Dell XPS 15 9550 Install Guide

Introduction:

The goal here is to maintain both a Windows and Linux partition. However, we want to utilize btrfs for linux and employ the power of subvolumes to allow for multiple distros, separate home dir, or other logical volume creation needs!

Windows Time:

Disable Secure Boot, Enable Legacy Mode

Utilize Windows Pro install iso
Install to usb using rufus
Clear out hdd of all prev partitions and create one for windows
It will create a System Reserved: Delete this, click the new one it had made, and click Next. (leave the 500MB it left at the front of the disk; use this for /boot later)

Go through the basic Windows 10 setup steps afterward. Don't worry if the install gets to "Just a Moment" and sits there. I rebooted once and it continued and then moved on...others have said to type "Ctrl+Alt+Delete 3 times, then ESC"

Install RAM

We bought G.Skill DDR4-2133 RAM (must maintain that speed!)

- Be sure to boot before putting the bottom on (verify it works)!
- If not seated properly, laptop won't boot (no screen) and front middle bottom light will blink orange!
- Ensure laptop off fully before going into it (especially post a failure)!

cfdisk

cfdisk /dev/nvmee0n1

Make Boot, set to Bootable (initial 2MB offset is automatically done for you! Look for "start" to be "2048")
Make Rest of disk for root
Remove bootable flag from Windows/NTFS partition and apply to /boot!

dm-crypt (cryptsetup options)

cryptsetup -y -v luksFormat -s 512 /dev/nvme0n1p3

(type YES)
cryptsetup luksOpen <p3> cryptroot

mkfs.btrfs -L root /dev/mapper/cryptroot
mkfs.ext2 <p2>

btrfs (mount options, subvolumes info)
mkdir root (so we now have /root/root)
mount -t btrfs /dev/mapper/cryptroot ./root
cd ./root (we're now in the mount location `/root/root`)
btrfs subvolume create ./rootfs

This isn't exactly necessary. You could just install at the default top level, instead of making a separate subvol for your rootfs. However, this means your home is still tied to your operating system install, which could cause problems later on with reinstalls, etc.. I choose a completely isolated and separate way of organizing things. Completely up to you ;)

> btrfs subvolume list -p .
btrfs subvolume create ./home
btrfs subvolume set-default 257 ./rootfs (# set the rootfs subvol as the default when mounting/booting the partition)

> Here we unmount the root level of the filesystem then remount with our new separate rootfs subvolume and preferred mount options for further installation of Arch. The options used here will be transferred to our /etc/fstab when we genfstab later on!

umount ./root
mount -t btrfs -o subvol=/rootfs,ssd,discard,space_cache /dev/mapper/cryptroot ./root
cd ./root
mkdir ./home
mount -t btrfs -o subvol=/home,ssd,discard,space_cache /dev/mapper/cryptroot ./home
mkdir ./boot
mount <p2> ./boot

...**Now follow Arch install guide from "Installation" heading**
For genfstab, ensure you provide the absolute path to your mounted rootfs i.e. `/root/root` in this example and also utilize `-U` to use uuid's!

genfstab -U /root/root >> ./root/etc/fstab

You'll also want to take a look at the generated fstab and ensure btrfs options are correct. For instance, mine tripled up on subvol entries (not actually a problem, but a little overkill):

subvolid=257,subvol=/rootfs,subvol=rootfs

Additionally, check your mount options (it uses the ones you used plus a few others, a little more explicit):

rw,relatime,ssd,discard,space_cache,{subvol_declarations(see above)}

initramfs:

For /etc/mkinitcpio.conf:
MODULES="nvme"
HOOKS="base udev autodetect modconf block keyboard keymap encrypt filesystems fsck"

Before building initramfs,

pacman -Sy btrfs-progs

now we can build initramfs:

mkinitcpio -p linux

Grub

Install the needed packages to setup our bootloader:

pacman -Sy grub os-prober

Edit /etc/default/grub with something like the following:

GRUB_CMDLINE_LINUX="cryptdevice=UUID=<uuid>:cryptroot"

Where <*uuid*> is the /dev/mapper/cryptroot uuid gotten from *blkid*, *lsblk*, or */dev/disk/by-uuid*

Now, let's install grub to the boot sector and make our grub.cfg:

grub-install --target=i386-pc --modules=part_msdos /dev/nvme0n1
grub-mkconfig -o /boot/grub/grub.cfg

Final Steps

Exit chroot, umount all fs'es, reboot

Drivers:

Touchpad: https://wiki.archlinux.org/index.php/Touchpad_Synaptics#Installation
Graphics:
https://wiki.archlinux.org/index.php/Bumblebee#Installing_Bumblebee_with_Intel.2FNVIDIA

Ensure you install the handlers:

pacman -Sy primus bbswitch

Then, *systemctl enable bumblebeed*

You can figure out if the Nvidia card is on by doing a *cat /proc/acpi/bbswitch*
(By default, you boot and operate normally without the card on. You can then use *primusrun* to launch applications which you want to utilize the card.)

See Also:

https://wiki.archlinux.org/index.php/Dell_XPS_15

Graphical Decryption Stage:

If you'd like the decryption prompt to be graphically pleasing, you can utilize plymouth to provide a nicer entry box and splash whilst booting to the operating system post key entry. Here's how!

Here we actually replace gdm with gdm-plymouth and enable the service.
We also need to install plymouth itself, which we get from antergos so we don't have to build it (we're just being lazy, you'll be fine grabbing it from the default arch pkg source)

> *yaourt -Sya antergos/plymouth gdm-plymouth*
> *sudo systemctl enable gdm-plymouth.service*

We then edit and rebuild our initramfs parameters to call the right encryption hook:

Open */etc/mkinitcpio.conf* and do the following:

> Instead of "encrypt", we use "plymouth-encrypt"
> Also, after "udev", we insert "plymouth"

Then, rebuild with "*mkinitcpio -p linux*"