

Passwords - Community Help Wiki

16-20 minutes

Contents

1. [Grub 2 Password Protection](#)
2. [GRUB 2 Password Protection Notes](#)
3. [How It Works](#)
4. [Warnings & Cautions](#)
5. [Setting Up Password Protection](#)
 1. [Superuser & Password Designation \(Required\)](#)
 2. [Other Users \(Optional\)](#)
 3. [Protecting Menuentries](#)
 1. [Adding Protection to Ubuntu Entries](#)
 2. [Adding Protection to Other Entries](#)
 3. [Protect the Windows Recovery Partition](#)
6. [Password Encryption](#)
7. [Links](#)

Grub 2 Password Protection

GRUB 2 offers basic password protection for its menu and terminal. The user can set up passwords to protect the entire menu or specific menuentries. Passwords can be required for all users or only for selected users. The passwords can be stored in encrypted or unencrypted format. This page will detail the procedures necessary to establish and use the GRUB 2 password option.

```
Enter username:
ubuntu
Enter password:
```

In this guide, when the term "GRUB" is used it refers to GRUB 2, version 1.99 or later. Some features such as encrypted passwords may not be available in earlier versions. Users can check the version of GRUB they are using with **grub-install -V**



Note: The password security available with GRUB 2 provides basic protection to prevent an unauthorized user from gaining access to the operating system(s) via the GRUB 2 menu. Persons with physical access to the computer can gain access to the files via other methods which GRUB 2 cannot prevent.

GRUB 2 Password Protection Notes

Grub 2 can establish password requirements on:

- All menuentries
- Specific menuentries
- For specific users
 - For example, user "Jane" can access Ubuntu but not the Windows recovery mode, which is only accessible by "John", the superuser.
- The administrator must enable password protection manually by editing the GRUB 2 system files.

- Users and passwords should be identified in the `/etc/grub.d/00_header` or another GRUB 2 script file.
- Unless universal protection of all menuentries is desired, the specific entries must be identified:
 - Manually by editing the Grub 2 `/etc/grub.d/` scripts such as `10_linux` and `30_os-prober`
 - Manually by editing a custom configuration file created by the user.
 - Either of the above methods enables GRUB 2 to automatically add the password requirement to the configuration file (`grub.cfg`) whenever **update-grub** is executed.
 - Manually by editing `/boot/grub/grub.cfg`. Edits to this file will be removed when **update-grub** is run and password protection will be lost.
- If any form of GRUB 2 password protection is enabled, the *superuser*'s name and password are required to gain access to the GRUB 2 command line and menu-editing modes.
- The username and/or password do not have to be the same as the Ubuntu logon name/password.
- Unless GRUB 2's password encryption feature is used, the password is stored as plain text in a readable file. See the [Password Encryption](#) section for guidance on using this feature.

How It Works

Once a *superuser* and password are identified, the password feature is enabled the GRUB 2 menu will appear as it does normally. When a menu item requiring a password is selected, the user will be prompted to enter the correct username and password. If entered correctly, the selected menuentry will continue to boot. If incorrect, the user will be returned to the GRUB 2 menu.

If GRUB 2 is set up to boot directly to a password-protected menuentry without displaying a menu, the username/password prompt will appear and booting will not occur until they are correctly entered.

With at least one user/password identified, access to the GRUB 2 terminal and menuentry editing is restricted to the superuser.

To enable password protection, GRUB 2 requires

- A *superuser*
- The *superuser*'s password
 - With only this information, GRUB 2 will prevent menu editing and access to its terminal.
- Optionally, additional users/passwords may be added to limit access to specific items.

To protect one or more menu items:

- Each menuentry to be protected must include information on its title line as to which users should be granted access.
- In Ubuntu Precise or earlier, if no user(s) are designated for a specific menuentry, access to that entry will be given to all users.
- In Ubuntu Quantal or later, if no user(s) are designated for a specific menuentry, access to that entry will be limited to the superuser.

Warnings & Cautions

- Errors in creating a password-protected GRUB 2 menu may result in an unbootable system. To restore a system with broken passwords, access and edit the GRUB 2 configuration files using the LiveCD or another OS.
- If password protection is enabled, only the designated superuser can edit a Grub 2 menu item by pressing "e" or access the GRUB 2 command line by pressing "c".
- If GRUB 2 is set up to boot automatically to a password-protected menuentry the user has no option to back out of the password prompt to select another menuentry. Holding the SHIFT key will not display the menu in this case. The user must enter the correct username and password. If unable, the configuration files will have to be edited via the LiveCD or other means to fix the problem.
- Users experimenting with GRUB 2 passwords should keep at least one non-protected menuentry and set the timeout to at least 1 second until testing is complete. This will allow booting to correct problematic settings.

Setting Up Password Protection

There are three steps to enabling Grub 2 password protection. The authorized users must be identified, their passwords must be designated, and menu items to be protected must be identified. Users and passwords are manually added to the `/etc/grub.d/00_header` file *. The file must be edited by an Ubuntu user with administrative authority (root) since it is a system file. The user/password information is automatically added to the GRUB 2 menu configuration file (`grub.cfg`) when **update-grub** is run.

To edit the `/etc/grub.d/00_header*`, `/etc/grub.d/10_linux`, and `/etc/grub.d/30_os-prober` files, open them as root with a text editor (such as `gedit`):

- ```
gksu gedit /etc/grub.d/00_header /etc/grub.d/10_linux
/etc/grub.d/30_os-prober
```

\* The *superuser/user* information and password do not *have* to be contained in the `/etc/grub.d/00_header` file. The information can be placed in any `/etc/grub.d` file as long as that file is incorporated into `grub.cfg`. The user may prefer to enter this data into a custom file, such as `/etc/grub.d/40_custom` so it is not overwritten should the Grub package be updated. If placing the information in a custom file, do *not* include the "cat << EOF" and "EOF" lines as the content is automatically added from these files.

## Superuser & Password Designation (Required)


A superuser must be designated. This user can select all menuentries, edit any items in the GRUB 2 menu during the boot process, and access the GRUB 2 terminal.

- The superuser is identified on its own line:
  - **set supersusers="user1"** Example: `set supersusers="John"`
- The format for adding the superuser password and any additional users and passwords:
  - **password <user> <password>** Example: `password John foo`



Note: GRUB 2 1.99 in 12.04 LTS doesn't protect submenu, ie. command line, entry editing and access to entries is not protected! See [bug 718670](#). The workaround is to export supersusers variable by adding to the above `export supersusers`. The above example would look like:

```
set superusers="John"
password John foo
export superusers
```

 GRUB 2 passwords are stored as plain text in readable files. GRUB 2 can encrypt the password using grub-mkpasswd-pbkdf2. See the [Password Encryption](#) section for details.

1. Open `/etc/grub.d/00_header` and add the following at the bottom of the file.

| <b>Generic Entry - Superuser Only:</b> | <b>Example of actual entry:</b> |
|----------------------------------------|---------------------------------|
| cat << EOF                             | cat << EOF                      |
| set superusers="user1"                 | set superusers="John"           |
| password user1 password1               | password John 1234              |
| EOF                                    | EOF                             |

## Other Users (Optional)

- Other users can be identified and given a password. A designated user can access unprotected and any to which he is specifically assigned.
  - Add the following non-bold text to the bottom of `/etc/grub.d/00_header`
  - Use this entry rather than the previous example.

| <b>Generic Entry: - Superuser + 1</b> | <b>Example of actual entry:</b> |
|---------------------------------------|---------------------------------|
| cat << EOF                            | cat << EOF                      |
| set superusers="user1"                | set superusers="John"           |
| password user1 password1              | password John 1234              |

|                          |                      |
|--------------------------|----------------------|
| password user2 password2 | password Jane 5678   |
| password user3 password3 | password Sergio 9012 |
| EOF                      | EOF                  |

With only the information above entered into the `/etc/grub.d/00_header` file the GRUB 2 menu and terminal can only be accessed by the superuser. All menu items can be accessed by all users. To password-protect one or more menuentries, continue with the next section.

The GRUB 2 menu can contain protected and unprotected items. The format for protecting a menu item consists of adding the user access information to the menuentry title line.

- The presence of `--unrestricted` disables password protection.
- The presence of `--users` enables password protection for the specified users. The superuser always has access.
  - `--users ""` authorizes **only** the superuser
- `--users "Jane"` authorizes the superuser **and** "Jane"
- `--users Jane,Sergio` authorizes the superuser **and** "Jane" **and** "Sergio"

Examples:

|                                        |                                                                                                          |
|----------------------------------------|----------------------------------------------------------------------------------------------------------|
| • All Users (No menuentry protection): | menuentry 'Ubuntu, with Linux 3.2.0-24-generic' --class ubuntu<br>-class os --unrestricted {             |
| Superuser Only:                        | menuentry 'Ubuntu, with Linux 3.2.0-24-generic' --class ubuntu<br>-class os <b>--users ""</b> {          |
| Superuser + Jane:                      | menuentry 'Ubuntu, with Linux 3.2.0-24-generic' --class ubuntu<br>-class os <b>--users Jane</b> {        |
| Superuser + Jane + Sergio:             | menuentry 'Ubuntu, with Linux 3.2.0-24-generic' --class ubuntu<br>-class os <b>--users Jane,Sergio</b> { |

## Adding Protection to Ubuntu Entries

There is currently no automated method of adding users or designating menu items to be protected. The user must manually edit the GRUB 2 scripts. The GRUB 2 menu is a compilation of the inputs of several scripts. The `/etc/grub.d/10_linux` file is responsible for adding the default Ubuntu OS to the GRUB 2 menu. The instructions below will automatically add password protection to all entries in the

*10\_linux* section of the GRUB 2 menu. The majority of these sections are devoted to editing the default scripts located in the */etc/grub.d/* folder.



Another option is to create a custom menu, add the menu items you wish (including those you want to protect), and disable the standard scripts. The users and passwords can be included in this file rather than in the *00\_header* file. This may be an easier method of assigning password protection - especially if only some of the menu items will be protected. The formatting is the same as described on this page, except the "cat << EOF" and "EOF" entries are not used. For more information on how to set up a custom menu, refer to [Grub 2/CustomMenus](#).



Before making these changes, it is recommended to save a copy of the */etc/grub.d/10\_linux* file to another location to serve as a backup. Do not leave the copy in the */etc/grub.d/* folder to ensure it is not run during updates.

In */etc/grub.d/10\_linux*, find the following line:

- `printf "menuentry '${title}' ${CLASS} {\n" "${os}" "${version}"`

Add **--users ' '**:

- `printf "menuentry '${title}' ${CLASS} --users ' ' {\n" "${os}" "${version}"`

Save the file, then run:

- `sudo update-grub`

The **--users ' '** tag will be added to every menuentry located in the "10\_linux" section of *grub.cfg*.



Note:

This document previously stated that you should add **--users ""** (with quotes instead of apostrophes) to the printf line above:

- `printf "menuentry '${title}' ${CLASS} --users "" {\n" "${os}" "${version}"`

However, doing so causes *grub.cfg* to be generated without the quotes after **--users**, which may cause GRUB to fail with a cryptic "alloc magic is broken" error message.

## Adding Protection to Other Entries

The GRUB 2 menu includes selections for operating systems other than the default Ubuntu OS via the */etc/grub.d/30\_os-prober* script. The instructions below will automatically add password protection to all entries in the *30\_os-prober* section of the GRUB 2 menu. This would include other Ubuntu and Linux installations, Windows, etc.

The *30\_os-prober* script looks for specific types of operating systems. The user can make the changes for all of the OS's via a single command, or individually by OS type as documented below.



Before making any changes the user should save a copy of the */etc/grub.d/30\_os-prober* file to another location to serve as a backup. Do not leave the copy in the */etc/grub.d/* folder to ensure it is not run during updates.

**All 30\_os=prober Entries:** The user can add password protection to all entries generated by */etc/grub.d/30\_os-prober* by running the following command which adds **--users** to each menuentry found by the script. Alternatively, the user can use the expanded entries to change only specific operating systems as detailed below.

To back up and then alter the */etc/grub.d/30\_os-prober* to add password protection to all entries:

- ```
sudo cp /etc/grub.d/30_os-prober /root/Desktop/30_os-prober
sudo sed 's/--class os /--class os --users /' -i /etc/grub.d/30_os-prober
```

Select OS Entries:

To enable password protection only on a specific type of operating system add **--users** immediately following "--class os".

OSX Example: menuentry "\${LONGNAME} (\${2}-bit) (on \${DEVICE})" --class osx --class darwin --class os **--users** {

- OSX:
 - menuentry "\${LONGNAME} (\${2}-bit) (on \${DEVICE})" --class osx --class darwin --class os {
- Windows:
 - menuentry "\${LONGNAME} (on \${DEVICE})" --class windows --class os {
- Linux/Ubuntu:
 - menuentry "\${LLABEL} (on \${DEVICE})" --class gnu-linux --class gnu --class os {
- Hurd:
 - menuentry "\${LONGNAME} (on \${DEVICE})" --class hurd --class gnu --class os {

Save the file, then run:

- ```
sudo update-grub
```

## Protect the Windows Recovery Partition

It is possible to provide password protection only to a specific partition. One obvious example would be the Windows recovery partition. While the concept could also be applied to other OS sections in the *30\_os-prober* file or even the *10\_linux* script, only the lines which search for the Windows OS will be described. Any Windows partition could be protected in the same manner by designating the partition.

Note: This technique will work only if the GRUB 2 menu identifies multiple Windows partitions and one of them is the recovery partition. If only one Windows partition is identified by GRUB due to Windows chainloading it's menus, only Windows in its entirety could be protected.

If the user would like only to place password protection on a Windows recovery partition, follow the previous guidance and then:

1. Determine the Windows Recovery partition (sda1, sda2, etc). Change sd**XY** to the correct values.
2. In the */etc/grub.d/30\_os-prober* file:
  - Change:

|   |                                                                       |
|---|-----------------------------------------------------------------------|
| ■ | cat << EOF                                                            |
|   | menuentry "\${LONGNAME} (on \${DEVICE})" --class windows --class os { |
|   | EOF                                                                   |

To:

|   |                                                       |
|---|-------------------------------------------------------|
| ■ | if [ \${DEVICE} = "/dev/sdXY" ]; then                 |
|   | cat << EOF                                            |
|   | menuentry "\${LONGNAME} (on \${DEVICE})" --users "" { |
|   | EOF                                                   |
|   | else                                                  |
|   | cat << EOF                                            |
|   | menuentry "\${LONGNAME} (on \${DEVICE})" {            |
|   | EOF                                                   |
|   | fi                                                    |

3. Save the file, then run:

```
sudo update-grub
```

## Password Encryption

**grub-mkpasswd-pbkdf2** Encrypted password protection has been available in all versions of Grub 2 but was improved in GRUB 1.99.



One of the drawbacks of the password setup discussed so far is that the passwords are entered in plain text in the GRUB 2 files. The degree of security can be greatly enhanced by using Grub 2's *grub-mkpasswd-pbkdf2* command. This command converts your desired password into a very long alphanumeric code which is placed in the GRUB 2 files. Your actual password is no longer visible in the Grub 2 scripts. While physical access to a computer can bypass the GRUB 2 menu, encryption makes it much more difficult for the casual hacker to determine your menu passwords.



It is worth repeating: Users experimenting with GRUB 2 passwords should keep at least one non-protected menuentry and set the timeout to at least 1 second until testing is complete. This will allow booting a menuentry without a password to correct problematic settings.

```
ubuntu@ubuntu:~$ grub-mkpasswd-pbkdf2
Enter password:
Reenter password:
Your PBKDF2 is grub.pbkdf2.sha512.10000.FC58373BCA15A797C418C1EA7FFB85B9A21798D94B007BF5A57
9449728ADF249EABE1511C7B4277CB354092C0568E9008C304384D23F7B62F767.E657080F51EC8DE44B7053122
13BA9B59B1290013B92B68DAED9B45462E109F40CA6A935C263A4D87575302FF368036B4D73321DFC566C5697CA
```

- To generate an encrypted password, open a terminal and run the following command:

- `grub-mkpasswd-pbkdf2`

- Enter the desired password and reenter it when prompted.
  - Copy the resulting code. In a terminal, highlight the code and CTRL-SHIFT-c to place it in memory. If highlighting the output with a mouse, it may be automatically stored in memory and be pasted by clicking the middle mouse button.
  - Paste the code after the username(s). Pasting can be accomplished in a text editor by either CTRL-v or middle mouse click.
  - The format for an encrypted password entry in `/etc/grub.d/00_header` would look similar to the following (shortened in the example):

|   |                                                                                       |
|---|---------------------------------------------------------------------------------------|
| ■ | set superusers="John"                                                                 |
|   | password_pbkdf2 John<br>grub.pbkdf2.sha512.10000.FC58373BCA15A797C418C1EA7FFB007BF5A5 |

## Links

[Grub2](#)

[Grub2/Installing](#)

[Grub2/Troubleshooting](#)

[Grub2/Upgrading](#)

External Links:

[GNU GRUB Manual](#)