

Ask Ubuntu is a question and answer site for Ubuntu users and developers. It's 100% free.

Sign up ×

How can I prevent someone from resetting my password with a Live CD?

Recently one of my friends came over to my place, within 15 minutes he hacked my account using a Live CD and reset the password in front of me. I was baffled to see such a thing. Please guide me to prevent such a future attempt using a Live CD.

security password live-cd

edited Nov 19 '11 at 9:56

 **N.N.**
5,059 10 37 78

asked Nov 8 '11 at 3:08

 **coder**
1,227 2 14 30

7 Unplug the optical disc drive. – **Anonymous** Nov 10 '11 at 23:44

5 It might interest you to know that it's possible to do the same thing in Windows; when I worked in IT at a large university, I had a CD with me to allow me to do so if needed. No computer is really secure if one can boot from other media. – **Kelley** Nov 12 '11 at 0:30

5 Answers

A quick and easy way to do this is to disable booting from CDs and USB sticks in your BIOS and set a BIOS password.

According to this [wiki page](#):

Placing passwords or locking menu items (in the Grub configuration files) does not prevent a user from booting manually using commands entered at the grub command-line.


However there's nothing stopping someone from just stealing your hard drive and mounting it on another machine, or resetting your BIOS by removing the battery, or one of the other methods that an attacker can use when they have physical access to your machine.

A better way would be to encrypt your drive, you can either do this by encrypting your home directory, or encrypting the entire disk:

- [How do I encrypt my home partition?](#)
- <https://help.ubuntu.com/community/FullDiskEncryptionHowto>

edited Nov 12 '11 at 0:11

answered Nov 8 '11 at 3:11

 **Jorge Castro**
25.4k 83 361 566

2 That is not sufficient—you have to also disable recovery mode and lock down GRUB2 so that boot options cannot be specified (or cannot be specified without entering a password). Once all this is done, not only does it not stop someone from stealing the hard drive, but someone who opens the computer can disable the BIOS password, and someone who doesn't want to open the computer can presumably just steal the whole computer. – **Elijah Kagan** Nov 8 '11 at 3:13

but buddy what if someone just took my hard disk opened **/etc/shadow** changed my encrypted password and then rebooted the the encrypted home directory will also open for him – **coder** Nov 8 '11 at 3:14

6 @shariq If someone changes your password just in **/etc/shadow** then the encrypted home directory will *not* open when someone logs on with the new password. In that case, the encrypted home directory would have to be mounted manually with the old password, and if nobody knew the old password, it would have to be cracked, which would be difficult and somewhat likely to fail. When you encrypt your home directory, changing your password by editing **/etc/shadow** does *not* change the password with which your data are encrypted. – **Elijah Kagan** Nov 8 '11 at 3:16

@ElijahKagan I can't any information on how to do that, so I've quoted the wiki page, if you find any more information feel free to edit it into my answer. – [Jorge Castro](#) Nov 8 '11 at 3:17

- 5 +1 The very first thing that came to my mind was encrypt the home directory. – [Nathan Osman](#) Nov 8 '11 at 3:24

Stand next to your computer while holding a tee-ball bat. Severely beat anyone who gets close.

Or lock it up.

If your computer is physically accessible, it is unsafe.

answered Nov 10 '11 at 18:02



[mdebusk](#)

267 2 2

- 11 How does this protect us from the men with big dogs and machine guns? :D – [jrg](#) ♦ Nov 10 '11 at 22:00

- 1 secure computer with dogs and use security cameras to protect dogs and in next room motion-guns – [Kangaroo](#) Nov 10 '11 at 22:38

- 1 +1 for the seriousness of this answer. You really did a good job here and deserve the votes. – [RolandiXor](#) ♦ Nov 12 '11 at 6:15

I wish, I could give more answer like this one :) – [Anwar Shah](#) Sep 28 '12 at 5:18

+1 for the great answer and the comments.. I simply couldn't stop my self to laugh loudly!! :D :) @jrg was at his best.. ha ha ha.. :) Good to see this type thing in askubuntu. – [Saurav Kumar](#) Oct 20 '13 at 1:04

First the warning...

The grub2 password protection procedure can be quite tricky and if you get it wrong there is a possibility of leaving yourself with a non-bootable system. Thus **always** make a full image backup of your hard-drive first. My recommendation would be to use [Clonezilla](#) - another backup tool such as [PartImage](#) could also be used.

If you want to practice this - use a virtual machine guest which you can rollback a snapshot.

let's begin

The procedure below protects unauthorised editing of Grub settings whilst booting - that is, pressing `e` to edit allows you to change the boot options. You could for example, force booting to single user mode and thus have access to your hard-disk.

This procedure should be used in conjunction with hard-disk encryption and a secure bios boot option to prevent booting from live cd as described in the associated answer to this question.

almost everything below can be copied and pasted one line at a time.

First lets backup the grub files we will be editing - open a terminal session:

```
sudo mkdir /etc/grub.d_backup
sudo cp /etc/grub.d/* /etc/grub.d_backup
```

Lets create a username for grub:

```
gksudo gedit /etc/grub.d/00_header &
```

Scroll to the bottom, add a new empty line and copy and paste the following:

```
cat << EOF
set superusers="myusername"
password myusername xxxx
password recovery 1234
EOF
```

In this example two usernames were created: *myusername* and *recovery*

Next - navigate back to the terminal (don't close `gedit`):

Natty and Oneiric users only

Generate an encrypted password by typing

```
grub-mkpasswd-pbkdf2
```

Enter your password you will use twice when prompted

```
Your PBKDF2 is
grub.pbkdf2.sha512.10000.D42BA2DB6CF3418C413373CD2D6B9A91AE4C0EB4E6AA20F89DFA027CA6E
```

The bit we are interested in starts `grub.pbkdf2...` and ends `BBE2646`

Highlight this section using your mouse, right click and copy this.

Switch back to your `gedit` application - highlight the text "xxxx" and replace this with what you copied (right click and paste)

i.e. the line should look like

```
password myusername
grub.pbkdf2.sha512.10000.D42BA2DB6CF3418C413373CD2D6B9A91AE4C0EB4E6AA20F89DFA027CA6E
```

all 'buntu versions (lucid and above)

Save and close the file.

Finally you need to password protect each grub menu entry (all files that have a line that begins *menuentry*):

```
cd /etc/grub.d
sudo sed -i -e '/^menuentry /s/ {/ --users myusername {/' *
```

This will add a new entry `--users myusername` to each line.

Run `update-grub` to regenerate your grub

```
sudo update-grub
```

When you try to edit a grub entry it will ask for your user name i.e. *myusername* and the password you used.

Reboot and test that username and password is being enforced when editing all of the grub-entries.

N.B. remember to press SHIFT during boot to display your grub.

Password protecting recovery mode

All of the above can easily be workaroud by using recovery mode.

Fortunately you can also force a username and password to use the recovery-mode menu entry. In the first part of this answer we create an additional username called *recovery* with a password of *1234*. To use this username we need to edit the following file:

```
gksudo gedit /etc/grub.d/10_linux
```

change the line from:

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

To:

```
if ${recovery} ; then
    printf "menuentry '${title}' --users recovery ${CLASS} {\n" "${os}" "${version}"
else
    printf "menuentry '${title}' ${CLASS} {\n" "${os}" "${version}"
fi
```

When using recovery use the username *recovery* and the password *1234*

Run `sudo update-grub` to regenerate your grub file

Reboot and test that you are asked for as username and password when trying to boot into recovery mode.

More Information - <http://ubuntuforums.org/showthread.php?t=1369019>

edited Nov 11 '11 at 13:22

answered Nov 10 '11 at 21:24



fossfreedom ♦

114k 23 255 307

Hi! I followed your tutorial and it works... except if you choose other versions, where you can use "E" key without password – [gsedej](#) Sep 20 '12 at 13:55

Raring doesn't have the line `printf "menuentry '${title}' ${CLASS} {\n" "${os}" "${version}"` for recovery but a similar one inside an if function. Could you advice how to edit it? – [papukaija](#) Jun 22 '13 at 1:42

It's important to remember that if someone has physical access to your machine, they will always be able to do things to your PC. Things like locking your PC case and BIOS passwords won't stop a determined person from taking your hard drive and data anyway.

answered Nov 11 '11 at 17:42



balloons

1,172 4 25

- 2 In some circumstances, these things *will* stop even a determined person from taking your hard drive and data way. For example, if it is a kiosk machine in an Internet cafe with good physical security (security cameras, security guards, watchful proprietor/clerks), a determined person might still try to physically steal the machine or its hard drive, but they would likely fail. – [Elijah Kagan](#) Nov 11 '11 at 19:41
- 2 As a separate point, if you steal the hard drive from a machine whose data are encrypted, then you would have to crack the encryption to access the data, and with good quality passwords, doing so might be prohibitively difficult...or perhaps even impossible. – [Elijah Kagan](#) Nov 11 '11 at 19:41

You can make it so that even in a case of resetting, the "resetter" won't be able to see the data.

To do this, just encrypt `/home`.

If you want to make it so that resetting isn't possible, something needs to be removed, which is in charge of changing the password.

edited Nov 12 '11 at 6:12

answered Nov 10 '11 at 22:52



Rolandixor ♦

392k 16 102 199



Kangaroo

1,202 2 13 28