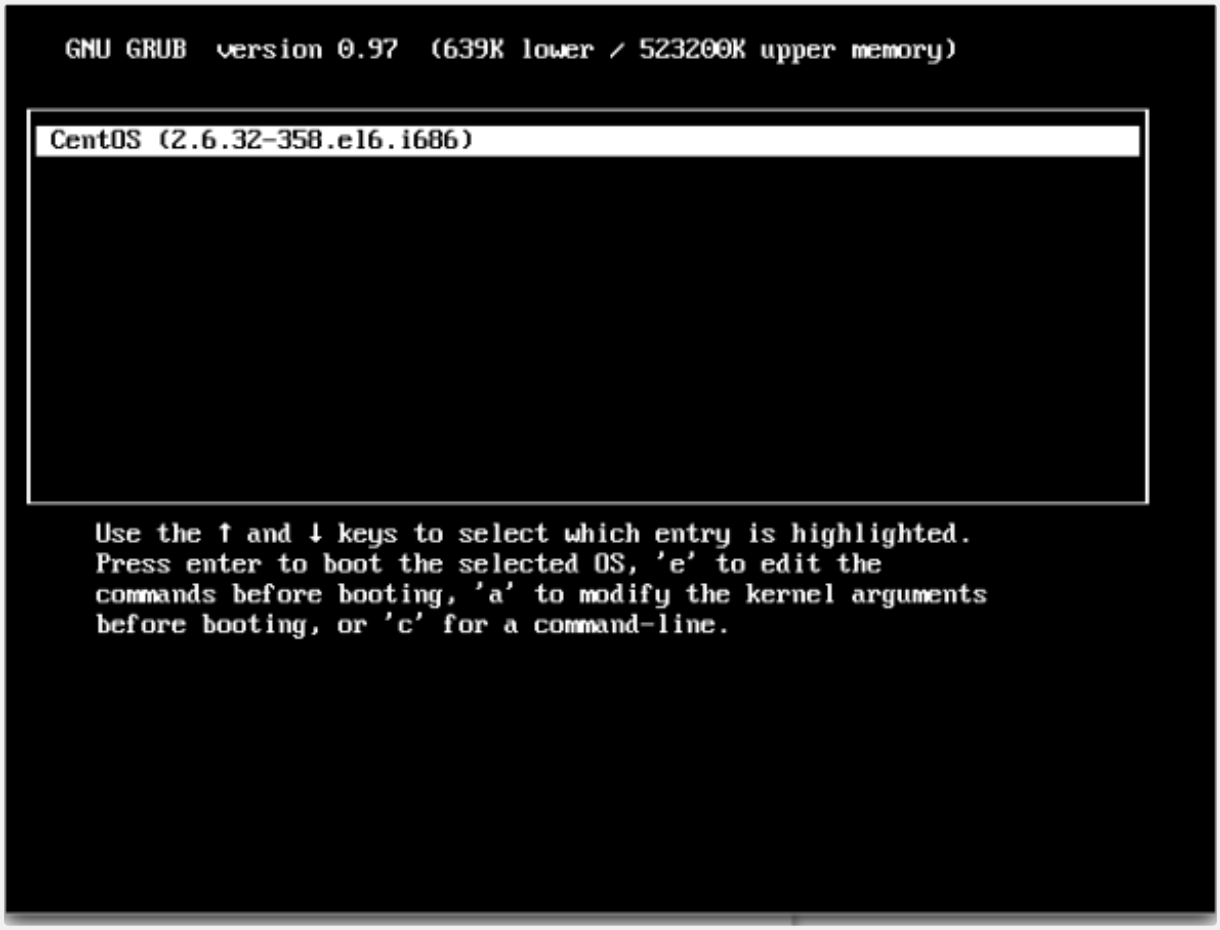
How to Hack Your Own Linux System

by Editor | Published: June 10, 2013 | Last Updated: January 7, 2015

Passwords are the sole criteria of system **Security** for most of the **System**. And when it comes to **Linux**, if you know the **root password** you owns the machine. **Passwords** are as a **Security** measure for **BIOS**, **Login**, **Disk**, **Application**, etc.

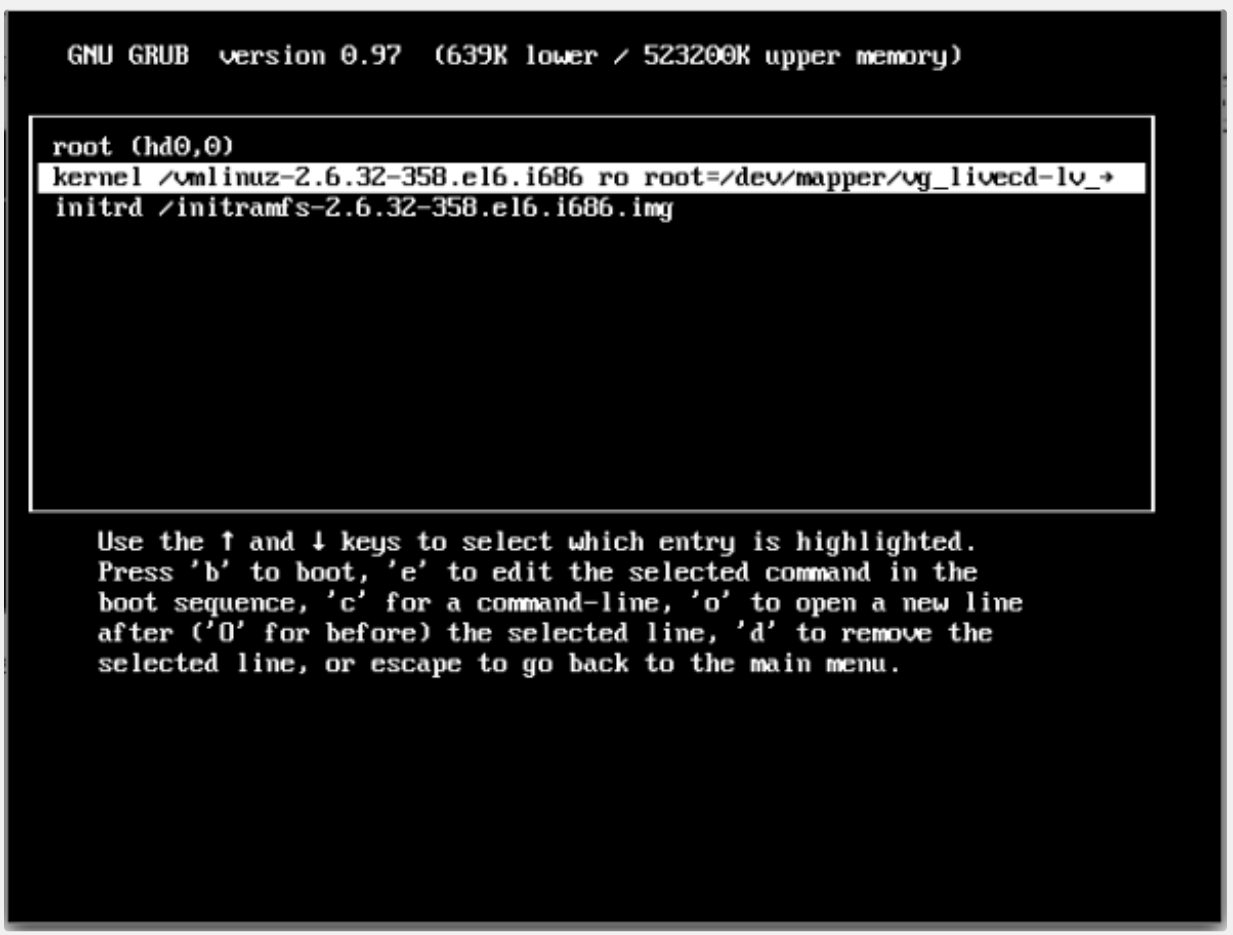
Linux is considered to be the most **Secure Operating System** to be hacked or cracked and in reality it is, still we will be discussing some of the loop-holes and exploits of a **Linux System**. We will be using **CentOS Linux** throughout the article as an article to crack our own machine's security.

Press any key to interrupt the boot, as soon as **Linux** machine boots and you will get a **GRUB** menu.



Linux Boot Screen

Press '**e**' to edit and go to the line starting with kernel (**Generally 2nd Line**).



Switch to Single User Mode

Now press '**e**' to edit the kernel and add '**1**' at the end of line (after one blank space) forcing it to start in single user mode and thus prohibiting it to enter default run-level. Press '**Enter**' to close the kernel editing and then boot to the altered option. For booting You need to press '**b**'

```
[ Minimal BASH-like line editing is supported.  For the first word, TAB
  lists possible command completions.  Anywhere else TAB lists the possible
  completions of a device/filename.  ESC at any time cancels.  ENTER
  at any time accepts your changes.]

<6 crashkernel=auto  KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet 1
```

Logged into Single User Mode

Now you are logged in to **single-user** mode.

```
Telling INIT to go to single user mode.
init: rc main process (971) killed by TERM signal
[root@localhost /]# _
```

Set root Password

Yeah! Now using '**passwd**' command we can change the **root password**. And once you have root password you owns the **Linux Machine** – Don't you Remember? You can now switch to graphical screen to edit anything and everything.

```
Telling INIT to go to single user mode.
init: rc main process (971) killed by TERM signal
[root@localhost /]# passwd
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost /]# _
```

Add new root Password

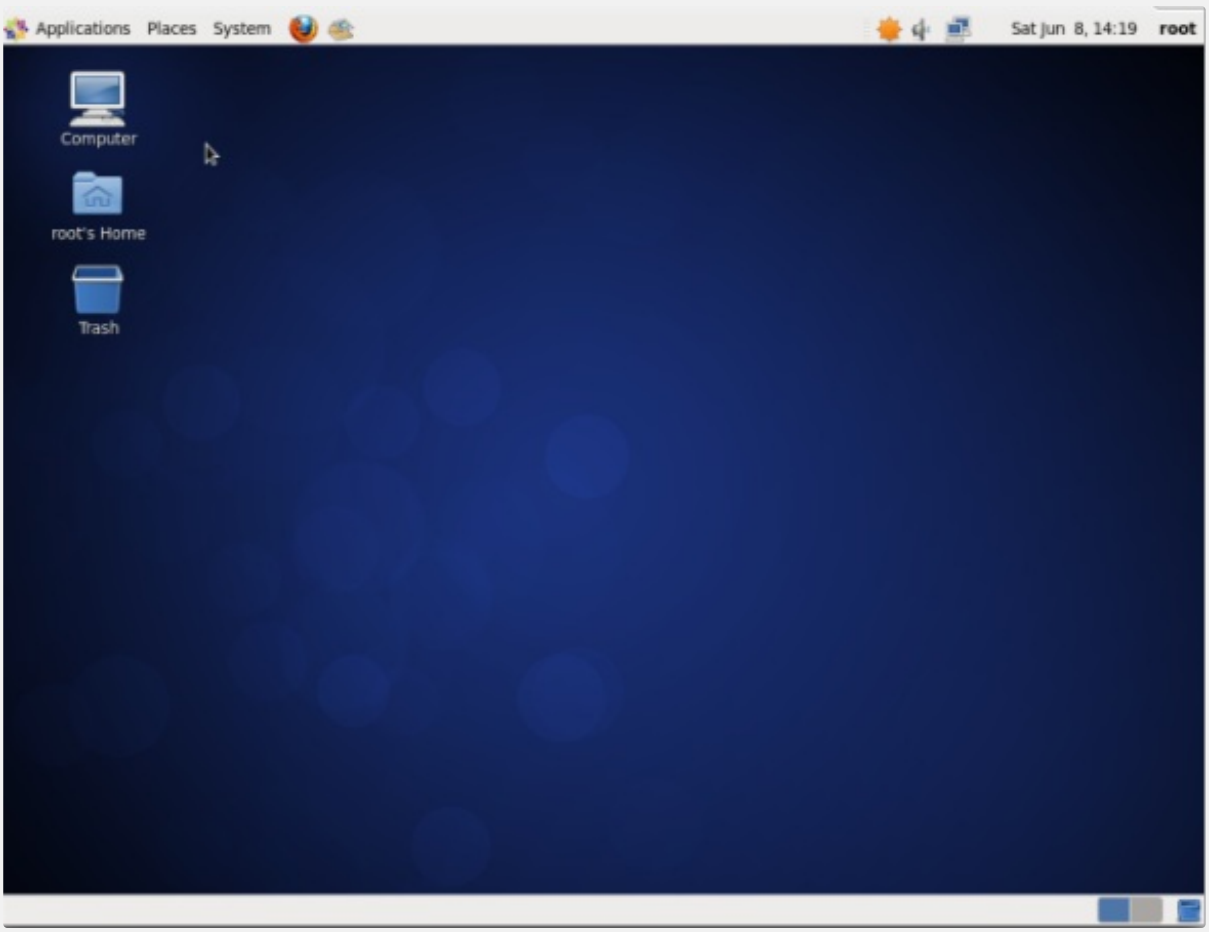
Note: In case the above '**passwd**' command doesn't work for you and you didn't get any output, it simply means that your **SELinux** is in enforcing mode and you need to disable it first, before proceeding further. Run following command at your prompt.

```
# setenforce 0
```

An then run the '**passwd**' command, to change **root password**. Moreover command.

Switch to X Windows

Use command "**init 5**" (**Fedora** Based) systems and "**gdm3**" (**Debian** Based) systems.



Switch to X Window

So was this not a cake-walk to hack a **Linux box**? Think about the scenario if somebody did this to your server, **Panic**! Now we will be learning how to safeguard our **Linux Machine** from being modified using single user mode.

How we breaked into the system? Using **Single-user** mode. **OK**, so the loophole here was – logging into single user mode without the need of entering any password.

Fixing this loophole i.e., **password protecting** the **single user** mode.

open file "**/etc/rc1.d/S99single**" in your favourite editor and search for line.

```
exec init -t1 s
```

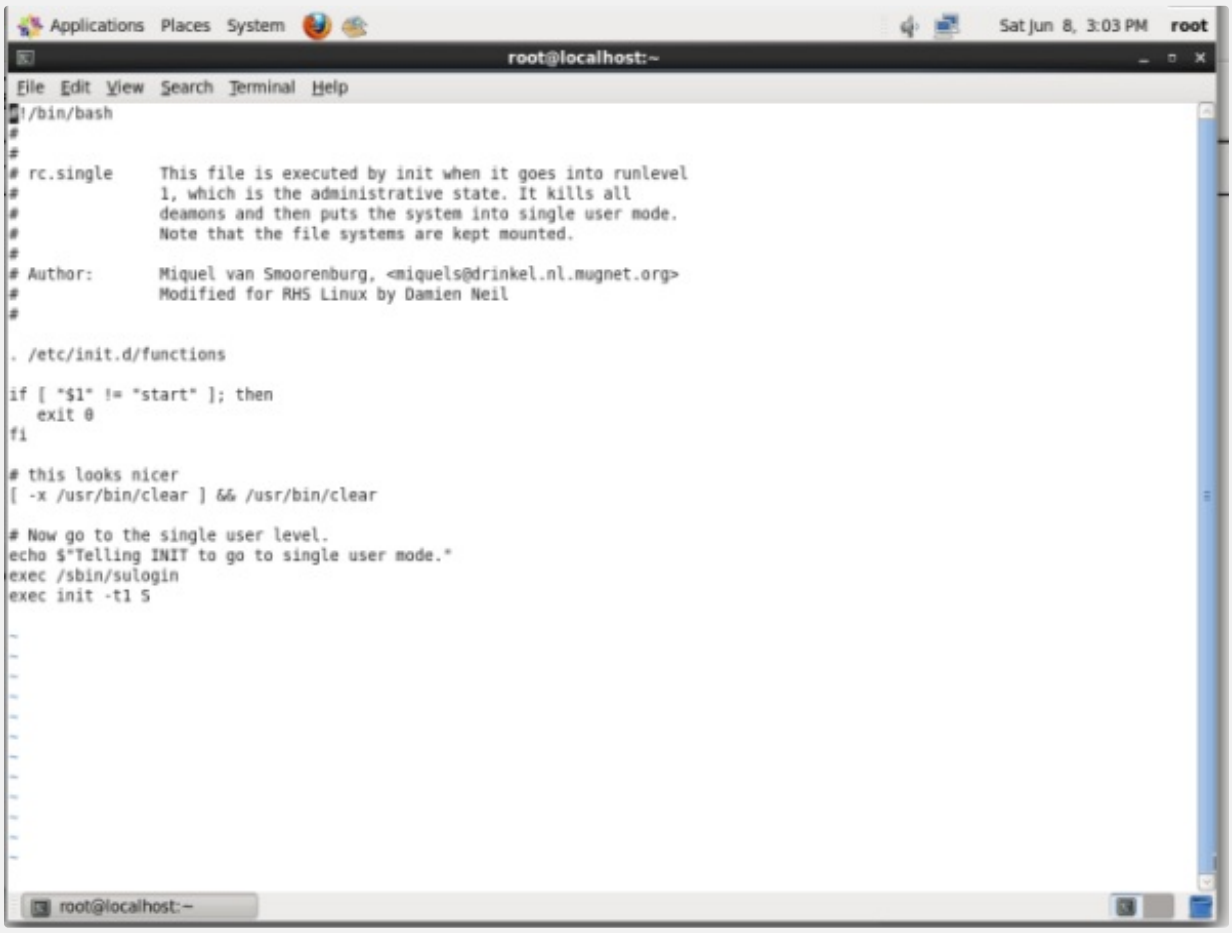
Just add the following line above it. save it an exit.

```
exec sbin/sulogin
```

Before

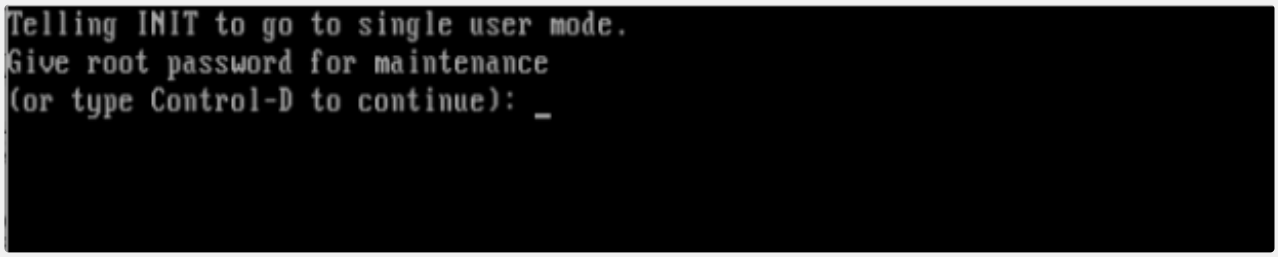
Before Preview

After



After Preview

Now before entering single user mode you will need to provide root password to proceed. Check again trying to enter single user mode after these changing above said file.



Enter Root Password for Single User Mode

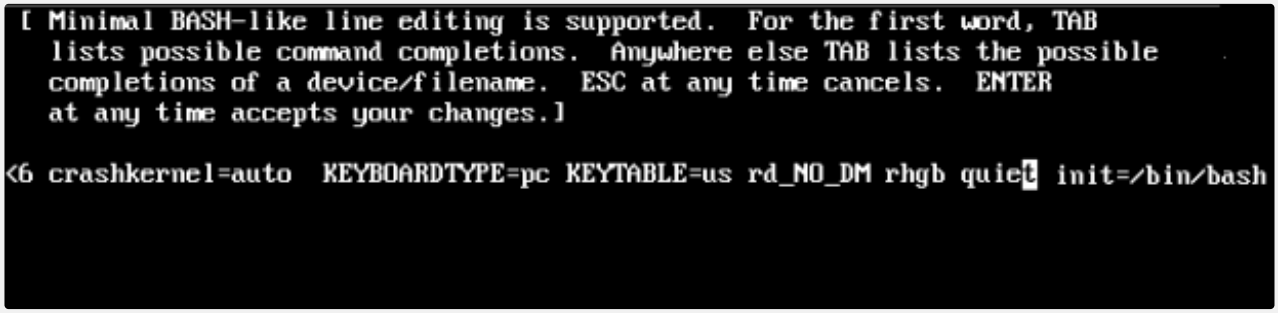
Why don't you check it, Yourself.

Hack Your Linux System Without Using Single User Mode

OK, so now you will be feeling better that your system is secure. However this is partially true. It is true that your **Linux Box** can't be cracked using single user mode but still it can be hacked the other way.

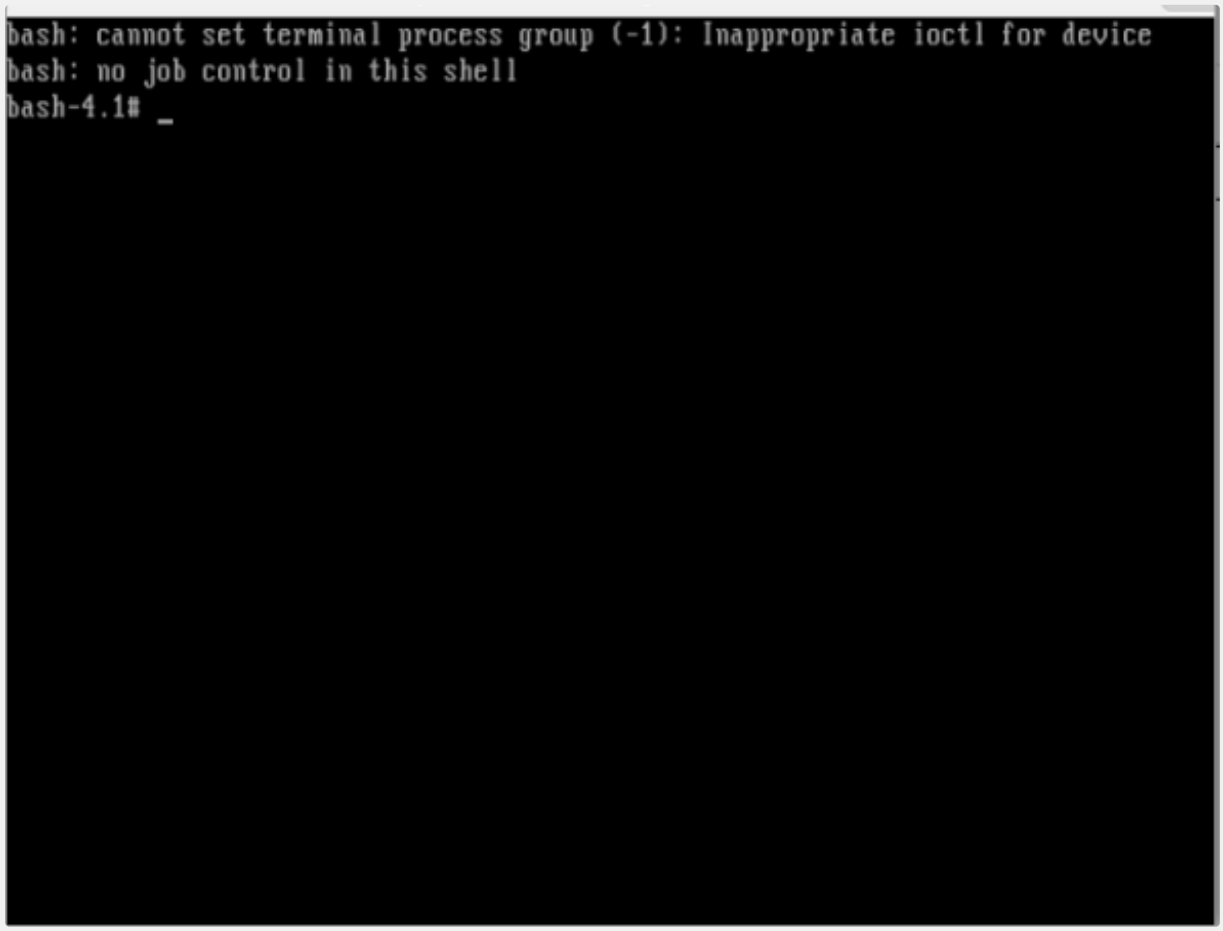
In the above step we modified the kernel to enter single user mode. This time also we will be editing the kernel but with a different parameter, let us see how ?

As a kernel parameter we added '**1**' in the above process however now we will be adding '**init=/bin/bash**' and boot using '**b**'.



Add 'init=/bin/bash'

And **OOPS** you again hacked into your system and the prompt is enough to justify this.



Hacked into Your System

Now Trying to change the **root password** using the same process as stated in the first method using '**passwd**' command, we got something like.

```
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
bash-4.1# passwd
Changing password for user root.
New password:
Retype new password:
passwd: Authentication token manipulation error
bash-4.1# _
```

Changing Root Password

Reason and Solution?

- **Reason**: The root (/) partition is mounted **Read only**. (Hence password was not written).
- **Solution**: Mount the root (/) partition with **read-write** permission.

To mount the **root partition** with **read-write** permission. Type the following command exactly.

```
# mount -o remount,rw /
```

```
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
bash-4.1# passwd
Changing password for user root.
New password:
Retype new password:
passwd: Authentication token manipulation error
bash-4.1# mount -o remount,rw /
bash-4.1# _
```

Mount / Partition in Read Write

Now again try to change the password of root using '**passwd**' command.

```
bash-4.1# mount -o remount,rw /
bash-4.1# passwd
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
bash-4.1# _
```

Change Password of root

Hurrah! You hacked into your **Linux System** once again. **Ohhh** man is the system so easy to exploit. **No!** the answer is no. All you need is to configure your system.

All the above two process involved tweaking and passing parameters to kernel. So if we do something to stop kernel tweaking obviously our Linux box would be Secure and not that easy to break. And in order to stop kernel editing at boot we must provide password to **boot loader**, i.e., **password protect** the **grub** (**Lilo** is another bootloader for **Linux** but we won't be discussing it here) boot loader.

Provide encrypted password to **bootloader** using '**grub-md5-crypt**' followed with your password. First encrypt the password

```
bash-4.1# grub-md5-crypt
Password:
Retype password:
$1$t8JvC1$8buXiBsfANd79/X3eIp9G1
bash-4.1#
```

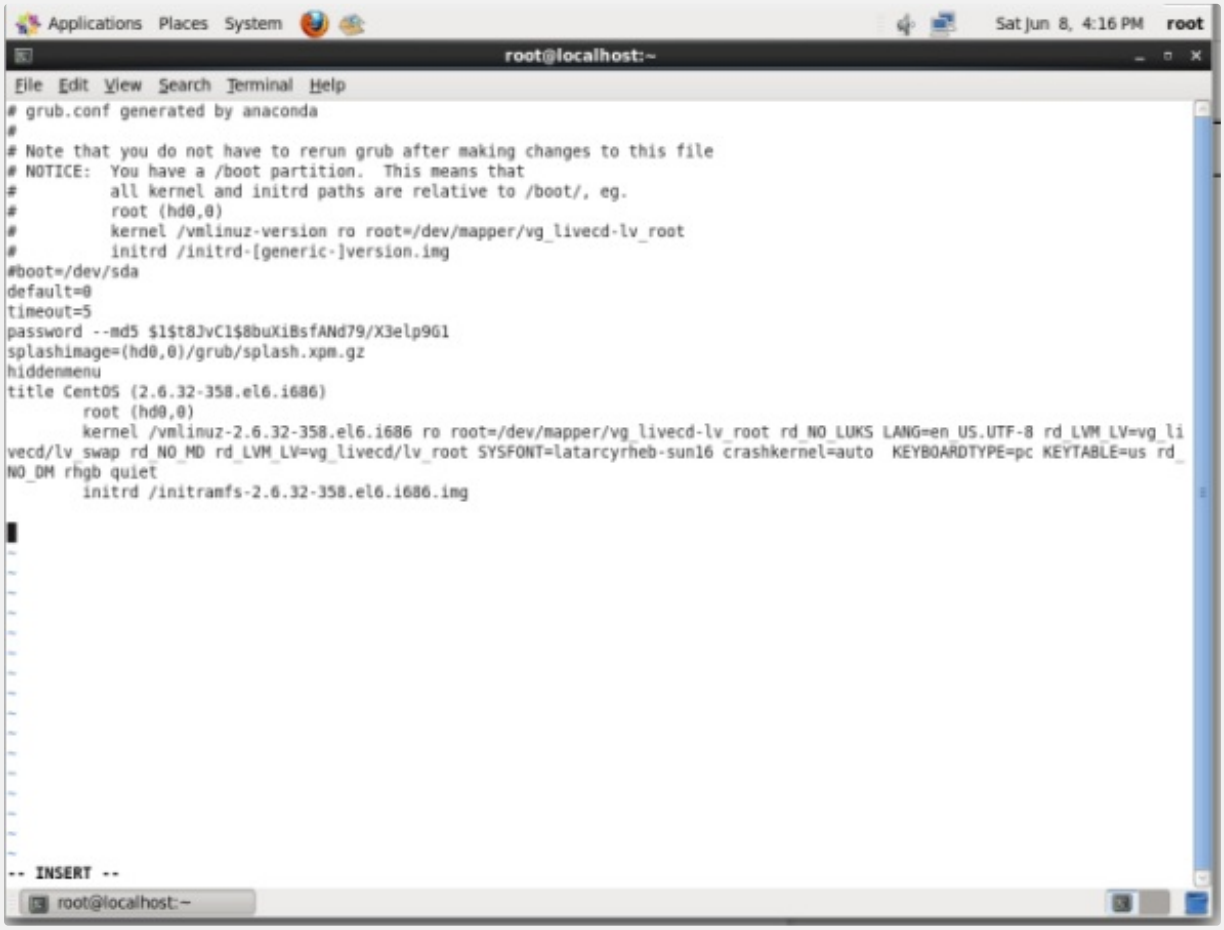
Password Protect Boot Loader

Copy the above encrypted password, exactly as it is and keep it safe we will be using it in our next step. Now open your '**grub.conf**' file using your favourite editor (location might be: **/etc/grub.conf**) and add the line.

```
password --md5 $1$t8JvC1$8buXiBsfANd79/X3elp9G1
```

Change "**1t8JvC1$8buXiBsfANd79/X3elp9G1**" with your encrypted password which you generated above and copied it safely to some other location.

The "**grub.conf**" file after inserting the above line, save and exit.



grub.conf Preview

Now Cross Checking, editing the kernel at boot, we got.

Cross Cheking Boot Loader

Now you would be breathing that you system is fully secure now and not prone to hack, however still the game is not over.

You better know that you can **enforce rescue mode** to **remove** and **modify** the password using a bootable image.

Just put your installation **CD/DVD** in your drive and select **Rescue Installed System** or use any other rescue image, you could even use a **Live Linux Distro**, mount the **HDD** and edit the '**grub.conf**' file to remove password line, reboot and again you are logged in.

Note: In **rescue mode** Your **HDD** is mounted under '**/mnt/sysimage**'.

```
# chroot /mnt/sysimage
# vi grub.conf (remove the password line)
# reboot
```

I know you would be asking- so where is the end. Well i would say is to.

- Password protect your **BIOS**.
- Change you **Boot** order to **HDD** first, followed by rest (**cd/dvd**, **network**, **usb**).
- Use Password sufficiently **Long**, **Easy** to remember, **Hard** to guess.
- Never write **Your Password** to anywhere.
- Obviously use **Uppercase**, **Lowercase**, **Numbers** and **Special Character** in your **password** thus making it hard to break.

This guide was just to make you aware of facts and tell you how to secure your System. **Tecmint.com** and the **writer** of this article strongly discourage this guide as a base of exploiting other's system. It is the sole responsibility of the reader if they engage in any such activity and for such kind of act neither the write nor **Tecmint.com** will be responsible.

Your **positive comments** makes us **feel good** and **encourages us** and that is always sought from you. **Enjoy** and **Stay Tuned**.

If You Appreciate What We Do Here On TecMint, You Should Consider:

1. Stay Connected to: Twitter | Facebook | Google Plus
2. Subscribe to our email updates: Sign Up Now
3. Use our Linode referral link if you plan to buy VPS (it starts at only $10/month).
4. Support us via PayPal donate - Make a Donation
5. Support us by purchasing our premium books in PDF format.
6. Support us by taking our online Linux courses

We are thankful for your never ending support.

NEXT STORY

The Power of Linux "History Command" in Bash Shell

PREVIOUS STORY
Install Scalpel (A Filesystem Recovery Tool) to Recover Deleted Files/Folders in Linux

YOU MAY ALSO LIKE...

Trouble Maker – Breaks Your Linux Machine and Ask You to Fix Broken Linux

10 DEC, 2013

Create Your Own Online Photo Gallery Albums Using Plogger

7 OCT, 2013

50 RESPONSES

Comments 14 Pingbacks 0

DJ June 8, 2016 at 12:13 am

Of course, physical access to a computer running ANY operating system is known to all security specialists to be a path to owning the machine. NOT NEWS.

Reply

Anwar June 4, 2016 at 2:08 pm

[ask]
INIT: Id "x" respaawning too fase: desable for 5 minute

Reply

Dipesh February 17, 2016 at 11:33 am

Awesome …Chains of Hacking tricks :)

Reply

Abdullah November 23, 2015 at 11:16 pm

Actually by passing the permission and re-setting the root password is a common practice, remember Linux is used to operate 24/7 as a server most of the time, hence no much of bios access.
But, if you own the machine you MUST encrypt it to protect your privacy (specially laptops), you have the encryption check-box when you install the system.
The bios is not our biggest problem but the external bootable device really.. solution is to encrypt.

I also encourage you to put a short Bios passcode, the idea of this short code is to know if someone accessed your machine, it is not a secret code, because you could easily reset it by cutting the current "the CMOS battery".

Reply

dan July 30, 2015 at 2:09 am

are you saying that on a standard ubuntu install with boot luks encryption and home directory encryption options checked during the install process that you can still get into the system? can both the boot and home directory passwords be bypassed easily unless i make further changes?

Reply

Avishek Kumar July 31, 2015 at 3:57 pm

I don't think it will work with boot LUKS encryption and home directory encryption, though i have not checked it personally.

Reply

ahmed May 19, 2015 at 3:26 pm

i am working on the operating system that acts the best possible window as per the req of user and used the resources as per the req of user and kill the extra things secondly its also have an other feature …. its run application automatically as user login with his articifial intelligence

Reply

Pim Dennendal January 13, 2015 at 9:33 pm

Interesting article on resetting your own root-password.

You missed one boot command-line parm which I find exceptionally usefull. It is "?init-/bin/sh". Excellent for getting into the pre-execution environment. This is usefull for examining the boot script(s).

See ./Documents/ .

Reply

Avishek Kumar January 16, 2015 at 12:17 pm

Yeah pim!
Thanks for the concern

Reply

KM Sitlhou January 12, 2015 at 8:05 pm

Have gone through the article and I must say that it is an eye-opener indeed.But , to be a hacker and really being able to break the root password would be to retrieve the root password itself and not resetting it.For example , there is a remote linux server somewhere around the world.I know the ip address of the server and so I want to compromise the server.In such a scenario , a real root password hacking would be being able to break the root password remotely and then owning the system.

So , is that possible ?

Reply

Avishek Kumar January 16, 2015 at 12:18 pm

No! Simply not.

Reply

l3thal March 27, 2016 at 7:32 pm

Avishek: KM: Yes, it is possible. Not in the same ways described by Avishek of course, but it is possible to remotely exploit some vulnerability on the server and gain root access.

Reply

Lee Hobson January 5, 2015 at 1:14 pm

I'm not able to get a GRUB menu, when I press any key to interrupt the boot.

Reply

Avishek Kumar January 16, 2015 at 12:18 pm

why don't you check log files?

Reply

« Older Comments

GOT SOMETHING TO SAY? JOIN THE DISCUSSION.

Comment

Name *

Email *

Website

Post Comment

Notify me of followup comments via e-mail. You can also subscribe without commenting.

How to Add Linux Host to Nagios Monitoring Server Using NRPE Plugin

Nagios 4.0.1 Released – Install on RHEL/CentOS 6.x/5.x and Fedora 19/18/17

Google Chrome 52 Released – Install on RHEL/CentOS 7/6 and Fedora 23-15

Install Cacti (Network Monitoring) on RHEL/CentOS 7.x/6.x/5.x and Fedora 21-12

Wine 1.8 Released After 17 Months of Development – Install on RHEL/CentOS and Fedora

Install Latest Apache 2.4, MySQL 5.5/MariaDB 10.1 and PHP 5.5/5.6 on RHEL/CentOS 7/6 & Fedora 24-18

- Linux from Scratch – Create Your Own Linux OS
- Linux Shell Scripting Cookbook, Second Edition
- Securing & Optimizing Linux: The Hacking Solution
- User Mode Linux – Understanding and Administration

LINUX MONITORING TOOLS

How to Limit the Network Bandwidth Used by Applications in a Linux System with Trickle
25 FEB, 2015

Install Mtop (MySQL Database Server Monitoring) in RHEL/CentOS 6/5/4, Fedora 17-12
1 OCT, 2012

linux-dash: Monitors “Linux Server Performance” Remotely Using Web Browser
28 APR, 2014

Monitor Server Logs in Real-Time with “Log.io” Tool on RHEL/CentOS 7/6
28 OCT, 2014

LINUX INTERVIEW QUESTIONS

10 VsFTP (Very Secure File Transfer Protocol) Interview Questions and Answers
3 FEB, 2014

10 Basic Interview Questions and Answers on Linux Networking – Part 1
2 AUG, 2014

11 Basic Linux Interview Questions and Answers
18 NOV, 2013

Nishita Agarwal Shares Her Interview Experience on Linux 'iptables' Firewall
4 JUN, 2015

OPEN SOURCE TOOLS

Installing Seafile (Secure Cloud Storage) with MySQL Database in RHEL/CentOS/SL 7.x/6.x
27 JUN, 2014

How to Install JAVA 8 (JDK 8u45) on Linux Systems
18 JUN, 2015

Install GIT to Create and Share Your Own Projects on GITHub Repository
29 OCT, 2013

A Career in Linux is What You Should Be Pursuing In 2014
27 FEB, 2014