of programs to profile all at once.

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

1. **Create profiles for the individual programs that make up your application.** Even though this approach is systemic, Novell AppArmor still only monitors those programs with profiles and their children. Thus, to get Novell AppArmor to consider a program, you must at least have autodep create an approximate profile for it. To create this approximate profile, refer to Section “autodep” (page 57).

2. **Put relevant profiles into learning or complain mode.** Activate learning or complain mode for all profiled programs by entering `complain /etc/subdomain.d/*` in a terminal window while logged in as root.

   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 “Complain or Learning Mode” (page 58) 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 genprof, this step can go on for days or weeks and can span complete system reboots.

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

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

Refer to Section “logprof” (page 65) for more information about using logprof.

5 **Repeat Steps 3-4.** 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/subdomain.d/` using vim. For help using vim to its fullest capacity, refer to Section “Subdomain.vim” (page 71).

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 enforce command, which works identically to the complain command, but sets the profiles to enforce mode.

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

8 **Rescan all profiles.** To have Novell AppArmor rescan all of the profiles and change the enforcement mode in the kernel, enter `/etc/init.d/subdomain restart`.

3.5.3 Summary of Profiling Tools

All of the Novell AppArmor profiling utilities are provided by the `subdomain-utils` RPM package and most are stored in `/usr/sbin`. The following sections introduce each tool.

autodep

This creates an approximate profile for the program or application you are autodepping. 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 Novell AppArmor. The minimum autodep approximate profile has at least a base include directive, which contains basic profile entries needed by most programs. For certain types of programs, 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 autodep program. The program argument can be either the simple name of the program, which 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.) and autodep generates an approximate profile, to be improved through the dynamic profiling that follows.

The resulting approximate profile is written to the `/etc/subdomain.d` directory using the Novell 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 autodep is to enter the following in a terminal window when logged in as root:

```
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/subdomain.d`.

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 of the programs you want to profile are in a directory and there are no other programs in that directory, the simple command `autodep /path/to/your/programs/*` creates nominal 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. You then need to manually hunt down the location of these programs and run the autodep program for each one. If the programs are in your path, autodep finds them for you. If they are not in your path, the standard Linux command `locate` might be helpful in finding your programs. If `locate` does not work (it is not installed by default on SUSE Linux), use `find . -name '*foo*' -print`.

Complain or Learning Mode

The complain or learning mode tool detects violations of Novell 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 Novell AppArmor tools to transform log events into improved profiles.

Manually activating the 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 a root user.

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

  ```
  complain [program1 program2 ...]
  ```

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

  ```
  complain /sbin/program1
  ```

- If the profiles are not in `/etc/subdomain.d`, type the following to override the default location:

  ```
  complain /path/to/profiles/ program1
  ```

- Specify the profile for *`program1`*, as follows:

  ```
  complain /etc/subdomain.d/sbin.program1
  ```

Each of the above commands activates the complain mode for the profiles/programs listed. The command can list either programs or profiles. If the program name does not include its entire path, then complain searches `$PATH` for the program. So, for instance, `complain /usr/sbin/*` finds profiles associated with all of the programs in `/usr/sbin` and put them into complain mode, and `complain /etc/subdomain.d/*` puts all of the profiles in `/etc/subdomain.d` into complain mode.

Enforce Mode

The enforce mode tool detects violations of Novell 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 turned on. Turn complain mode on when you want the Novell AppArmor profiles to control the access of the program that is profiled. 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 a root user.

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

  ```
  enforce [program1 program2 ...]
  ```

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

  ```
  enforce /sbin/program1
  ```

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

  ```
  enforce /path/to/profiles/program1
  ```

- Specify the profile for *`program1`*, as follows:

  ```
  enforce /etc/subdomain.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/subdomain.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, enforce searches `$PATH` for the program. For instance, `enforce /usr/sbin/*` finds profiles associated with all of the programs in `/usr/sbin` and puts them into enforce mode. `enforce /etc/subdomain.d/*` puts all of the profiles in `/etc/subdomain.d` into enforce mode.

genprof

genprof (or Generate Profile) is Novell AppArmor's profile generating utility. It runs 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 Novell AppArmor, marks the syslog, and prompts the user to execute the program and exercise its functionality. Its syntax is as follows:

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

If you were to create a profile for the the Apache Web server program httpd2-prefork, you would do the following in a root shell:

1. Enter `rcapache2 stop`.
2. Next, enter `genprof httpd2-prefork`.

   Now genprof does the following:

   - Resolves the full path of httpd2-prefork based on your shell's path variables. You can also specify a full path. On SUSE Linux, the full path is `/usr/sbin/httpd2-prefork`.
   - 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 autodep program described in Section 3.5.3, “Summary of Profiling Tools” (page 56).

     NOTE

     There is a naming convention relating the full path of a program to its profile filename so that the various Novell AppArmor profiling

tools can consistently manipulate them. The convention is to replace a forward slash (/) with period (.) so that the profile for `/usr/sbin/httpd2-prefork` is stored in `/etc/subdomain.d/usr.sbin.httpd2-prefork`.

- 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:

```
Oct  9 15:40:31 SubDomain: PERMITTING r access to
/etc/apache2/httpd.conf (httpd2-prefork(6068) profile
/usr/sbin/httpd2-prefork active /usr/sbin/httpd2-prefork)
```

- Marks syslog with a beginning marker of log events to consider. Example:

```
Sep 13 17:48:52 h2o 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 so 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, which can be used after you have executed the program functionality:

- S runs logprof against the system log from where it was marked when genprof was started and reloads the profile.

  If system events exist in the log, Novell AppArmor parses the learning mode log files. This generates a series of questions that you must answer to guide 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 Apache* (page 105).

5 Answer two types of questions:

- A resource is requested by a profiled program that is not in the profile (see Example 3.1, "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 3.2, "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 to the profile or to add the program into the profile. The following two figures show an example of each one. Subsequent steps describe your options in answering these questions.

Example 3.1 *Learning Mode Exception: Controlling Access to Specific Resources*

```
Reading log entries from /var/log/messages.
Updating subdomain profiles in /etc/subdomain.d.

Profile: /usr/sbin/xinetd
Execute: /usr/sbin/vsftpd

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

Dealing with execute accesses is complex. You must decide which of the three kinds of execute permissions to grant the program:

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 the less program as a pager or the Mozilla Web browser using the Acrobat program 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 fails with permission denied. This is most useful if the parent program is invoking a global service, such as DNS lookups or sending mail via your system's MTA.

unconfined (ux)
: The child runs completely unconfined without any Novell AppArmor profile being applied to the executed resource.

Example 3.2 *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
```

The above menu shows Novell AppArmor suggesting directory path entries that have been accessed by the application you are profiling. It might also require you to define execute permissions for entries.

Novell AppArmor provides one or more pathnames or includes. By clicking the option number, select from one or more of the following options, then proceed to the next step.

NOTE

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

`#include`
: This is the section of a Novell 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 clicking *Glob* as described in the next step. For information about globbing syntax, refer to Section 3.6, “Pathnames and Globbing” (page 73).

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

6 After you select the pathname or include, you can process it as an entry into the Novell AppArmor 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 to build the profile:

Press Enter
: Allows access to the selected directory path.

Allow
: Allows access to the specified directory path entries. Novell AppArmor suggests file permission access. For more information, refer to Section 3.7, "File Permission Access Modes" (page 74)

Deny
: Prevents the program from accessing the specified directory path entries. Novell AppArmor then moves on 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 you enter does not actually satisfy the event that prompted the question in the first place, Novell AppArmor asks you for confirmation and lets you reenter the expression.

Glob
: Clicking this modifies the directory path (by using wild cards) to include all files in the suggested entry directory. Double-clicking it grants access to all files and subdirectories beneath the one shown.

  For more information on globbing syntax, refer to Section 3.6, "Pathnames and Globbing" (page 73).

Glob w/Ext
: Clicking 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. Double-clicking it grants access to all files (with the particular extension) and subdirectories beneath the one shown.

Edit
: Lets you edit the selected line. The new edited line appears at the bottom of the list.

Abort
: Aborts logprof, dumping all rule changes entered so far and leaving all profiles unmodified.

Finish
: Closes logprof, saving all rule changes entered so far and modifying all profiles.

7 To view and edit your profile using vim, enter `vim /etc/subdomain.d/`*`profilename`* in a terminal window. To enable syntax coloring when you edit a Novell AppArmor profile in vim, use the commands `:syntax on` then `:set syntax=subdomain`. For more information about about vim and syntax coloring, refer to Section “Subdomain.vim” (page 71).

logprof

logprof is an interactive tool used to review the learning or complain mode output found in the syslog entries then generate new entries in Novell AppArmor security profiles.

When you run 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 syslog. logprof uses this information to observe program behavior.

If a confined program forks and execs another program, logprof sees this and asks the user which execution mode should be used when launching the child process. The following execution modes are options for starting the child process: *ix*, *px*, and *ux*. 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 autodep run on them and are loaded into Novell AppArmor, if it is running.

When logprof exits, profiles are updated with the changes. If the SubDomain 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.

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

`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/subdomain.d/`.

`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/messages`.

`logprof -m "string marker in logfile"`
: Marks the starting point for logprof to look in the system log. logprof ignores all events in the system log before the specified mark is seen. If the mark contains spaces, it must be surrounded with quotes to work correctly. Example: `logprof -m e2ff78636296f16d0b5301209a04430d`

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

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

However, there might 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*` | logprof -f -`.

logprof Example 1

Following is an example of how logprof addresses httpd2-prefork accessing the file `/etc/group`. The example uses `[]` to indicate 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 pulls in a predefined set of Novell 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 nameservice 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:

Press Enter
: Allows access to the selected directory path.

Allow
: Allows access to the specified directory path entries. Novell AppArmor suggests file permission access. For more information about this, refer to Section 3.7, “File Permission Access Modes” (page 74).

Deny
: Prevents the program from accessing the specified directory path entries. Novell AppArmor then moves on 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 you enter does not actually satisfy the event that prompted the question in the first place, Novell AppArmor asks you for confirmation and lets you reenter the expression.

Glob
: Clicking this modifies the directory path (by using wild cards) to include all files in the suggested entry directory. Double-clicking it grants access to all files and subdirectories beneath the one shown.

  For more information about globbing syntax, refer to Section 3.6, “Pathnames and Globbing” (page 73).

Glob w/Ext
: Clicking 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. Double-clicking it grants access to all files (with the particular extension) and subdirectories beneath the one shown.

Edit
: Lets you edit the selected line. The new edited line appears at the bottom of the list.

Abort
: Aborts logprof, dumping all rule changes entered so far and leaving all profiles unmodified.

Finish
: Closes logprof, saving all rule changes entered so far and modifying all profiles.

logprof Example 2

In an example from profiling vsftpd, we 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 SUSE Linux 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, Novell 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 of the 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, logprof replaces the suggested path of `/y2k.jpg` with `/*`. Or 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).

The above 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. You must decide which of the three kinds of execute permissions to grant:

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 the less program as a pager or the Mozilla Web browser using the Acrobat program 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 fails with permission denied. This is most useful if the parent program is invoking a global service, such as DNS lookups or sending mail via your system's MTA.

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

In the following example, the `/usr/bin/mail` mail client is being profiled and 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)nconstrained / (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 ball or an RPM file and it shows you the inventory of these containers.

You do not want to automatically run rpm when reading mail messages (that leads directly to a Microsoft* Outlook–style virus attacks, because rpm has the power to install and modify system programs) and 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/mail/less` in this context, the less program is far less dangerous than it would be without Novell AppArmor protection.

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

- The rule written into the profile is px, which forces the transition to the child's own profile.
- logprof constructs a profile for the child and starts building it, in the same way that it built the parent profile, by ascribing events for the child process to the child's profile and asking the logprof user questions as above.

Finally, you might want to grant the child process very powerful access by specifying *Unconfined*. This writes `ux` into the parent profile so that when the child runs, it runs without any Novell AppArmor profile being applied at all. This means running with no protection and should only be used when absolutely required.

Subdomain.vim

A syntax coloring file for the vim text editor highlights various features of an Novell AppArmor profile with colors. Using vim and the Novell AppArmor syntax mode for vim, you can see the semantic implications of your profiles with color highlighting. Use vim to view and edit your profile by typing vim at a terminal window.

To enable the syntax coloring when you edit a Novell AppArmor profile in vim, use the commands `:syntax on` then `:set syntax=subdomain`. Alternatively, you can place these lines in your `~/.vimrc` file:

```
syntax on
set modeline
set modelines=5
```

When you enable this feature, vim colors the lines of the profile for you:

Blue
: `#include` lines that pull in other Novell AppArmor rules and comments that begin with `#`

White
: Ordinary read access lines

Brown
: Capability statements and complain flags

Yellow
: Lines that grant write access

Green
: Lines that grant execute permission (either ix or px)

Red
: Lines that grant unconfined access (ux)

Red background
: Syntax errors that are not loading properly into the SubDomain modules

NOTE

There is a security risk when using these lines in your `.vimrc` file, because it causes vim to trust the syntax mode presented in files you are editing. It might enable an attacker to send you a file to open with vim that might do something unsafe.

Use the `subdomain.vim` and `vim` man pages and the `:help` syntax from within the vim editor for further vim help about syntax highlighting. The Novell AppArmor syntax is stored in `/usr/share/vim/current/syntax/subdomain.vim`.

Unconfined

The `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 Novell AppArmor profiles. It requires root privilege and that it not be confined by a Novell AppArmor profile.

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
- An executable started before a Novell AppArmor profile is loaded does not appear in the output, despite running without confinement
- 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.

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.

3.6 Pathnames and Globbing

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 characters, except `/`. Example: An arbitrary number of path elements, including entire directories.
`**`	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}`	Expand to one rule to match `ab` and one rule to match `cd`. Example: A rule that matches `/{usr,www}/pages/**` to grant access to Web pages in both `/usr/pages` and `/www/pages`.

3.7 File Permission Access Modes

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

`r`	read mode
`w`	write mode
`px`	discrete profile execute mode
`ux`	unconstrained execute mode
`ix`	inherit execute mode
`l`	link mode

3.7.1 Read Mode

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 such as `strace(1)`, `ltrace(1)`, and `gdb(1)`).

3.7.2 Write Mode

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

3.7.3 Discrete Profile Execute Mode

This mode requires that a discrete security profile is defined for a resource executed at a Novell AppArmor domain transition. If there is no profile defined, the access is denied. Incompatible with *inherit* and *unconstrained* execute entries.

3.7.4 Unconstrained Execute Mode

Allows the program to execute the resource without any Novell AppArmor profile being applied to the executed resource. Requires listing execute mode as well. Incompatible with *inherit* and *discrete profile* execute entries.

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 `subdomain(7)` man page.

3.7.5 Inherit Execute Mode

Prevents the normal Novell AppArmor domain transition on `execve(2)` when the profiled program executes the resource. Instead, the executed resource inherits the current profile. Incompatible with *unconstrained* and *discrete profile* execute entries.

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 infrequently used.

3.7.6 Link Mode

The link mode mediates access to symlinks and hardlinks and the privilege to unlink (or delete) files. When a link is created, the file that is linked to must have the same access permissions as the link created (with the exception that the destination does not have to have link access).

4 Managing Profiled Applications

After creating profiles and immunizing your applications, SUSE Linux becomes more efficient and better protected if you perform Novell AppArmor profile maintenance, which involves tracking common issues and concerns. You can deal with common issues and concerns before they become a problem by setting up event notification by e-mail, running periodic reports, updating profiles from system log entries (which is essentially running the logprof tool through YaST), and dealing with maintenance issues. Instructions for performing each of these tasks are available:

- Section 4.1, "Monitoring Your Secured Applications" (page 77)
- Section 4.5, "Maintaining Your Security Profiles" (page 103).

4.1 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, generating periodic reports, or integration into a third-party reporting mechanism. The following sections provide details for using these features and finding additional resources.

- Section 4.2, "Setting Up Event Notification" (page 78)
- Section 4.3, "Reports" (page 81)

- Section 4.4, “Reacting to Security Events” (page 102)

4.2 Setting Up Event Notification

Security event notification is an Novell AppArmor feature that informs a specified e-mail recipient when systemic Novell AppArmor activity occurs. This feature is currently available via YaST.

When you enter an e-mail address, you are notified via e-mail when Novell AppArmor security events occur. You can enable three types of notifications, which are:

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

```
dhcp-101.up.wirex.com has had 10 security events since Tue Oct 12 11:10:00
 2004
```

Summary Notification
: The 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:

```
SubDomain: 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
: The 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 logprof tool (see Section “logprof” (page 65)) uses to interpret profiles. For example:

```
Oct  9 15:40:31 SubDomain: PERMITTING r access to /etc/apache2/httpd.conf
 (httpd2-prefork(6068) profile /usr/sbin/httpd2-prefork active
/usr/sbin/httpd2-prefork)
```

NOTE

To configure event notification, refer to Section 4.2.2, “Configuring Security Event Notification” (page 79). After configuring security event notification, read the reports and determine whether events require follow up. Follow up may include the procedures outlined in Section 4.4.1, “Receiving a Security Event Rejection” (page 102).

4.2.1 Severity Level Notification

You can set up Novell AppArmor to send you event messages for things that are in the severity database and above the level that you select.These are numbered one through ten, ten being the most severe security incident. The `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.

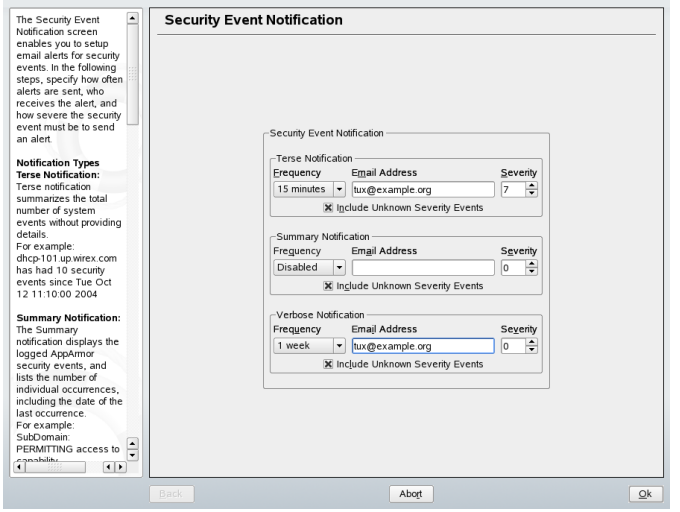
4.2.2 Configuring Security Event Notification

Security event notification is a Novell AppArmor feature that informs you when systemic Novell AppArmor activity occurs. When you select a notification frequency (receiving daily notification, for example), you activate the notification. You are required to enter an e-mail address, so you can be notified via e-mail when Novell AppArmor security events occur.

NOTE

You must set up a mail server on your SUSE Linux 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, you have the option to enable *Terse*, *Summary*, or *Verbose* event notification, which are defined in Section 4.2.1, “Severity Level Notification” (page 79). To be sent a notification e-mail outlining recent Novell AppArmor security events, determine your notification type preference.

3 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. Otherwise you receive an error message. Separate multiple e-mail addresses with commas.

4 For each notification type that you would like 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

5 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. Refer to Section 4.2.1, “Severity Level Notification” (page 79) for more information about severity levels.

6 Click *OK*.

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

8 Click *File* → *Quit* in the YaST Control Center.

4.3 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 cumbersome messages only useful to the logprof tool. You can 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:

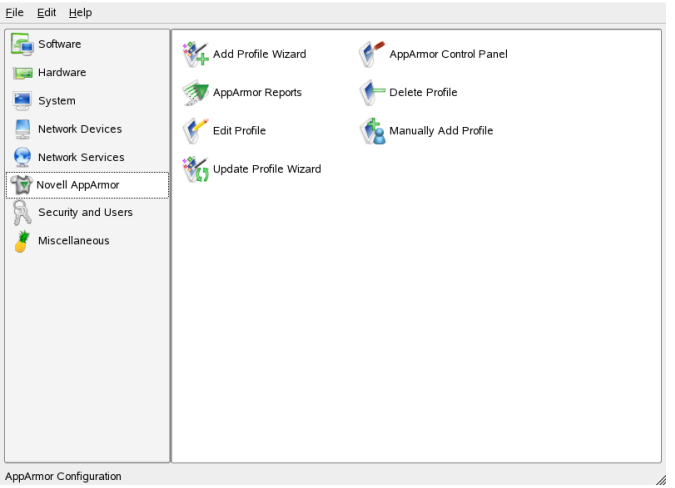
Executive Security Summary
: A combined report, consisting of one or more security incident reports from one or more machines. This report can provide a single view of security events on multiple machines. For more details, refer to Section "Executive Security Summary" (page 91).

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 88).

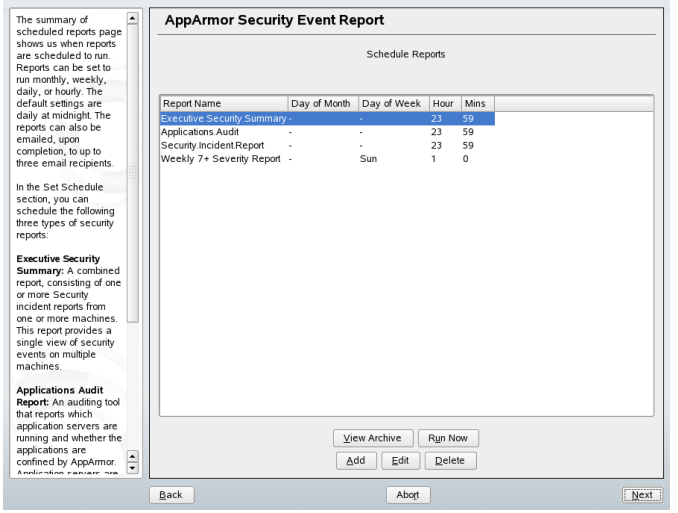
Security Incident Report
: A report that displays application security for a single host. It reports policy violations for locally confined applications during a specific time period. You can edit and customize this report or add new versions. For more details, refer to Section "Security Incident Report" (page 89).

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

1. To run reports, open *YaST* → *Novell AppArmor*. The Novell AppArmor interface opens.



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 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 4.3.1, "Viewing Archived Reports" (page 84).

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 4.3.2, "Run Now: Running On-Demand Reports" (page 93).

Add
: Creates a scheduled security incident report. For *Add* instructions, proceed to Section 4.3.3, "Adding New Reports" (page 95).

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.

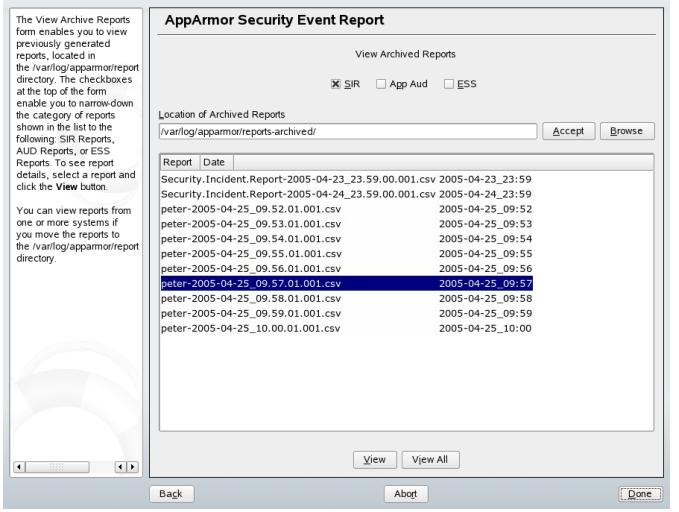
Abort
: Returns you to the Novell AppArmor main screen.

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

4.3.1 Viewing Archived Reports

View Reports enables you to specify the location of a cumulation 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*.



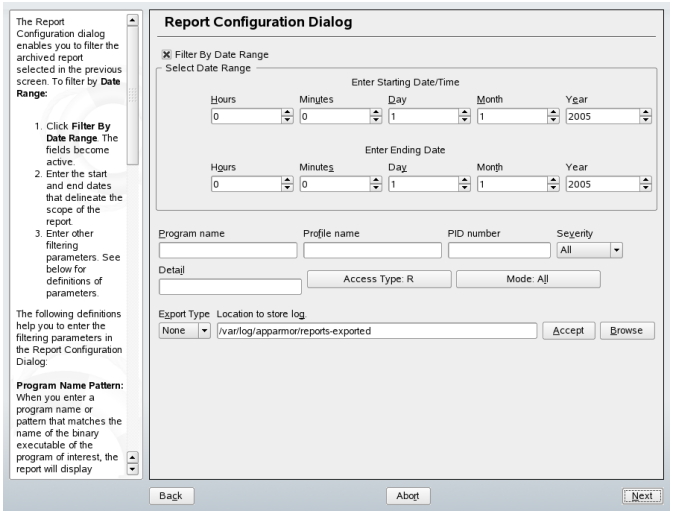
2 Select the report type to view. Toggle between the different types (*SIR* (Security Incident Report), *App Aud* (Application Audit), and *ESS* (Executive Security Summary).

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 87).

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 Level
: 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: `r` (read) `w` (write) `l` (link) `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. You can enter a pathname for your exported report by typing the full pathname in the field provided.

Location to Store Log
: Enables you to change the location that the exported report is store. 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.

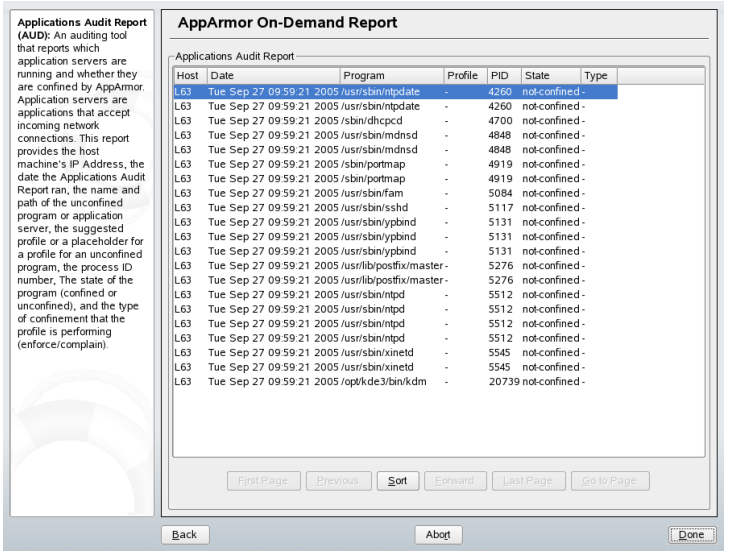
- For the application audit report, refer to Section “Application Audit Report” (page 88).
- For the security incident report, refer to Section “Security Incident Report” (page 89).

- For the executive summary report, refer to Section “Executive Security Summary” (page 91).

Application Audit Report

An auditing tool that reports which application servers are running and whether they are confined by AppArmor. Application servers are applications that accept incoming network connections. This report provides the host machine’s IP address, the date the application audit report ran, the name and path of the unconfined program or application server, the suggested profile or a placeholder for a profile for an unconfined program, the process ID number, the state of the program (confined or unconfined), and the type of confinement that the profile is performing (enforce or complain).

The following screen represents an application audit report:



The following are definitions for the fields in the application audit report:

Host
: The machine protected by AppArmor for which the security events are being 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
: 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).

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.

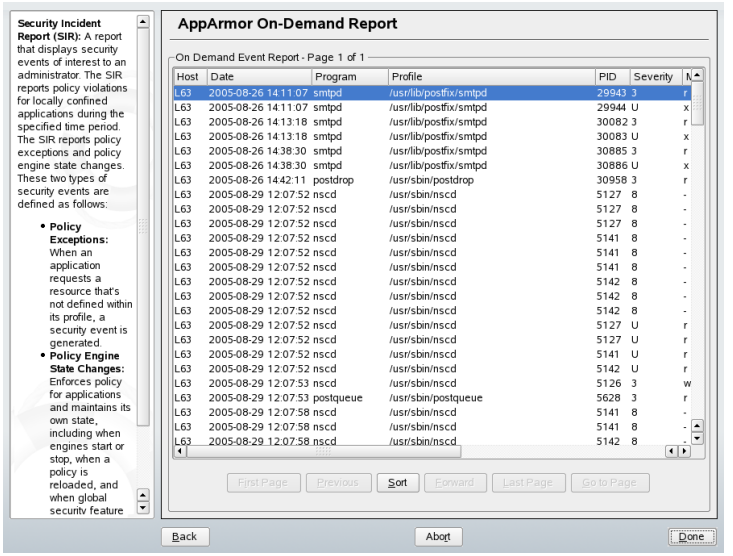
Security Incident Report

A 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 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.

The following screen represents an SIR report:



The following are definitions for the fields in the SIR report:

Host
: The machine protected by AppArmor for which the security events are being 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
: 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).

Severity
: Severity levels of events are reported from the severity database. The severity database defines the importance of potential security events and numbers them one through ten, ten being the most severe security incident. The severity levels are determined 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).

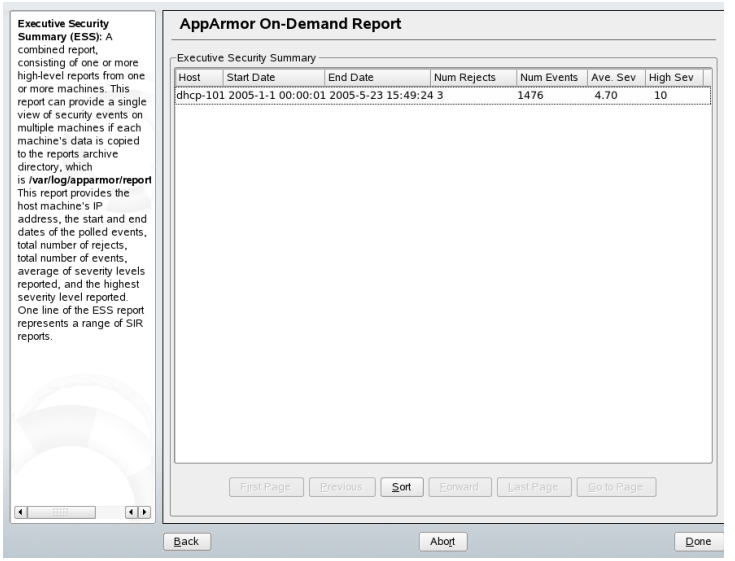
Detail
: A source to which the profile has denied access.This includes capabilities and files. You can use this field to report the resources to which the profile prevents access.

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 reports archive directory, which is `/var/log/apparmor/reports-archived`. This report provides the host machine's IP address, the start and end dates of the polled events, total number of rejects, total number of events, average of severity levels reported, and the highest severity level reported. One line of the ESS report represents a range of SIR reports.

The following screen represents an executive security summary:



The following are definitions for the fields in the executive security summary:

Host
: The machine protected by AppArmor for which the security events are being 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 of Rejects
: In the date range given, the total number of security events that are rejected access attempts.

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

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

High Severity

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

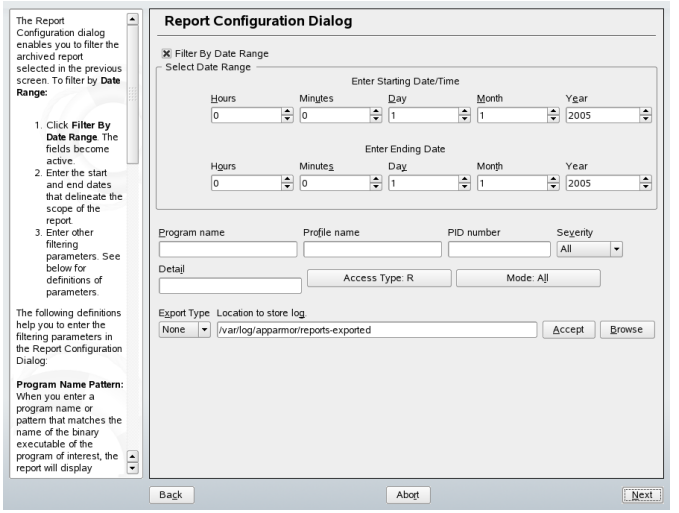
4.3.2 Run Now: Running On-Demand Reports

The *Run Now* report feature enables you to instantly extract report information from the Novell AppArmor event logs without waiting for scheduled events. Return to the beginning of this section if you need help navigating to the main report screen (see Section 4.3, “Reports” (page 81)). 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. For *Application Audit* and *Executive Security Summary* reports, proceed to Step 6 (page 95).

3. The *Report Configuration Dialog* displays 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 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).

Severity Level
: 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. You can enter a pathname for your exported report by typing in the full pathname 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.

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

- For the application audit report, refer to Section “Application Audit Report” (page 88).
- For the recurity incident report, refer to Section “Security Incident Report” (page 89).
- For the executive summary report, refer to Section “Executive Security Summary” (page 91).

4.3.3 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.

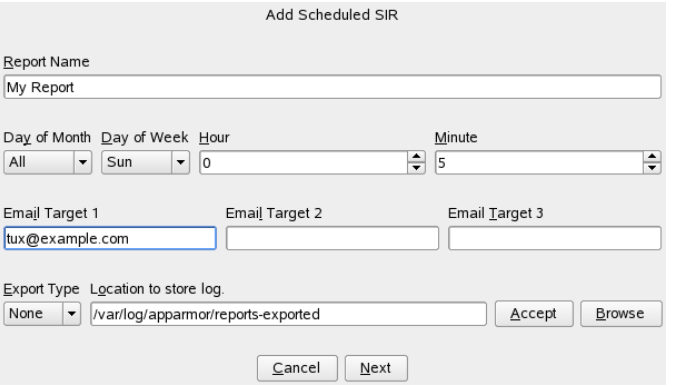
You can configure a daily, weekly, monthly, or hourly report to run for a specified period. You can set the report to display rejections for certain severity levels or to filter by program name, profile name, severity level, or denied resources. This report can be exported to an HTML (Hypertext Markup 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 4.3, “Reports” (page 81)).

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 discern one report from the next.

   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`.

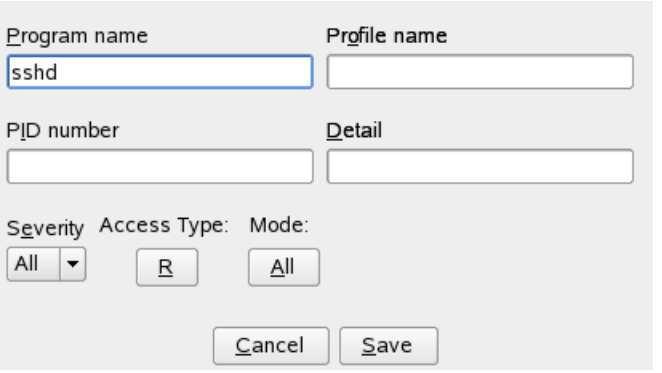
Hour and Minute
: Select the time. This specifies the hour and minute that you would like the reports to run. If you do not change the time, selected reports runs at midnight. If neither month nor day of week are selected, the report runs daily at the secified time.

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. You can enter a pathname for your exported report by typing in the full pathname 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
: 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 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.

4.3.4 Editing Reports

From the AppArmor *Reports* screen, you can select and edit a report. The stock reports cannot be edited or deleted.

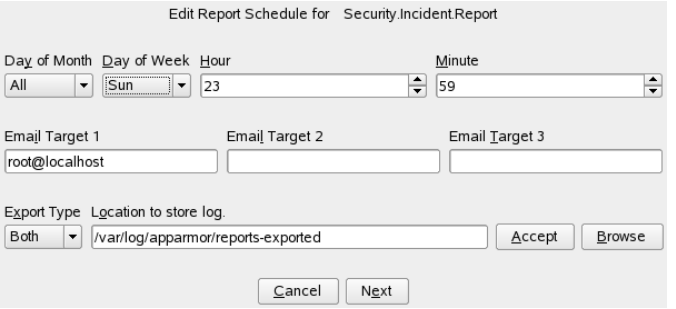
NOTE

Return to the beginning of this section if you need help navigating to the main report screen (see Section 4.3, “Reports” (page 81)).

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

1. From the list of reports in the *Schedule Reports* window, select the report to edit.

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



3. Enter 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`.

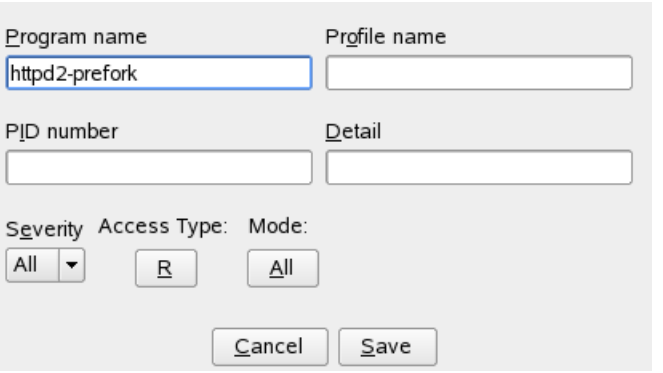
   Hour and Minute
   : Select the time. This specifies the hour and minute that you would like the reports to run. If you do not change the time, selected report runs at midnight. If neither the day of the month nor day of the week is selected, the report runs daily at the specified time.

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. You can enter a pathname for your exported report by typing the full pathname 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.



5 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 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.

4.3.5 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 comfirmation 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*.

4.4 Reacting to Security Events

There are a few common maintenance issues that you should regularly inspect and deal with according to the rules that you have established. The following are some common maintenance issues that you might encounter:

- Section 4.4.1, “Receiving a Security Event Rejection” (page 102).
- Section 4.5.2, “Changing Your Security Profiles” (page 104).

4.4.1 Receiving a Security Event Rejection

When you receive a 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 rejection represents normal application behavior, running logprof at the command line or the *Update Profile Wizard* in Novell AppArmor allows you to iterate through all reject messages. By selecting the one that matches the specific reject, you can automatically update your profile.

If the rejection 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.

4.4.2 Changing Application Security

Users can always manually edit the profile, using vim at the command line or *Edit Profile* in YaST.

4.5 Maintaining Your Security Profiles

In a production enviroment, 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 backup and restore security policy files, plan for software changes, and allow any needed modification of security policies that your enviroment dictates. These items are covered in the following sections:

- Section 4.5.1, “Backing Up Your Security Profiles” (page 103).
- Section 4.5.2, “Changing Your Security Profiles” (page 104).
- Section 4.5.3, “Introducing New Software into Your Environment” (page 104).

4.5.1 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/subdomain.d
   ```

   The simplest method to ensure that your security policy files are regularly backed up is to include the directory `/etc/subdomain.d` in your 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.

4.5.2 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 3.3.3, “Editing a Profile” (page 39).

4.5.3 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 enviroment, the best method for updating your profiles is one of the following:

- Run the profiling wizard by selecting *Add Profile Wizard* in YaST. This updates your application profile set with the current productions using minimal effort. For step-by-step instructions, refer to Section 3.3.1, “Adding a Profile Using the Wizard” (page 27).

- Run genprof by typing `genprof` in a terminal while logged in as root. For detailed instructions, refer to Section “genprof” (page 60).

If you intend to deploy a patch or upgrade directly into a production enviroment, 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 logprof. For detailed instructions, refer to Section “logprof” (page 65).

- Run the profiling tools 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 3.3.5, “Updating Profiles from Syslog Entries” (page 42).

Profiling Your Web Applications Using ChangeHat Apache

5

A Novell® AppArmor profile represents 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.

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

For more information see the `change_hat` man page.

5.1 Apache ChangeHat

Novell AppArmor provides a `mod_change_hat` module for the Apache program. The `mod_change_hat` module works on your SUSE Linux to make the Apache web server become "ChangeHat aware." It is installed if Apache is on your system.

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-dev/templates/classic/images/bar_left.gif`)
- `DEFAULT_URI`
- `HANDLING_UNTRUSTED_INPUT`

If you have the required Apache 2 on your system, the `mod_change_hat` module is automatically installed with Novell AppArmor as well as added to the Apache configuration. Apache 1.3 is not supported.

NOTE

If you install `mod_change_hat` without Novell AppArmor, you need to make sure the Apache load module has a command in the config file that loads the `mod_change_hat` module by adding the following line to your Apache configuration file:

```
LoadModule change_hat_module modules/mod_change_hat.so
```

5.1.1 Tools for 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. Manage ChangeHat-aware applications much more flexibly at the command line, but the process is also more complicated. Both methods allow you to manage the hats for your application and populate them with profile entries.

In the following steps, we walk you through a demo 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. This allows you to 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, you may safely select *Use Default Hat* to process this URI in the default hat, which is the default security profile.

In the demo, we create a new hat for the URI `phpsysinfo-dev` and its subsequent accesses. Using the profiling utilities, we delegate what is added 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-dev` URI is passed to the Apache Web server.

In this demo, we generate a profile for the application phpsysinfo (refer to `http://phpsysinfo.sourceforge.net` for more information). The phpsysinfo-dev package is assumed to be installed under `/srv/www/htdocs/phpsysinfo-dev/` in a clean (new) install of Novell AppArmor.

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

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

3 Click *Create Profile*. The *AppArmor Profiling Wizard* window opens.

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

NOTE

Restart any program you are profiling at this point.

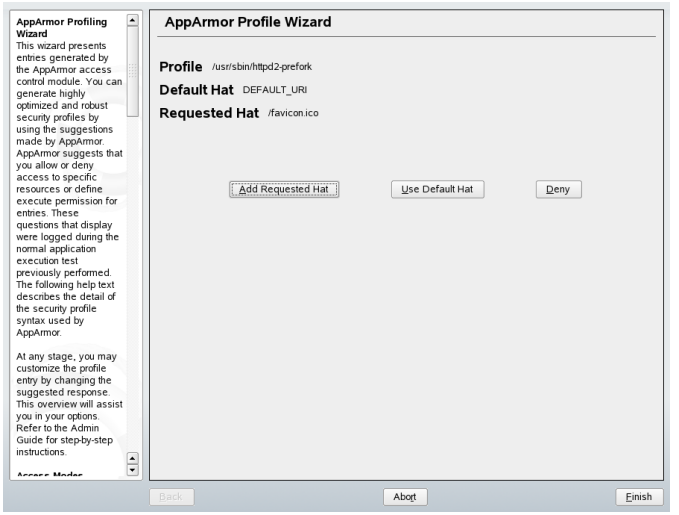
5 Open `http://localhost/phpsysinfo-dev/` in a Web browser window. The browser window should display network usage and system information.

NOTE

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

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

7 In our demo, logprof first prompts us with *Add Requested Hat* or *Use Default Hat* because it noticed that a URI was accessed `phpsysinfo-dev`. Select *Add Requested Hat*.



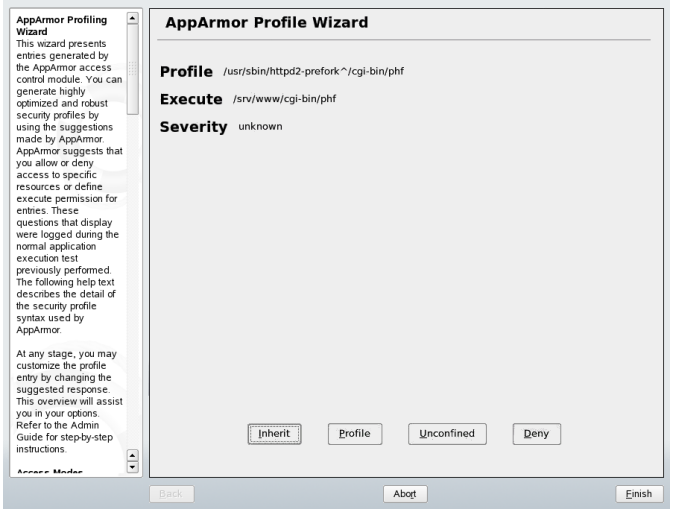
8 Click *Allow*.

Choosing *Add Requested Hat* in the previous step creates a new hat in the profile and specifies that 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 phpsys-info-dev 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

Selecting *Unconfined* can create a significant security hole and should be done with caution.



a Select *Inherit* for the `/bin/bash` path. This adds `/bin/bash/` (accessed by Apache) to the phpsysinfo-dev hat profile with the necessary permissions.

b 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 Section 3.3.1, “Adding a Profile Using the Wizard” (page 27).

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

The following is an example of what a phpsyinfo-dev hat might resemble.

Example 5.1 *Example phpsysinfo-dev Hat*

```
^phpsysinfo {
  #include <program-chunks/base-files>
    /bin/df                                 ix,
    /bin/bash                               ix,
    /dev/tty                                rw,
    /etc/SuSE-release                       r,
    /etc/fstab                              r,
    /etc/hosts                              r,
    /etc/mtab                               r,
    /proc/**                                r,
    /sbin/lspci                             ix,
    /srv/www/htdocs/sysinfo/**              r,
    /sys/bus/pci/devices                    r,
    /sys/devices/**                         r,
    /usr/bin/who                            ix,
    /usr/share/pci.ids                      r,
    /var/log/apache2/{access,error}_log     w,
    /var/run/utmp                           r,
  }
```

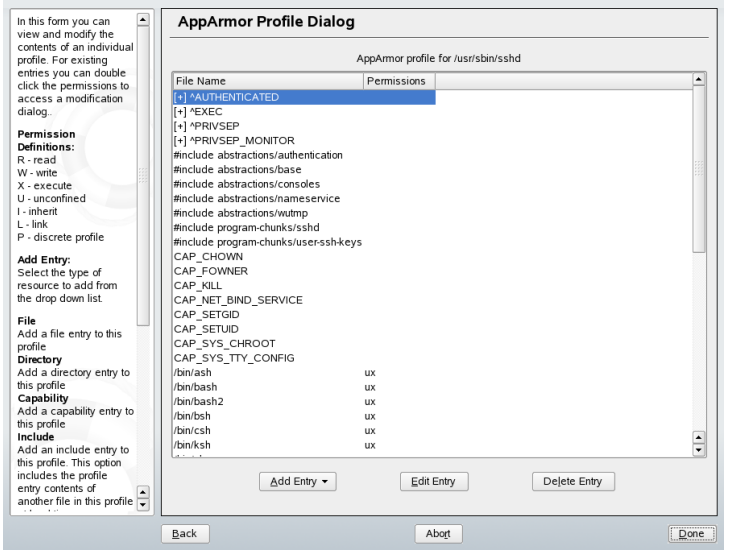
NOTE

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

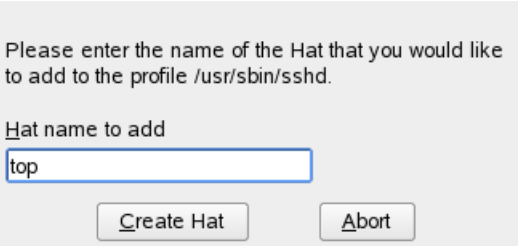
5.1.2 Adding Hats and Entries to Hats

When you use the *Edit Profile* dialog (for instructions, refer to Section 3.3.3, “Editing a Profile” (page 39)) or when you add a new profile using *Manually Add Novell AppArmor Profile* (for instructions, refer to Section 3.3.2, “Manually Adding a Profile” (page 34)), you are given the option of adding hats (subprofiles) to your Novell AppArmor profiles.

You can add a ChangeHat subprofile from the *AppArmor Profile Dialog* window.



1 From the *AppArmor Profile Dialog* window, click *Add Entry* then select *Hat*.The *Enter Hat Name* dialog box opens:



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 an example of an Novell AppArmor profile, refer to Example 5.1, “Example phpsysinfo-dev Hat” (page 111).

5.2 Apache Configuration for mod_change_hat

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 nonexistent file in an existing directory), is accepted or rejected. For Apache documentation on virtual host directives, refer to `http://httpd.apache.org/docs-2.0/mod/core.html#virtualhost`

The change_hat specific configuration keyword is `ImmDefaultHatName` and is used similarly to `ImmHatName`, for example, `ImmDefaultHatName 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, thereby 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_change_hat` attempts to apply hats.

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

2. A hat named by the entire URI path.

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

4. `DEFAULT_URI` (and 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_change_hat` should use a specific hat:

```
<Location /foo/>
      ImmHatName 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 pathname in the file system, in the following example:

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

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

```
<Location "/sysinfo">
          ImmHatName sysinfo
</Location>
```

The following hat should then work for phpsyinfo:

```
^sysinfo {
  #include <program-chunks/base-files>
    /bin/df                                 ix,
    /bin/bash                               ix,
    /dev/tty                                rw,
    /etc/SuSE-release                       r,
    /etc/fstab                              r,
    /etc/hosts                              r,
    /etc/mtab                               r,
    /proc/**                                r,
    /sbin/lspci                             ix,
    /srv/www/htdocs/sysinfo/**              r,
    /sys/bus/pci/devices                    r,
    /sys/devices/**                         r,
    /usr/bin/who                            ix,
    /usr/share/pci.ids                      r,
    /var/log/apache2/{access,error}_log     w,
    /var/run/utmp                           r,
  }
```

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

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

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

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

6 Support

This chapter outlines maintenance-related tasks. Learn how to update Novell® AppArmor and SubDomain and get a list of available man pages providing basic help on 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. Finally, get an overview of the support options provided with your copy of SUSE Linux.

6.1 Updating Novell AppArmor Online

Updates for Novell AppArmor packages will be provided through YOU (YaST Online Update). Retrieve and apply them exactly like for any other package that ships as part of a SUSE Linux product.

6.2 Using the Man Pages

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

Table 6.1 *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)`

- `subdomain.conf(5)`
- `subdomain.d(5)`
- `subdomain.vim(5)`
- `subdomain(7)`
- `subdomain_parser(8)`

6.3 For More Information

More information about the AppArmor product can be found on the Novell AppArmor product page at Novell: `http://www.novell.com/products/apparmor/`.

The product documentation for Novell AppArmor including this document can be found under `http://www.novell.com/documentation/apparmor/` or in the installed system under `/usr/share/doc/packages/subdomain-docs/`.

If you want to discuss SUSE Linux and AppArmor with others, subscribe to the SUSE Linux mailing list (`mailto:suse-linux-e-subscribe@suse.com`).

6.4 Troubleshooting

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

SUSE Linux is installed, but AppArmor does not appear in the YaST menu
: AppArmor is installed by default if either the GNOME or KDE desktop is chosen at installation time. If you choose *Minimal Graphical System* or *Text Mode*, AppArmor is not included by default. In these cases, use YaST to install the missing packages. For more information about this, refer to *Novell AppArmor Powered by Immunix 1.2 Installation and QuickStart Guide*.

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 applications audit report. You can filter dates and times to narrow down the specific periods when application behavior began.

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 mod_change_hat) or make configuration changes to Apache, you should run through profiling Apache again to catch any additional rules they need to be added to the profile.

Reports are not being sent via 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.

  Users must be aware of the mail size limits and should check their archives if e-mails have not been received.

Excluding certain profiles from the list of profiles used
: AppArmor always loads and applies all profiles that are available in its profile directory (`/etc/subdomain.d/`). If you decide not to apply a profile to a certain application, either delete the appropriate profile or move it to another location where AppArmor would not check for it.

AppArmor operation can generate various errors. Here is a list of possible errors and how to resolve them.

Can’t find subdomain_parser
: If you run logprof as a non-root user, such as tux, you are likely to see this error:

```
tux@localhost:~> /usr/sbin/logprof
Can’t find subdomain_parser.
```

NOTE

You should run logprof only as root.

/usr/sbin/genprof must be run as root
: Running genprof as a non-root user produces a similar result:

```
tux@localhost:~> /usr/sbin/genprof
/usr/sbin/genprof must be run as root.
```

Unloading SubDomain profiles..failed
: You must run the subdomain start and subdomain stop scripts as root. Running them as a non-root user produces this result:

```
tux@localhost:~> /etc/init.d/subdomain stop
/sbin/subdomain_parser: Sorry. You need root priveleges to run this
program.
Unloading SubDomain profiles..failed
```

Subdomain parser error
: The example below shows the syntax of the entire parser error.

  Manually editing Novell AppArmor profiles can introduce syntax errors. If you attempt to start or restart SubDomain with syntax errors in your profiles, you see error results like this:

```
localhost:~ # /etc/init.d/subdomain start
Loading SubDomain profiles
Subdomain parser error, line 2: Found unexpected character: 'h'
Profile /etc/subdomain.d/usr.sbin.squid failed to load
failed
```

The version of AppArmor that you are running does not allow the creation of this profile.
: To upgrade to a fully functional version of Novell AppArmor, contact sales@novell.com.

6.5 Support for SUSE Linux

Useful support information for SUSE Linux is available in a number of sources. If you encounter problems with the installation or use of SUSE Linux that you are unable to solve, our experienced support staff can offer practical assistance with the free installation support for registered products and the incident-based support by phone or e-mail. Nearly all common customer problems can be eliminated quickly and competently.

6.5.1 Advanced Support

Qualified support is available by phone and e-mail at transparent rates. SUSE Linux 10.0 comes with 90-day installation support. Additionally, if you are running SUSE Linux for personal use, you can take advantage of our at-home Advanced Support program. You can reach us by phone:

- Germany: 0190-86 28 00 (1.86 €/minute)
- Austria: 0900-47 01 10 (1.80 €/minute)
- Switzerland: 0900-70 07 10 (3.13 SFr/minute)
- Rest of Europe: Phone: +44-1344-326-666, Price: € 46 including VAT. Monday-Friday from 12:00 to 18:00 CET
- United States and Canada: Phone: +1-800-796-3700. Price: $39 including tax. Monday-Friday from 09:00 a.m. to 06:00 p.m. EST or 06:00 a.m. to 03:00 p.m. PST.
- All other countries: Phone: +44-1344-326-666, Price: € 46 including VAT, Monday-Friday, 12:00-18:00 CET

One incident covers up to twenty minutes of assistance from our experienced support staff. The payment is credit-card based. Visa, Eurocard, and Mastercard are accepted. Financial transactions may be handled by our service partner, Stream / ECE EMEA Ltd.

Please be aware that the phone numbers may change during the sales cycle of SUSE Linux 10.0. Current numbers as well as a detailed listing of the subjects covered by the Advanced Support Service can be found at http://www.novell.com/usersupport.

NOTE

While our expert staff will do their best to provide top-quality support, we cannot guarantee a solution.

We endeavor to help you as quickly and precisely as possible. The effort and time needed is considerably reduced if the question is formulated clearly. Please have answers to the following questions ready before contacting us:

1. Which program and version are you using? During which process does the problem occur?

2. What exactly is the problem? Try to describe the error as precisely as possible, using phrases with words such as *when* (for example, "When X is pressed, this error appears").

3. What hardware do you use (graphics card, monitor, printer, ISDN card, etc.)?

Detailed documentation can be found in manuals, online help, and the Support Database. In most cases, even problems that seem more difficult to solve are covered in the comprehensive documentation included with SUSE Linux. The SUSE Help Center on your desktop provides additional information about installed packages, the vital HOWTOs, and info pages.

You can access the latest Support Database articles online at `http://www.novell.com/usersupport`. By means of the Support Database, which is one of the most frequently used databases in the Linux world, we offer our customers a wealth of analysis and solution approaches. You can retrieve tested solutions using the keyword search, history function, or version-dependent search.

6.5.2 Free Installation Support

Our free installation support is provided for a period of 90 days following the activation of your registration code (starting latest with the release of a new version). If you cannot find an answer to your question in any of the available information sources, we will gladly provide assistance for the following issues:

- Installation on a typical private workstation or laptop equipped with a single processor, at least 256 MB RAM, and 3 GB of free hard disk space.

- Resizing of one Windows partition that occupies the entire hard disk.

- Installation of a local ATAPI CD or DVD drive.

- Installation on the first or second hard disk in an IDE-only system (`/dev/hda` or `/dev/hdb`) or supported S-ATA system, excluding RAID.
- Integration of a standard keyboard and standard mouse.
- Configuration of the graphical user interface (without the hardware acceleration feature of the graphics card).
- Installation of the boot manager in the MBR of the first hard disk or on a floppy disk without modifying the BIOS mapping.
- Setup of Internet access with a supported PCI ISDN card or external serial modem (not USB). Alternatively, setup of DSL based on PPPoE with a supported NIC.
- Basic configuration of an ALSA-supported PCI sound card.
- Basic configuration of a locally-attached compatible printer with YaST.
- Basic configuration of an IDE CD writer for use with k3b (CD burning application) without changing the jumper setting.
- Configuration of a supported PCI ethernet card for LAN access with either DHCP (client) or static IP. This does not include the configuration of the LAN or any other computers or network components. It also does not cover the configuration of the computer as a router. Fault analysis is limited to checking for proper loading of the kernel module and the correct local network settings.
- Configuration of an e-mail client (only Evolution and KMail) for collecting mail from a POP3 account. Fault analysis is limited to checking for proper settings in the e-mail client.
- Support for the package selection Standard System.
- Upgrade from the previous version of the product.
- Kernel updates (only official SUSE Linux update RPMs).
- Installation of bug fixes and security updates from ftp.suse.com or a SUSE FTP mirror using YOU or the manual method.

For a detailed listing of the subjects covered by the free installation support, please check `http://www.novell.com/usersupport`.

Contact Information for Free Installation Support

- http://www.novell.com/usersupport
- usersupport@novell.com
- Germany: Phone: 0180-500 36 12 (12 Cent/min) (Monday through Friday from 13:00 to 17:00 CET)
- Austria: Phone: +43 1 36 77 4440 (Monday through Friday from 13:00 to 17:00 CET)
- Switzerland: Phone: +41 43 299 7800 (Monday through Friday from 13:00 to 17:00 CET)
- UK: Phone: +44-1344-326-666 (Monday through Friday from 13:00 to 17:00 GMT)
- United States and Canada: Phone: +1-800-796-3700 (Monday through Friday from 12:00 p.m. to 6:00 p.m. EST or 09:00 a.m. to 03:00 p.m. PST)
- France: Phone: +33 1 55 62 50 50 (Monday through Friday from 13:00 to 17:00 CET)
- Spain: Phone: +34 (0)91 375 3057 (Monday through Friday from 13:00 to 17:00 CET)
- Italy: Phone: +39 02 2629 5555, support is available in Italian (Monday through Friday from 13:00 to 17:00 CET)
- Czech Republic: E-mail: support@portal.suse.cz (Monday through Friday)
- All other countries: Support is provided in English only. Phone: +44-1344-326-666 (Monday through Friday from 12:00 to 18:00 CET)

For the most recent contact information, refer to http://www.novell.com/products/linuxprofessional/support/contact.html.

Important Notes

1. Only customers with a valid, activated registration code are entitled to free support. You can activate your registration code at http://www.novell.com/usersupport.

2. The registration code is not transferable to another person.

3. The free support covers only the initial installation on one computer. Refer to our Web site for further information.

4. We can provide support only for hardware supported by SUSE Linux. Refer to our Component Database at www.novell.com/usersupport/hardware for information about supported hardware components.

5. There are no guaranteed turnaround times for mail inquiries.

Contact Recommendations

Misspelled commands, links, or directory names often cause frustrating problems and are particularly common during phone conversations. To help prevent this problem, please send us a brief description of your question or problem by e-mail. You will receive a reply soon after that provides a practical solution.

6.6 Reporting Bugs for AppArmor

The developers of AppArmor and SUSE Linux are eager to deliver products of the highest quality. Your feedback and your bug reports help us to keep up the good work. So, 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:

a. Click *Create New Account* on the *Login to Continue* page.

b. 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*.

   Either 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 you want to file the bug against. In your case, this would be your SUSE Linux 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 below.

   You may create attachments to your bug report holding screen shots, log files, or test cases.

9. Click *Submit* after you have entered all the details to send your report to the developers.

Glossary

Apache
: Apache is a freely available UNIX-based Web server. It is currently the most commonly used Web server on the Internet. More information about Apache can be found 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 vulnerabilty 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.

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, SUSE Linux, 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 Identifiers. 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.immuix.com/index.html`, 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.