PAM-modules

version 2.2, 1 January 2018

Sergey Poznyakoff.

Short Contents

Table of Contents

1 Introduction to PAM-modules

PAM-modules is a collection of various pluggable authentication modules. This manual describes each module in detail. The reader is expected to be sufficiently proficient with general UNIX administration issues and with Pluggable Authentication Modules (PAM) in particular.

Each module is configurable from its command line. Modules that require such amounts of configuration data, that are inconvenient to pass from the command line (see Chapter 6 [sql], page 15), implement their separate configuration files.

Several command line options are common for all modules. These are:

`debug[=`*`level`*`]`
: Change debugging level (0 <= *level* <= 100). The debugging information will be logged via `syslog` channel `auth.debug`. Notice, that debugging output can reveal authentication credentials. In particular, user password is displayed on debugging level 100.

`audit`
: Log full debugging information (equivalent to `debug=100`).

`waitdebug[=`*`interval`*`]`
: Wait for *interval* seconds before starting. This option is intended for the package developers and is not enabled, unless you configure the package with `--enable-debug` option. Most probably you will not need this option. The following description is provided in case you decide to participate in `PAM-modules` development:

  When this option is present, the module displays the following diagnostics in `syslog auth.crit` channel:

  ```
  WAITING FOR DEBUG
  ```

  and waits for *interval* seconds (default 3600) before actually starting to do anything. The developer is supposed to attach to the process with a debugger, set the `interval` variable to 0 and to continue execution of the module in the debugging mode.

Some modules perform PAM *item expansion* on their arguments. It is a feature similar to shell's variable expansion. During item expansion, any occurrence of `$`*`name`* in a string is replaced by the value of the PAM item *name*. If the item in question is not defined, an empty string is substituted instead. A limited support for the shell-style default values is available: namely, the notation `${`*`item`*`:-`*`value`*`}` expands to the value of *item* if it is set, and to *value* otherwise. Notice, that *value* must be a literal value (string or numeric).

The following table lists PAM item names:

'`service`'
: `PAM_SERVICE`. The service name (which identifies the PAM stack that will be used).

'user' PAM_USER. The username of the entity under whose identity service will be given.

'tty' PAM_TTY. The terminal name: prefixed by '/dev/' if it is a device file; for graphical, X-based, applications the value for this item is usually the $DISPLAY environment variable.

'rhost' PAM_RHOST. The requesting hostname (the hostname of the machine from which the PAM_RUSER entity is requesting service). That is 'PAM_RUSER@PAM_RHOST' identifies the requesting user. In some applications, PAM_RHOST may be 'NULL'.

'ruser' PAM_RUSER. The requesting entity: user's name for a locally requesting user or a remote requesting user. In some cases, PAM_RUSER may be 'NULL'.

'prompt' PAM_USER_PROMPT. The string used when prompting for a user's name. The default value for this string is 'Please enter username: '.

'password'
PAM_AUTHTOK. The authentication token (often a password).

2 Verify PAM Access

The `pamck` utility checks if a user can be authenticated using PAM. The user name is specified in the command line, so the simplest invocation is:

```
$ pamck user
```

When used this way, `pamck` first authenticates '`user`', by calling `pam_authenticate`, and then performs account management (`pam_acct_mgmt`). If both functions return success, the utility prints '`OK`' on the standard output and exits with zero code. In case of failure, it displays diagnostics on standard error and exits with error code 2.

It exits with code 1 in case of usage error (e.g. wrong command line option).

If password is required, the utility asks about it, and waits for the user input. When reading user input, terminal echo is turned off to prevent password compromising.

Alternatively, the password may be given on the command line, as the second argument:

```
$ pamck user pass
```

By default, `pamck` uses PAM service '`check`'. Another service name may be supplied using the `-s` command line option:

```
$ pamck -s login user
```

The `-g` command line option allows to select the PAM management group to check. It takes the name of the group as an argument. Allowed group names are:

auth	Authentication group. Call `pam_authenticate`.
acct	Account management. Call `pam_acct_mgmt`.
open	Session management. Call `pam_open_session`.
close	Session management. Call `pam_close_session`.
pass	Password management. Call `pam_chauthtok`.

The following table summarizes available command line options:

`-s` *`service`*
: Select service name to use.

`-g` *`group`*
: Select PAM management group to check.

`-h`
: Print short help summary and exit.

`-v`
: Print program version and copyright information and exit.

3 Authentication against an alternative shadow file.

The `pam_fshadow` module provides authentication against an alternative `shadow` file, or `passwd` / `shadow` pair (or pairs). There are two main operation modes: *plain* mode, in which `pam_fshadow` uses only one `passwd`/`shadow` pair, and *virtual domain* mode, which allows to select the pair to use based on the authentication token (the user name). First, let's describe the plain mode.

3.1 Using `pam_fshadow` in plain mode.

Plain mode is the default operation mode for `pam_fshadow`. In this mode, the module checks the supplied user name and authentication token against the `passwd`/`shadow` pair located in the system configuration directory (which is set when configuring the package and defaults to *`prefix`*`/etc`). This default location can be changed using the `sysconfdir` option (see below). The authentication is performed as follows:

First, the user name is looked up in `passwd` file and the corresponding record is fetched. If this record contains a valid password hash (i.e. its second field is at least 2 characters long), the system `crypt` function is called on the supplied authentication token with the retrieved hash as its second argument (the `seed`) and its result is compared with the hash. If the two strings compare equal, the user is authenticated successfully.

Otherwise, if `passwd` contains no password, the shadow file is examined and hash retrieved from there is used. If the record retrieved from the shadow file has not expired, and if its password hash field matches the authentication token (using the algorithm described above), the user is authenticated successfully.

Several options are provided to alter the default behavior. All of them, except `sysconfdir`, have the same effect in the virtual domain mode as well. The table below summarizes these options.

`nopasswd` Do not require `passwd` file to be present. Only `shadow` is used for authentication.

`noshadow` Do not require `shadow` file to be present. Only `passwd` is used for authentication. Notice, that it is an error to specify both `nopasswd` and `noshadow`.

`sysconfdir=`*`dir`*
Set full name of the directory where `shadow` and `passwd` are located. By default the system configuration directory will be used.

`use_authtok`
: Do not prompt the user for password, take it from the saved authentication tokens. This option is useful when `pam_fshadow` is used as a non-first module in a stack of authentication modules.

The following example illustrates the use of `pam_fshadow` in plain mode in `pam.conf` file:

```
tuhs auth  required   pam_fshadow.so \
                      sysconfdir=/home/tuhs/tuhs/etc nopasswd use_authtok
```

3.2 Using `pam_fshadow` in virtual domain mode.

In *virtual domain* mode, `pam_fshadow` uses the user name to determine where to look for the `passwd`/`shadow` file pair. The name is split into *user name proper* and *authentication domain*. The configuration directory name is then constructed by concatenating the system configuration directory, a directory separator character ('`/`'), and the name of the authentication domain. Then, authentication proceeds as described above for the plain mode. If the supplied user name does not match the regular expression, `pam_fshadow` proceeds as in plain mode.

This mode is enabled by the option `regex`, which supplies a regular expression to split user names. This regular expression must contain two parenthesized *groups*. First of them is used to extract the user name, and the second one is used to extract the authentication domain. For example, the following option:

```
regex=(.*)@(.*)
```

instructs `pam_fshadow` to use any characters before the '`@`' as the user name, and anything following it as the authentication domain.

Several options are provided, that control the type of regular expression and the way of retrieving authentication data from the user name. These options are:

`basic` Use basic regular expression.

`extended` Use extended regular expression. This is the default.

`ignore-case`
`icase` Use case-insensitive regular expression.

`case` Use case-sensitive regular expressions (default).

`revert-index`
: Use group #2 as the user name and group #1 as the authentication domain.

As an example, consider the following `pam.conf` entry:

```
check auth required pam_fshadow.so \
   sysconfdir=/etc/auth regex=(.*)@(.*) extended
```

It instructs `pam_fshadow` to use '`@`' as the username/domain separator and to look up password databases under the `/etc/auth` directory. For

example, if the supplied user name was '`smith@ftp`', then the module will look for the user name '`smith`' in files `/etc/auth/ftp/passwd` and `/etc/auth/ftp/shadow`.

3.3 Summary of pam_fshadow options

This section summarizes all `pam_fshadow` command line options:

`basic` Use basic regular expressions. See Section 3.2 [virtual domain mode], page 6.

`extended` Use extended regular expression (default). See Section 3.2 [virtual domain mode], page 6.

`ignore-case`
`icase` Use case-insensitive regular expressions. See Section 3.2 [virtual domain mode], page 6.

`nopasswd` Use only `shadow` for authentication. See [pam_fshadow common options], page 5.

`noshadow` Use only `passwd` for authentication. See [pam_fshadow common options], page 5.

`regex=`*`expr`*
Define a regular expression for splitting user name into the proper name and authentication domain.

`revert-index`
In the regular expression introduced by `regex`, group #1 selects authentication domain, and group #2 selects user name. See Section 3.2 [virtual domain mode], page 6.

`sysconfdir=`*`dir`*
Assume *dir* as the system configuration directory. See [pam_fshadow common options], page 5.

`use_authtok`
Do not prompt the user for password, take it from the saved authentication tokens.

See [pam_fshadow common options], page 5.

4 Authentication using regular expressions.

The module `pam_regex` is a general-purpose tool for authentication using regular expressions. You can use it, for example, to allow or deny access depending on whether the user name matches a given regular expression. Another possible use is to modify user names following a predefined pattern (as in `sed`), to supply modules that follow it in the PAM stack with a normalized user name.

As a quick start example, the following `pam.conf` entry forbids access for any user names that look like email addresses:

```
httpd auth  required  pam_regex.so sense=deny regex=.*@.*
```

Here, the argument `regex` supplies a regular expression to match against, and `sense=deny` states that any name matching this expression must be denied.

4.1 Using `pam_regex` to control access.

To control access depending on supplied user name, two options are provided. The option `regex` introduces a regular expression with which to compare a user name:

`regex=`*`expression`*
: Compare user name with *expression*. By default, extended regular expressions with case-sensitive matching are used, but this can be changed using other options (see below).

When this option is used, `pam_regex` allows only login attempts with user names that match *expression*. The `sense` command line option is provided to control that behavior:

`sense={allow|deny}`
: What to do if the user name matches the *expression*. The value '`allow`' means to return `PAM_SUCCESS`, '`deny`' means to return `PAM_AUTH_ERR`. Default is '`allow`'.

4.2 Using `pam_regex` to alter user names.

Another common use for `pam_regex` is to alter user names. This mode is enabled when the `transfer` option is used in the command line:

`transform=`*`expression`*
: Transform the user name using given regular expression.

Its argument, *expression*, is a `sed`-like replace expression of the form:

```
s/regexp/replace/[flags]
```

where *regexp* is a *regular expression*, *replace* is a replacement for each file name part that matches *regexp*. Both *regexp* and *replace* are described in detail in Section "The 's' Command" in *GNU sed*.

As in `sed`, you can give several replace expressions, separated by a semicolon.

Supported *flags* are:

'`g`' Apply the replacement to *all* matches to the *regexp*, not just the first.

'`i`' Use case-insensitive matching

'`x`' *regexp* is an *extended regular expression* (see Section "Extended regular expressions" in *GNU sed*).

'*`number`*' Only replace the *number*th match of the *regexp*.

Note: the *posix* standard does not specify what should happen when you mix the '`g`' and *number* modifiers. `Pam_regex` follows the GNU `sed` implementation in this regard, so the interaction is defined to be: ignore matches before the *number*th, and then match and replace all matches from the *number*th on.

Any delimiter can be used in lieue of '`/`', the only requirement being that it be used consistently throughout the expression. For example, the following two expressions are equivalent:

```
s/one/two/
s,one,two,
```

Changing delimiters is often useful when the *regex* contains slashes. For instance, it is more convenient to write `s,/,-,` than `s/\//-/`.

The following example converts the user name to lower case and removes any suffix starting from the '`@`' symbol:

```
pam_regex.so extended transform=s/.*/\L&/g;s/@.*//
```

Both `transform` and `regex` can be used simultaneously. For example, the following command line first converts the user name to lower case and removes anything after the '`@`' symbol, and then compares it to the given regular expression. Access is denied if the resulting user name matches the expression.

```
pam_regex.so extended transform=s/.*/\L&/g;s/@.*// \
             regex=^(anoncvs|anonymous)$ sense=deny
```

4.3 Summary of `pam_regex` options:

`basic` Use basic regular expressions.

`case` Use case-sensitive regular expressions (default).

`extended` Use extended regular expressions (default).

`ignore-case`
`icase` Use case-insensitive regular expressions.

`regex=`*`expression`*
Compare user name with *expression*.

`sense={allow|deny}`
: What to do if user name matches the *expression*. The value '`allow`' means to return `PAM_SUCCESS`, '`deny`' means to return `PAM_AUTH_ERR`. Default is '`allow`'.

`user=`*`string`*
: Upon successful matching, set PAM user name to *string*.

5 Log arbitrary messages to syslog.

The `pam_log` module is a diagnostic tool. It works similarly to the shell `echo` command, outputting its arguments to the `syslog`. The module can be used in any PAM service stack.

In order to be discerned from arguments, all `pam_log`'s options begin with a dash ('`-`'). They must precede any non-option arguments. If the first non-option argument happens to begin with a dash, you can inhibit its special handling by placing '`--`' before it.

After collecting all options, the module scans the rest of its command line arguments, performs item expansion (see [item expansion], page 1) and outputs the resulting string to the syslog.

The following table lists all the supported options:

`-audit` Similar to `audit` in other modules (see Chapter 1 [Intro], page 1).

`-debug[=`*`level`*`]`
: Similar to `debug` in other modules (see Chapter 1 [Intro], page 1).

`-noopen` Reserved for future use.

`-waitdebug[=`*`interval`*`]`
: Similar to `waitdebug` in other modules (see Chapter 1 [Intro], page 1).

`-pri=`*`facility.priority`*
: Send log messages to the given syslog facility and priority. The *facility* part can be any of: '`user`', '`daemon`', '`auth`', '`authpriv`', '`local0`', '`local1`', '`local2`', '`local3`', '`local4`', '`local5`', '`local6`', '`local7`'.

  The *priority* is any of the following: '`emerg`', '`alert`', '`crit`', '`err`', '`warning`', '`notice`', '`info`', '`debug`'.

  Either *facility* or *priority* (but not both) can be omitted, in which case the following defaults are used: *facility*=`authpriv`, *priority*=`info`.

`-tag=`*`label`*
: Use *label* as the syslog tag, instead of the module name.

The following example illustrates the use of this module:

```
cvs auth       required  pam_regex.so extended \
    regex=^(anoncvs|anonymous)$ sense=allow
cvs account    requisite pam_log.so -tag CVS-ACCESS \
    -pri=daemon.info User ${user:-unknown} is granted CVS access
cvs account    required   pam_permit.so
cvs session    required   pam_permit.so
```

6 SQL Authentication and Session Management.

The package provides two modules for SQL authentication and session management: `pam_mysql`, for MySQL and `pam_pgsql` for PostgreSQL. Both modules share the same set of options and provide similar functionality.

Connecting to an SQL database requires a set of credentials that cannot be conveniently passed via the command line. Therefore, both SQL modules use a special *configuration file* to obtain the necessary data. By default, this file is located in the system configuration directory (usually, `/usr/local/etc`), and is named `pam_sql.conf`. However, another location can be specified in the command line, using `config` command line option.

The command line options understood by both modules are:

`config=`*`file`*
: Read SQL access credentials from the given *file*.

`use_authtok`
: Do not prompt the user for password, take it from the saved authentication tokens. This option is useful when this module is not the first in the stack of authentication modules.

6.1 Configuration File.

Configuration file has a simple line-oriented syntax. Empty lines and lines beginning with '`#`' are ignored. Nonempty lines consist of a keyword and its value, separated by any amount of white space.

Long statements can be split over several lines by placing '\' character at the end of each line, e.g.:

```
query select password \
      from users \
      where user_name='$user'
```

Basic configuration statements provide SQL credentials needed for accessing the database:

`host` *`hostname`*
: Sets hostname or IP address of the machine where the database is running. If the database is only listening on the local socket (`--skip-networking` for MySQL, or lack of `-i` for PostgreSQL), then `host` should be the name of the local socket.

`port` *`number`*
: Sets the SQL port number. This statement is optional. Use it only if your database is running on a port different from the standard.

`db` *`database`*
: Sets database name.

`login` *string*
: Sets SQL user name.

`pass` *password*
: Sets SQL user password.

6.2 Using SQL modules in authentication stack.

When used in the `auth` stack, both SQL modules work as follows. First, the module connects to the database using credentials supplied in the configuration file (see the previous section). Then, it retrieves the value of `passwd-query` from the configuration file and performs PAM item expansion over it (see [item expansion], page 1). The resulting query is sent to the SQL server. If this query produces a non-empty result, the first column from the first tuple is used as encrypted user password and compared with the supplied authentication token. If it matches, the user is authenticated successfully. The comparison consists of the following checks, performed in that order until one of them returns match or the list is exhausted:

1. System `crypt` function.
2. MySQL password encoding algorithm (for MySQL only)
3. Compare MD5 sum of the token with the encrypted password.
4. Compare passwords using LDAP algorithm.
5. Compare both strings literally (only if `allow-plaintext-pass` is set in the configuration file.

The following configuration keywords can be used to disable or enable particular stages of the comparison. The value *bool* should be '`yes`', '`true`' or '`t`' to indicate `true`. Any other value is taken to mean `false`.

`allow-plaintext-pass` *bool*
: The returned password may be plaintext. Without this option, it is supposed to be encrypted using the system `crypt` function.

`allow-ldap-pass` *bool*
: The returned password may be a LDAP-style password hash, i.e. the hash value encoded as base-64 and prefixed with a hashing algorithm name in curly braces. This variable is `true` by default.

`allow-md5-pass` *bool*
: The returned password may be encrypted using MySQL `md5` function. This keyword is specific for `pam_mysql`.

`allow-mysql-pass` *bool*
: The returned password may be encrypted using MySQL `password` function. This keyword is specific for `pam_mysql`.

6.3 Setting PAM environment from an SQL database.

This is an experimental feature, available when compiled with `Linux PAM` libraries. It allows to pass some additional information from the database to the application program using PAM environment.

Special configuration keyword `setenv-query` defines an SQL query for setting the environment. After expanding PAM items (see [item expansion], page 1), this query is executed and the first tuple (row) is taken from its result. Each column in this tuple creates an environment variable: the column name becomes the name of environment variable, the column value becomes the variable value.

Consider for example, the following SQL table:

```
CREATE TABLE userprop (
  username varchar(32),
  dir varchar(128),
  uid int,
  gid int
);
```

which contains, among others, the following data:

```
("smith", "/var/spool/dir/1", 16, 10000)
```

Let the configuration file contain this query definition:

```
setenv-query SELECT dir as home, uid, gid \
             FROM userprop \
             WHERE username='$user'
```

Now assume that the user '`smith`' is authenticated using `pam_mysql`. The `setenv-query` is executed. Then, after `pam_authenticate` the PAM environment will contain:

```
home=/var/spool/dir/1
uid=16
gid=10000
```

6.4 Using SQL modules for session management.

Both `pam_mysql` and `pam_pgsql` can be used for session management. This makes it possible to use your SQL database instead of system `wtmp`/`utmp` files, or as a complement to them.

To enable SQL session management, the configuration file must define the following two variables:

`session-start-query` *query*
: Defines the query to be executed when the session begins.

`session-stop-query` *query*
: Defines the query to be executed when the session ends.

Before executing, both queries are subject to item expansion (see [item expansion], page 1).

As an example, consider the following configuration file statements:

```
session-start-query INSERT INTO acct \
                    (status, username, tty, starttime) \
                    VALUES(0, '$user', now(), '$tty')
session-stop-query  UPDATE acct \
                    SET status=1,
                        sessiontime=age(now(), starttime) \
                    WHERE username='$user'
```

They assume that the PostgreSQL table '`acct`' has the following structure:

`status int`
: Status of the record: '`0`' if the session is active, '`1`' if it is closed.

`username varchar(32)`
: User name.

`tty varchar(16)`
: TTY from where the user logged in.

`starttime timestamp`
: Time when the session was started.

`sessiontime interval`
: Duration of the session if `status=1`.

6.5 Summary of configuration statements.

This section summarizes all available configuration file statements. For each statement it provides a short description and a reference to the section in this manual where it is described.

`allow-ldap-pass` *`bool`*
: The returned password may be a LDAP-style password hash, i.e. the hash value encoded as base-64 and prefixed with a hashing algorithm name in curly braces. This variable is `true` by default. See Section 6.2 [sql auth], page 16.

`allow-md5l-pass` *`bool`*
: The returned password may be encrypted using MySQL `md5` function. This keyword is specific for `pam_mysql`. See Section 6.2 [sql auth], page 16.

`allow-mysql-pass` *`bool`*
: The returned password may be encrypted using MySQL `password` function. This keyword can be used only in `pam_mysql` configuration. See Section 6.2 [sql auth], page 16.

`allow-plaintext-pass` *`bool`*
: The returned password may be plaintext. Without this option, it is supposed to be encrypted using the system `crypt` function. See Section 6.2 [sql auth], page 16.

`db` *database*
: Sets the database name. See Section 6.1 [config], page 15.

`port` *number*
: Defines the SQL port number. See Section 6.1 [config], page 15.

`login` *string*
: Sets the SQL user name. See Section 6.1 [config], page 15.

`pass` *password*
: Sets the SQL user password. See Section 6.1 [config], page 15.

`passwd-query` *query*
: Defines the query used to obtain the user's password from the database. The *query* is subject to item expansion (see [item expansion], page 1).

  See Section 6.2 [sql auth], page 16, for a detailed description.

`session-start-query` *query*
: Defines the query to be executed on session start. The *query* is subject to item expansion (see [item expansion], page 1). See Section 6.4 [sql session], page 17, for a detailed description.

`session-stop-query` *query*
: Defines the query to be executed on session stop. The *query* is subject to item expansion (see [item expansion], page 1). See Section 6.4 [sql session], page 17, for a detailed description.

`setenv-query` *query*
: This query is available when the package is compiled with `Linux PAM` implementation. It allows to select arbitrary data from the database and to store them in PAM environment. The first tuple returned from *query* is selected, the column names are used as environment variable names, and column values as their values.

  The *query* is subject to item expansion (see [item expansion], page 1).

  See Section 6.3 [sql setenv], page 17, for a detailed description.

7 pam_ldaphome

The `pam_ldaphome` facilitates maintenance of a centralized LDAP user database. It can be installed as a part of authentication or session management stack. When invoked, it creates the user home directory, if it does not already exist, and ensures his `.ssh/authorized_keys` is in sync with the database.

Apart from common options, this module understands only one implementation-specific option:

`config=`*file*
: Read configuration from *file*. Default is `pam_ldaphome.conf` in *sysconfdir*.

Actual module configuration is read from the configuration file.

7.1 Configuration file for `pam_ldaphome`

`Pam_ldaphome` reads its configuration from two files: the configuration file supplied with the `config` command line option and the system-wide LDAP configuration file `/etc/ldap.conf`.

The syntax of the former is described in Section 6.1 [config], page 15. Allowed keywords are discussed below.

The syntax of the `/etc/ldap.conf` configuration file is described in Section "LDAP configuration file" in *ldap.conf(5) manpage*. Its parsing can be suppressed using the `ldap-config` statement (see below).

From `/etc/ldap.conf`, the following statements are used: '`base`', '`binddn`', '`bindpw`', '`tls_cacert`', '`uri`'. The '`ssl`' statement is understood if its value is '`start_tls`' or '`off`'. Other values are silently ignored.

In general, all statements defined below can appear in both files. However, since `/etc/ldap.conf` is read by other system utilities as well, we do not recommend using `pam_ldaphome`-specific keywords in it.

The values read from `pam_ldaphome` configuration file override those obtained from the standard LDAP configuration file.

LDAP configuration

`base` *searchbase* [pam_ldaphome config]
: Use *searchbase* as the starting point for the search instead of the default, e.g.:

```
base dc=gnu,dc=org,dc=ua
```

`binddn` *dn* [pam_ldaphome config]
: Use the Distinguished Name *dn* to bind to the LDAP directory. Example:

```
binddn cn=Manager,dc=gnu,dc=org,dc=ua
```

`bindpw` *password* [pam_ldaphome config]
: If `binddn` statement is used, this statement supplies the password for simple authentication.

`bindpwfile` *file* [pam_ldaphome config]
Read password for simple authentication from *file*.

`filter` *expr* [pam_ldaphome config]
Sets the LDAP filter expression to return a user profile. The *expr* should conform to the string representation for search filters as defined in RFC 4515.

`ldap-config` *file* [pam_ldaphome config]
Read LDAP configuration from *file* (default – `/etc/ldap.conf`). Special value '`none`' disables this feature.

`ldap-version` *v* [pam_ldaphome config]
Sets the LDAP version to use. Valid values for *v* are '`2`' and '`3`' (the default).

`pubkey-attr` *text* [pam_ldaphome config]
Defines the name of the attribute which holds the user public key.

`tls` *val* [pam_ldaphome config]
Controls whether TLS is desired or required. If *val* is '`no`' (the default), TLS will not be used. If it is '`yes`', the module will issue the '`StartTLS`' command, but will continue anyway if it fails. Finally, if *val* is '`only`', TLS is mandatory, and the module will not establish LDAP connection unless '`StartTLS`' succeeds.

`tls-cacert` *val* [pam_ldaphome config]
`tls_cacert` *val* [pam_ldaphome config]
Full pathname to the CA certificate file. Used if TLS is enabled. The second form ('`tls_cacert`') is for use in `/etc/ldap.conf` file.

`uri` *arg* [pam_ldaphome config]
Sets the URI of the LDAP server to consult for the user profile. Example:

```
uri ldap://127.0.0.1/
```

Home directory creation

`allow-home-dir` *path* [pam_ldaphome config]
If present, this option controls where `pam_ldaphome` should try to create home directories. Its value is a list of directories separated by colons. The user's home directory will be created only if the directory part of its name is listed in *path*.

`copy-buf-size` *n* [pam_ldaphome config]
Sets the size of the buffer used to copy files from the skeleton directory to the newly created home. The default size is 16384 bytes.

`home-dir-mode` *mode* [pam_ldaphome config]
Sets the mode (octal) for the created user directories.

`skel` *dir* [pam_ldaphome config]
Supplies the name of a *skeleton directory*. The contents of this directory is copied to the newly created user home directory. The file modes and permissions are preserved.

Authorized keys file

`authorized_keys` *name* [pam_ldaphome config]
Sets the pathname (relative to the home directory) for the authorized keys file. The default is '`.ssh/authorized_keys`'. For normal operation, this value must be the same as the value of '`AuthorizedKeysFile`' variable in `sshd_config`. Unless you change the latter, there's no need to edit it.

`import-public-keys` *bool* [pam_ldaphome config]
When set to '`no`', disables importing public keys from LDAP. You may wish to use this option if you are using `openssh` 6.1 or later with `ldappubkey` as '`AuthorizedKeysCommand`'.

`keyfile-mode` *mode* [pam_ldaphome config]
Sets the mode (octal) for the created authorized keys file.

`user-keys-boundary` *string* [pam_ldaphome config]
User key files can contain both keys managed by `pam_ldaphome` and added by the user. These two groups of keys must be separated by a special comment line, which informs the module that all keys below it must be retained.

This feature is enabled by the `user-keys-boundary` setting. The delimiting comment is formed as '`#`*string*'. E.g. if the configuration file contains:

```
user-keys-boundary :user-defined
```

then the line '`#:user-defined`' can be used to delimit ldap-synchronized and user-specific keys.

Access control

`allow-groups` *group* [*group*...] [pam_ldaphome config]
Only handle members of the listed groups.

`min-gid` *n* [pam_ldaphome config]
Sets the minimal GID. For users with GIDs less than *n*, `pam_ldaphome` returns PAM_SUCCESS immediately.

`min-uid` *n* [pam_ldaphome config]
Sets the minimal UID. For users with UIDs less than *n*, `pam_ldaphome` returns PAM_SUCCESS immediately. This allows you to have a set of basic users whose credentials are kept in the system database and who will not be disturbed by `pam_ldaphome`. See also '`min-gid`' and '`allow-groups`'.

Initialization script

The following statements instruct `pam_ldaphome` to invoke an external command after initializing the user home directory. This can be used to customize the files copied from the skeleton directory according to the user.

`exec-timeout` *seconds* [pam_ldaphome config]
Sets maximum time the `initrc-command` is allowed to run. If it runs longer than *seconds*, it will be terminated with a '`SIGKILL`', and the module will return PAM_SYSTEM_ERR.

`initrc-command` *command* [pam_ldaphome config]
Run `command` after populating the user home directory with files from the skeleton directory.

The user login name is passed to the command as its argument. Before invoking, the current working directory is changed to the user home, standard input is closed, and standard output is redirected to standard errror.

The command is run under the current user privileges, unless the variable `initrc-root` is set to true.

The command should exit with code 0 on success. If it exits with a non-zero code, `pam_ldaphome` will report '`PAM_SYSTEM_ERR`'.

`initrc-root` *bool* [pam_ldaphome config]
When set to `true`, `initrc-command` will be run with root privileges. In this case, the environment variable `PAM_LDAPHOME_USER` will be initialized to the name of the user who is trying to log in.

`initrc-log` *file* [pam_ldaphome config]
This statement redirects the standard output and error from the `initrc-command` to *file*.

`initrc-environ` *env* ... [pam_ldaphome config]
Modifies the environment of `initrc-command`.

This statement takes one or more arguments. Each argument can be one of:

- (a dash) Clear the environment. This is understood only when used as the first argument.

-*name* Unset the environment variable *name*.

-*name*=*val*
Unset the environment variable *name* only if its value is *val*.

name Retain the environment variable *name*.

name=*value*
Define environment variable *name* to have given *value*.

name+=*value*
: Retain variable *name* and append *value* to its existing value. If no such variable is present in the environment, it is created and *value* is assigned to it. However, if *value* begins with a punctuation character, this character is removed from it before the assignment. This is convenient for using this construct with environment variables like `PATH`, e.g.:

```
PATH+=:/sbin
```

In this example, if `PATH` exists, '`:/sbin`' will be appended to it. Otherwise, it will be created and '`/sbin`' will be assigned to it.

name=+*value*
: Retain variable *name* and prepend *value* to its existing value. If no such variable is present in the environment, it is created and *value* is assigned to it. However, if *value* ends with a punctuation character, this character is removed from it before assignment.

The *value* part can be enclosed in single or double quotes, in which case the usual shell dequoting rules apply.

7.2 Example of pam_ldaphome configuration

This example assumes you are using GNU/Linux. The aim of this configuration is to allow remote access via sshd to users present only in the LDAP database, using ssh shared-key authentication. The exact way of achieving this depends on the version of `opennsh` daemon in use. The `openssh` version 6.2p1 introduced a possibility to obtain public keys by invoking an external command, so there are two main usage cases, as described in the subsections that follow.

7.2.1 Openssh versions prior to 6.2p1

The user public keys are kept in '`grayPublicKey`' attribute of his LDAP entry. When a user logs in for the first time, his home directory does not exist yet and consequently `sshd` is not able to verify his key. Therefore it falls back to the interactive authentication (it is supposed, of course, that '`UsePAM`' is set to '`yes`' in the sshd configuration file). The authentication stage is supposed to create user home directory, populate his `.ssh/authorized_keys` with his public keys and present user with a descriptive text prompting him to cancel his current authentication attempt and retry it again. The corresponding `pam.conf` section looks as follows:

pam.conf

```
sshd auth [success=ok try_again=1 default=die] \
    pam_ldaphome.so
```

```
sshd auth [success=done ignore=ignore default=die] \
    pam_unix.so
sshd auth [default=die]  pam_echo.so file=/etc/ldaphome.txt
```

The first line does most of the job. If `pam_ldaphome.so` succeeds in creating the user directory it will return '`try_again`'. This will cause skipping the next stack entry, so control will go to `pam_echo.so`, which will print a descriptive text from `/etc/ldaphome.txt` and exit indicating authentication failure.

The `pam_ldaphome.so` module returns '`success`' if the user who is trying to log in should not be handled by it (e.g. because his UID is less than the '`min-uid`' setting, etc.). In this case, authentication will be handled by `pam_unix.so`. This allows normal system accounts to function as usual. This is very important, because it will allow to access the machine even when the LDAP database is not available for some reason.

pam_ldaphome.conf

The `pam_ldaphome.so` configuration handles users with uids and gids greater than or equal to 1000 and pertaining to the group '`remote`'. User home dirs are populated from the `/etc/skel` directory.

```
min-uid 1000
min-gid 1000
allow-groups remote
skel /etc/skel
base dc=gnu,dc=org,dc=ua
filter (&(objectClass=posixAccount)(uid=$user))
pubkey-attr grayPublicKey
```

Schema

The LDAP schema should include an attribute to keep the user public keys. The author uses the following schema:

```
# depends upon:
#    nis.schema

# Attribute Definitions
attributetype ( 1.3.6.1.4.1.9163.2.1.0 NAME 'grayPublicKey'
    DESC 'SSH public key'
    EQUALITY caseExactIA5Match
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )
# Object Class Definitions
objectclass ( 1.3.6.1.4.1.9163.2.2.0 NAME 'grayAccount'
    DESC 'Abstraction of an employee account'
    SUP posixAccount AUXILIARY
    MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )
    MAY ( userPassword $ loginShell $ gecos $ grayPublicKey ) )
```

/etc/nsswitch.conf

The '`passwd`' and '`group`' entries in `/etc/nsswitch.conf` file should be as follows:

```
passwd:         files ldap
group:          files ldap
```

7.2.2 Openssh versions 6.2p1 and newer

Versions of `openssh` starting from 6.2p1 are able to read public keys from the standard output of an external program. This can be used to improve the configuration described in the previous subsection so that the user is not required to cancel his session upon the very first connection. To that effect, `pam-modules` includes the utility `ldappubkey`, distributed in the `examples` subdirectory (see Section 7.3 [ldappubkey], page 27). Copy that utility to a convenient location (`/usr/libexec` would be a wise choice), and add the following two lines to your `/etc/ssh/sshd_config` file:

```
AuthorizedKeysCommand           /usr/libexec/ldappubkeys
AuthorizedKeysCommandUser       nobody
```

Two points should be observed. First, the argument to `AuthorizedKeysCommand` (and all its pathname components) must be owned by root and be writable only for the owner. Second, the use of `AuthorizedKeysCommandUser` statement is mandatory. Of course, you can chose any suitable user (not necessarily '`nobody`').

After restarting `sshd`, it will invoke `ldappubkeys` on each log in attempt with the login name of the user as its argument. The utility will look up that user in the LDAP database, and if found, will print his piblic keys on its standard output. The `sshd` will then read the keys and try to authorize user against each of them. If none of the keys matches the private key supplied by the user, `sshd` will attempt public keys read from the user's `~/.ssh/authorized_keys` file (or another file, if overridden by the `AuthorizedKeysFile` statement in `/etc/ssh/sshd_config`).

Most of the configuration described in the previous subsection remains in effect. However, the authentication stack won't be invoked if `ldappubkeys` functions successfully. The `pam_ldaphome` module must be invoked as a part of '`session`' stack instead. The following example assumes it is invoked at the top of the stack:

```
sshd session [success=ignore try_again=ignore default=die] \
   pam_ldaphome.so
```

7.3 ldappubkey

The `ldappubkey` utility is a simple Perl program which takes user login name as its argument and produces on the standard output public ssh keys for that user, each on a separate line. The program is designed for use with `openssh` version 6.2p1 or higher. It is distributed in the `examples`

subdirectory and is not installed by default. The only prerequisite for its use is the `Net::LDAP` module. See Section 7.2.2 [Use of pam_ldaphome with openssh version 6.2p1], page 27, for instructions of its use.

The utility looks up for its configuration in the following files: `/etc/ldap.conf`, `/etc/ldap/ldap.conf` and `/etc/openldap/ldap.conf`. These files are tried in this order and the first one of them that exists is read.

The following configuration statements are used (all keywords are case-insensitive):

`uri` *ldap*[*si*]*://*[*name*[*:port*]] ... [ldap.conf]
Specifies the URI of the LDAP server (or servers) to connect to. The default is '`ldap://127.0.0.1`'.

`base` *dn* [ldap.conf]
Specifies the default base DN to use when performing LDAP operations. The base must be specified as a Distinguished Name in LDAP format.

`binddn` *dn* [ldap.conf]
Specifies the default DN to bind as.

`bindpw` *password* [ldap.conf]
Specifies the password to use with `binddn`.

`uid` *attr* [ldap.conf]
Defines the name of the attribute to use instead of `uid`. The LDAP record is searched using the following filter:

```
(&(objectClass=posixAccount)(attr=login))
```

`publickeyattribute` *attr* [ldap.conf]
Name of the attribute which holds the public key. Default is '`grayPublicKey`' (see [ldap-schema], page 26).

7.4 usergitconfig

The `examples` subdirectory of the `pam-modules` distribution contains a program `usergitconfig` which is designed to customize user's `.gitconfig` file using attributes from his LDAP entry.

The command reads the `.gitconfig` file and replaces any occurrence of '`${attr}`' with the value of the LDAP attribute *attr*. Not defined attributes are replaced with empty strings.

To use this utility with `pam_ldaphome`, first make sure you have Perl `Net::LDAP` module installed. Copy `usergitconfig` to some location of preference (say, `/usr/libexec`), and add the following to `pam_ldaphome` configuration file:

```
skel /etc/skel
initrc-command /usr/libexec/usergitconfig
```

The `/etc/skel` directory should contain the file `.gitconfig`. Suppose its contents is as follows:

```
[user]
        name = ${cn}
        email = ${mail}
```

Then, after successful completion of `pam_ldaphome`, the user's `.gitconfig` file will contain his real name and email set properly from the database.

For the `gituserconfig` LDAP configuration options, see [ldap.conf statements], page 28.

8 pam_umotd

The `pam_umotd` module displays a user-specific *message of the day* (MOTD). The text can be taken either from a disk file, or read from the standard output of a program launched for that purpose.

This module is Linux-specific.

The module is normally started as a part of the *session* stack, e.g.:

```
session optional pam_umotd.so file=/etc/motd
```

The `file` option specifies the file to read the MOTD from. By default the output size is limited to 2000 bytes (a usual 80x25 screen-worth of characters). If the input file is bigger than that, it will be truncated. The size limit can be controlled using the `max-size` parameter:

```
session optional pam_umotd.so max-size=1024 file=/etc/motd
```

Another safety-related parameter is `max-la`, which controls the maximum 5-minute load average, under which the message will be displayed. If the current LA is greater than this value, the module will return immediately without displaying anything[1].

The MOTD can be generated on the fly, by launching an external program and displaying its output. This allows you to create dynamic, user-specific MOTDs. To select this mode, use the `exec` parameter. The rest of arguments after this parameter are taken to be the name of the program to be run and its command line arguments. Before starting the program, the arguments undergo item expansion (see [item expansion], page 1). For example:

```
session optional pam_umotd.so max-size=1024 max-la=5.0 timeout=5 \
            exec /usr/bin/genmotd ${user} ${tty}
```

This example runs the program `/usr/bin/genmotd` passing it the user login name and the tty name as its argument. Notice the `timeout` parameter, which controls the maximum time (in seconds) the program will be allowed to run. If it runs longer than that, it will be killed. The default timeout is 10 seconds.

8.1 Summary of pam_umotd options

This section summarizes the options understood by `pam_umotd`.

`file=`*`filename`*
: Read and display text from file *filename*.

`exec`
: Execute a program and display its output. The rest of arguments after this parameter are taken to be the program name and its command line arguments. The arguments are subject to item expansion (see [item expansion], page 1). The program inherits the current environment.

[1] As of version 2.2 this functionality relies on the file `/proc/loadavg`.

`timeout=`*n*
: Limit the execution time of the program started via the `exec` option to *n* seconds. The default value is 10.

`max-size=`*n*
: Limit the output size to *n* bytes. Default is 2000.

`max-la=`*d*
: Exit immediately if the 5-minute load average is greater than or equal to *d* (a floating-point number).

9 pam_groupmember

The `pam_groupmember` module checks whether the user is member of one or more groups. Both primary and supplementary groups are checked. The list of groups to be checked is given with the `groups` option. Its argument is a comma-separated list of group names of numeric IDs, prefixed with '`+`' sign.

The module returns PAM_SUCCESS if the user is member of one of the supplied groups and PAM_AUTH_ERR on otherwise. The return value can be inverted using the `sense=deny` option.

Additionally, the module can return PAM_USER_UNKNOWN if the user is not known and PAM_AUTHINFO_UNAVAIL if unable to retrieve the user name.

The `pam_groupmember` module can be used in any PAM service stack.

9.1 Summary of `pam_groupmember` options

`groups=`*`group-list`*
: Defines groups to check against. The argument is a comma-separated list of group names or IDs. Group IDs must be prefixed with a plus sign.

`sense={allow|deny}`
: What to do on success. The value '`allow`' means to return `PAM_SUCCESS`, '`deny`' means to return `PAM_AUTH_ERR`. Default is '`allow`'.

10 How to Report a Bug

Email bug reports to bug-pam-modules@gnu.org.ua.

As the purpose of bug reporting is to improve software, please be sure to include maximum information that is needed to reproduce the bug. The information needed is:

- Version of the package you are using.
- Compilation options used when configuring the package.
- Conditions under which the bug appears.

Appendix A GNU Free Documentation License

Version 1.2, November 2002

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within

that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque

copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If

there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

   You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

   The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

   In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire

aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.1 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (C) *year your name*.
> Permission is granted to copy, distribute and/or modify this document
> under the terms of the GNU Free Documentation License, Version 1.2
> or any later version published by the Free Software Foundation;
> with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
> A copy of the license is included in the section entitled ‘‘GNU
> Free Documentation License’’.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

> with the Invariant Sections being *list their titles*, with
> the Front-Cover Texts being *list*, and with the Back-Cover Texts
> being *list*.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Concept Index

This is a general index of all issues discussed in this manual.

H

I

L

M

N

P

R

S

T

U

V

W

X