GNU/Linux Administration - Support #642

GNet Developer Arch Desktop Installation

08/21/2015 05:44 PM - Daniel Curtis

Status:	Closed	**Start date:**	07/08/2014
Priority:	Normal	**Due date:**	
Assignee:	Daniel Curtis	**% Done:**	100%
Category:	Workstation	**Estimated time:**	8.00 hours
Target version:	Arch Linux	**Spent time:**	4.50 hours

Description

This is a guide on how I have setup one of my desktops for developing defferent project. The primary hard drive consists of 2 partitions:

- /dev/sda1 - Windows 10 (Dummy OS, installation not covered in this guide)
- /dev/sda2 - Arch
- /dev/sdb1 - USB Boot partition

- I used cfdisk to partition the main hard drive:

```
cfdisk /dev/sda
```

The Windows partition installs its bootloader on the primary hard drive. The intention is to have the Arch boot off of a USB drive, but will only boot into Windows if the drive is not present.

The general software load out consists of:

- **VirtualBox w/ Guest Additions**: Virtual computing software
- **LibreOffice**: Office suite
- ~~**TrueCrypt**: High-grade encryption tool~~
- **Tomb**: Modern open source encryption management tool, replaces TrueCrypt
- **Windows Network Browsing**: For connecting to Windows shares
- **Firefox**: Open Source Web browser
- **Chromium**: Open Source Chrome Web browser
- **Chrome**: Google Proprietary Chrome Web browser
- **Thunderbird**: Mail client
- **ownCloud Client**: Personal cloud client
- **Pidgin**: Instant messaging client
- **BleachBit**: Browser, mail, application cleaning application
- **GIMP**: Image editing
- **Filezilla**: FTP/SFTP Client
- **git**: Source code management
- **KeePass2**: Password management
- **VLC**: Media player
- **Flash**: Closed source media and content plugin
- **Arduino**: Arduino Integrated Development Environment
- **Fritzing**: Prototyping software
- **PlayOnLinux**: Front-end to Wine
- **Steam**: Digital gaming distributor
- **Komodo Edit**: Open Source IDE
- **Wireshark** : Network traffic analysis tool

Preparing the two partitions

I decided to use LUKS on the root partition.

- Format the partitions, if any custom options are wanted, this is where you would specify them.:

```
cryptsetup -i 15000 -c aes-xts-plain:sha512 -y -s 512 luksFormat /dev/sda2
```

1. **NOTE**: This will prompt you for a passphrase to use for encrypting the partition. If I were truly paranoid I would use a keyfile with the -d flag and generate a 1K random keyfile:

   ```
   dd if=/dev/urandom of=/path/to/keyfile bs=1K
   ```

- Now map open the LUKS partition to tad them to the device mapper:

  ```
  cryptsetup luksOpen /dev/sda2 root
  ```

- Next, create the filesystem for the containers. I chose ext4, though the choice in filesystem is user-preferential; I would like to try ZFS at some point.

  ```
  mkfs.ext4 /dev/mapper/root
  ```

- Mount the new encrypted partitions:

  ```
  mount /dev/mapper/root /mnt
  mkdir /mnt/boot
  ```

Prepare the USB bootloader

This is one layer in my defense-in-depth, needing a USB with the bootloader installed onto. If I were a tad more paranoid, I would include the usage of a keyfile.

I usually add a 512MB ext4 partition to the end of a USB drive, this will be enough room for a few kernels. Using cfdisk will simplify the task:

```
cfdisk /dev/sdb
```

- Once the partition is created and formatted to the appropriate filesystem, mount the USB drive to the installation path /boot folder:

  ```
  mount /dev/sdb1 /mnt/boot
  ```

Install the base system

- Generate an fstab:

  ```
  mkdir /mnt/etc
  genfstab -p /mnt >> /mnt/etc/fstab
  ```

- Now its time to install the base system:

  ```
  pacstrap /mnt base base-devel grub openssh
  ```

  - chroot into the newly installed system:

    ```
    arch-chroot /mnt
    ```

  - Write your hostname to /etc/hostname:

```
echo 'archdev' >> /etc/hostname
```

- Symlink /etc/localtime to /usr/share/zoneinfo/Zone/SubZone:

```
ln -s /usr/share/zoneinfo/America/Los_angeles /etc/localtime
```

- Uncomment the selected locale in /etc/locale.gen and generate it with:

```
vi /etc/locale.gen
:%s/#en_US.UTF-8/en_US.UTF-8
:wq
locale-gen
```

- Configure /etc/mkinitcpio.conf as needed and create an initial RAM disk with:

```
mkinitcpio -p linux
```

1. **NOTE**: Make site to add the ***encrypt*** word to the mkinitcpio.conf HOOKS section:

HOOKS="... encrypt ... filesystems ..."

- Set a root password:

```
passwd
```

- Configure the network again for newly installed environment:

```
cp /etc/netctl/examples/ethernet-dhcp /etc/netctl/wired
netctl enable wired.service
```

- Enable SSH

```
systemctl enable sshd.service
```

Install the bootloader

- Before installing the booloader to the USB drive, the bootloader must be configured for the encrypted root partition. This can be done by making the following modification to /etc/default/grub:

```
GRUB_CMDLINE_LINUX_DEFAULT="root=/dev/mapper/root cryptdevice=/dev/sda2:root quiet"
```

- Now install GRUB onto the USB drive:

```
grub-install --target=i386-pc --recheck --debug /dev/sdb
grub-mkconfig -o /boot/grub/grub.cfg
```

Exit the install environment and reboot

- At this point the system will be bootable from the USB drive. Exit and reboot the out of the installation environment:

```
exit
umount /mnt/boot
umount /mnt
```

```
reboot
```

Add an administrator user

It is generally a good idea not to run command directly as root, but rather as an administrative user using the sudo wrapper command.

- First install sudo:

```
pacman -S sudo
```

- And create a user:

```
useradd -m -g users -s /bin/bash bob
```

- Set a password for bob:

```
passwd bob
```

- Then create the sudo group and add the user bob to the sudo group:

```
groupadd sudo
usermod -aG sudo bob
```

- Allow anyone part of the sudo group run sudo commands:

```
visudo
```

  - And add the following line to add bob to the sudo file:

```
%sudo ALL=(ALL) ALL
```

Install a desktop environment

There are many choices for desktop environments, I went through a few before I returned to my favorite (LXDE). Here are a few popular ones just for reference:

- GNOME

```
pacman -S gnome
systemctl enable gdm.service
systemctl start gdm.service
```

- KDE

```
pacman -S kde
systemctl enable kdm.service
systemctl start kdm.sercice
```

- XFCE

```
pacman -S xfce4 xorg xorg-xinit
echo 'exec startxfce4' >> ~/.xinitrc
startx
```

- LXDE

```
pacman -S lxde xorg xorg-xinit dbus gvfs gvfs-smb
echo 'exec startlxde' >> ~/.xinitrc
startx
```

Install Wifi

- Since my laptop has a weird broadcom wifi card in it, I needed to use b43-fwcutter to install wireless drivers:

```
curl -LO http://downloads.openwrt.org/sources/broadcom-wl-4.178.10.4.tar.bz2
tar xjf broadcom-wl-4.178.10.4.tar.bz2
cd broadcom-wl-4.178.10.4/linux
sudo b43-fwcutter -w /lib/firmware wl_apsta.o
```

- Then install the wpa_supplicant package:

```
sudo pacman -S wpa_supplicant
```

- And finally create the netctl configuration, start and enable the network profile:

```
cp /etc/netctl/examples/wireless-wpa /etc/netctl/wireless-net
netctl enable wireless-net
netctl start wireless-net
```

Install the packages

- For the packages I require through the Arch repositories, I will just run with one command:

```
sudo pacman -S chromium firefox filezilla keepass vlc base-devel wget bleachbit calibre cifs-u
tils epdfview flashplugin geany gimp git gparted gqrx leafpad libreoffice mpv nmap pidgin play
onlinux remmina rsync steam thunderbird virtualbox virtualbox-host-modules virtualbox-guest-is
o virtualbox-host-dkms linux-headers wpa_supplicant_gui wireshark-cli wireshark-gtk handbrake
openshot kdenlive dvdauthor
```

Install yaourt

- Install wget:

```
pacman -S wget
```

- Install yaourt

1. Google Chome

```
yaourt google-chrome
```

2. Tomb

```
yaourt tomb
```

3. caffeine-systray

```
yaourt caffeine-systray
```

4. owncloud-client

```
yaourt owncloud-client
```

5. fritzing

```
yaourt fritzing
```

6. arduino

```
arduino
```

7. komodo-edit

```
yaourt komodo-edit
```

Caveats & Notes

Prepare VirtualBox host

- The vboxdrv kernel module needs to be loaded:

```
sudo modprobe vboxdrv
```

- Users also need to be added to the vboxusers group in order to use VirtualBox:

```
gpasswd --add bob vboxusers
```

- Build DKMS modules

```
dkms install vboxhost/4.3.14
```

- Enable DKMS modules at boot

```
systemctl enable dkms.service
```

NOTE: Intel Galileo uses a different Arduino IDE

Since I am developing on the Intel Galileo, I needed to grab Intel's IDE from https://communities.intel.com/docs/DOC-22226

Building the above arduino package from the AUR may resolve issues if there are any problems installing the Intel version of the arduino IDE.

NOTE: Arduino and Galileo users must be added to the lock and uucp groups

The arduino board communicates with the computer via a serial connection or a serial over USB connection. So the user needs read/write access to the serial device file. Udev creates files in /dev/tts/ owned by group uucp so adding the user to the uucp group gives the required read/write access:

```
gpasswd -a $USER uucp
gpasswd -a $USER lock
```

Note: You will have to logout and login again for this to take effect.

The arduino board appears as /dev/ttyACMx so if the above doesn't work try adding the user to the group tty:

```
gpasswd -a $USER tty
```

Before uploading to the Arduino, be sure to set the correct serial port, board, and processor from the Tools menu.

Fix the corrupt text with Steam

Steam needs to have its own fonts, this can be installed by doing the following:

```
mkdir ~/SteamFonts && cd ~/SteamFonts
wget https://support.steampowered.com/downloads/1974-YFKL-4947/SteamFonts.zip
unzip SteamFonts.zip
sudo cp * /usr/share/fonts/TTF/
sudo chown -R root.root /usr/share/fonts/TTF/
```

tmpfs

- It is sometimes useful to offload the /tmp folder to RAM by using tmpfs:

```
echo 'tmpfs    /tmp          tmpfs    nodev,nosuid,size=2G          0  0' >> /etc/fstab
```

Resources

- http://zfsonlinux.org/
- http://en.wikipedia.org/wiki/ZFS
- http://www.zfsbuild.com/2010/05/26/zfs-raid-levels/
- http://nex7.blogspot.ch/2013/03/readme1st.html
- http://wintelguy.com/raidcalc.pl
- https://calomel.org/zfs_raid_speed_capacity.html
- https://pthree.org/2012/04/17/install-zfs-on-debian-gnulinux/
- http://docs.oracle.com/cd/E19253-01/819-5461/

Related issues:

Copied from GNU/Linux Administration - Support #410: GNet Developer User Arch...	**Closed**	**07/08/2014**

History

#1 - 08/21/2015 05:44 PM - Daniel Curtis

- Copied from Support #410: GNet Developer User Arch Installation added

#2 - 08/24/2015 12:14 PM - Daniel Curtis

- Status changed from New to In Progress

- % Done changed from 0 to 50

- Description updated

#3 - 08/26/2015 03:08 PM - Daniel Curtis

- Description updated

- % Done changed from 50 to 80

#4 - 09/11/2015 07:30 PM - Daniel Curtis

- Status changed from In Progress to Resolved

- % Done changed from 80 to 100

#5 - 10/18/2015 01:43 PM - Daniel Curtis

- Description updated

#6 - 11/27/2015 04:47 PM - Daniel Curtis

- Status changed from Resolved to Closed

#7 - 04/19/2016 08:00 PM - Daniel Curtis

- Description updated

#8 - 07/15/2016 07:41 PM - Daniel Curtis

- Description updated