Chroot From Installation Media

Slackware is full of tools that can help when the system becomes unstable and cannot boot. One example is upgrading the kernel image and forgetting to run `lilo` afterwards.

In order to gain access to your system without booting directly to it, it is possible to use an installation media such as Slackware CD1 or the DVD. Once the installation media loads and starts, you can change the media's root directory into a mounted hard-disk partition and use it as the root directory, thus running commands directly from it and affecting it.

Volume Preparation

The examples here are very basic and expect that the entire system is located in a single partition. If you have a different partition table, please make sure that you mount all the needed partitions with care.

In the simplest of examples, a single hard disk was partitioned locally, normally and it is not encrypted in any way. In this case, all you need to do is to make sure what is the partition's 'name' is and continue to the mounting section.

If this is not the case, and you are using *LVM/EVMS* or an encrypted volume you will need to prepare the volumes before you can mount and chroot into them:

To unlock your LUKS partition, you will need to 'open' it and give it a name with this command (sdxn as example):

```
cryptsetup luksOpen /dev/sdxn crypted (any name will do)
```

At this point you will be prompted to insert the pass-phrase to unlock the volume. This partition will be mapped to '/dev/mapper/crypted'.

For LVM volumes you need to make sure that the system can recognize and activate the volume. This is done by running the commands:

- `vgscan --mknodes` *: searches for logical volumes, this might take a while*
- `vgchange -ay` *: activates the found volumes*
- If more than one group was found, you can select which one to activate with `vgchange -ay groupName`

Volume Mounting

After the preparation stage we can continue to mount the volume(s). You can use `/mnt` as it is only needed if you are planning to run the setup program.

After we make sure which partitions are needed, we need to mount them:

```
mount /dev/<location> /mnt
```

Here are 3 common examples.

1. The first is just a simple installation with everything installed under / mounted under /dev/sda1. No LVM or additional drives.

```
mount /dev/sda1 /mnt
```

2. In the next example we have two drives. The user has set up his or her Slackware system to use /dev/sdb1 for /home and /dev/sda1 for everything else.

```
mkdir /mnt/home # We need a directory to mount which needs to be under the
directory we intend to chroot into.
mount /dev/sda1 /mnt
mount /dev/sdb1 /mnt/home
```

3. For the third example the user has used LVM and has already made their volume group known to the kernel. The user is using the logical volumes “root, usr, home, opt, var, srv” all under one volume group labeled “myvg”. In addition, this user has used LUKS encryption and therefore has placed a small /boot under /dev/sda1.

```
mkdir /mnt/{boot,usr,home,opt,var,srv} # Create the necessary directories.
mount /dev/myvg/root /mnt
mount /dev/myvg/usr  /mnt/usr
mount /dev/myvg/home /mnt/home
mount /dev/myvg/opt  /mnt/opt
mount /dev/myvg/var  /mnt/var
mount /dev/myvg/srv  /mnt/srv
mount /dev/sda1      /mnt/boot
```

We could have also used a for loop for everything except /boot (/dev/sda1) and / (/dev/myvg/root) in this example:

```
for dir in usr home opt var srv
do
  mount /dev/myvg/$dir /mnt/$dir
done
```

Next, we need to prepare three virtual directories to be used by the environment. Those are `/dev`, a directory with virtual files that represent hardware devices, `/proc`, a directory with virtual files that represent processes and `/sys` which contains the kernel and other system files:

```
mount -o bind  /dev /mnt/dev
mount -o bind /proc /mnt/proc
mount -o bind /sys  /mnt/sys
```

Chrooting

Once the partition is mounted, we can chroot to it:

```
chroot /mnt /bin/bash
```

If LVM preparation was involved it might be necessary to re-run `vgscan --mknodes` and `vgchange -ay` as they were created inside the installation media's ramdisk and not in the mounted partition.

The bash prompt that you see here is a bash prompt started on the actual system you mounted. You can now work in this environment naturally. Operations will happen on your system, not from the installation media.

For example you can do these steps :

- run the command `/usr/share/mkinitrd/mkinitrd_command_generator.sh` to create the initrd.gz if you forgot to re-create it after a kernel upgrade
- edit the file `/etc/lilo.conf` to make changes to your lilo boot sequence, for instance selecting a different installed kernel
- run the command`/sbin/lilo -v` to apply the changes so that the modifications you made area applied upon the next boot

If the system uses elilo instead of lilo you can :

- repair elilo consecutive to a kernel upgrade, often we forget to re-generate the initrd.gz file
- mount the efi partition to `/mnt/boot/efi` in addition to the root partition described in the Volume Mounting section
- run the command `/usr/share/mkinitrd/mkinitrd_command_generator.sh -k 5.15.117` to create the initrd.gz if you forgot to re-create it after a kernel upgrade
  - use the kernel version applicable to your system
- run `eliloconfig` from the chroot location /

Sources

* Originally written by cmyster.
* LVM and LUKS information comes primarily from Eric's READMEs on your installation disc.

howtos, slackware administration, chroot, recovery, author cmyster