FreeBSD Administration - Support #799

Dual Boot Windows 10 and PCBSD With GELI Encrypted ZFS Root

04/16/2016 12:07 AM - Daniel Curtis

Status:	Closed	**Start date:**	04/17/2016
Priority:	Normal	**Due date:**	
Assignee:	Daniel Curtis	**% Done:**	100%
Category:	Workstation	**Estimated time:**	3.00 hours
Target version:	PCBSD	**Spent time:**	5.50 hours

Description

This is a guide on how I set up my laptop to dual boot Windows 10 and PCBSD with a GELI encrpyted ZFS root on a Dell Inspiron 15-3521 UEFI based system.

The setup uses Windows 10 as the primary OS, but the PCBSD partition will booted from a USB flash drive. This guide assumes that the Windows 10 partition has been installed and adequately shrunk.

- When the PCBSD Installation message appears, choose **Text Install** / **Emergency Console**.
- Select **Utility** then **Shell**.
- Get a list of available drives:

```
camcontrol devlist
```

  - *Example output*:

```
<VB0250EAVER HPG9>                 at scbus0 target 0 lun 0 (pass0,ada0)
<Sony USB Stick>                   at scbus6 target 0 lun 0 (pass4,da0)
```

Swap

- Create the **swap** slice:

```
gpart add -s 4G -t freebsd-swap -a 4k -l swap0 ada0
```

  - *Example output*:

```
ada0p8 added
```

- Encrypt the swap space:

```
geli onetime -d -e AES-XTS -l 256 -s 4096 /dev/gpt/swap0
```

USB Bootloader

- Create the boot partition and install the bootcode on the USB drive:

```
gpart create -s gpt da0
gpart add -l gptboot0 -s 512k -t freebsd-boot -a 4k da0
gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 da0
gpart set -a bootme -i 1 da0
```

- Create the ZFS **bootpool** on the USB drive and mount it:

```
gpart add -l boot0 -t freebsd-zfs da0
mkdir -p /tmp/mnt/bootpool
zpool create -m none -o altroot=/tmp/mnt/bootpool bootpool /dev/gpt/boot0
mkdir -p /tmp/mnt/bootpool/boot/zfs
mount_nullfs /tmp/mnt/bootpool/boot/zfs /boot/zfs
```

GELI ZFS Root

- Create the disk0 slice:

```
gpart add -t freebsd-zfs -a 4k -l disk0 ada0
```

  - *Example output*:

```
ada0p9 added
```

- Encrypt the OS slice:

```
mkdir /tmp/mnt/bootpool/boot/metadata_backup
geli init -b -s 4096 -e AES-XTS -l 256 -B /tmp/mnt/bootpool/boot/metadata_backup/ada0p9.eli /d
ev/ada0p9
```

  - **NOTE**: This will store a copy of the GELI metadata on the USB drive, in case bad things happen.

- Attach the encrypted slice:

```
geli attach /dev/ada0p9
```

- Create the **xpool** ZFS pool on top of the GELI encrypted slice, then export it:

```
mkdir -p /tmp/mnt/xpool
zpool create -o altroot=/tmp/mnt/xpool -o cachefile=/tmp/zpool.cache -m none -f xpool /dev/ada
0p9.eli
zpool export xpool
```

- Next import the **xpool** ZFS pool and create the root dataset and settings:

```
zpool import -o altroot=/tmp/mnt/xpool -o cachefile=/tmp/zpool.cache xpool
zpool set bootfs=xpool xpool
zfs set checksum=fletcher4 xpool
zfs set atime=off xpool
zfs create xpool/ROOT
zfs set mountpoint=/ xpool/ROOT
```

  - Then create some additional system datasets:

```
zfs create -o canmount=off xpool/ROOT/usr
zfs create -o canmount=off xpool/ROOT/var
zfs create -o compression=on   -o exec=on  -o setuid=off xpool/ROOT/tmp
zfs create -o compression=gzip -o setuid=off  xpool/ROOT/usr/ports
zfs create -o compression=off  -o exec=off -o setuid=off xpool/ROOT/usr/ports/distfiles
```

```
zfs create -o compression=off  -o exec=off -o setuid=off xpool/ROOT/usr/ports/packages
zfs create -o compression=gzip -o exec=off -o setuid=off  xpool/ROOT/usr/src
zfs create -o compression=lzjb xpool/ROOT/usr/obj
zfs create -o compression=lzjb -o exec=off -o setuid=off xpool/ROOT/var/crash
zfs create -o compression=off  -o exec=off -o setuid=off xpool/ROOT/var/empty
zfs create -o compression=lzjb -o exec=on  -o setuid=off xpool/ROOT/var/tmp
```

- Set the permissions of the temp directories in the zfs mount:

```
chmod 1777 /tmp/mnt/xpool/tmp
chmod 1777 /tmp/mnt/xpool/var/tmp
```

- Remount the **bootpool**:

```
umount /boot/zfs
mkdir /tmp/mnt/xpool/bootpool
zfs set mountpoint=/tmp/mnt/xpool/bootpool bootpool
zpool export bootpool
zpool import bootpool
mkdir -p /tmp/mnt/xpool/bootpool/boot/zfs
mount_nullfs /tmp/mnt/xpool/bootpool/boot/zfs /boot/zfs
```

- Extract the base.txz and kernel.txz to the zfs root to install the base system:

```
cat /dist/base.txz | tar --unlink -xpJf - -C /tmp/mnt/xpool
cat /dist/kernel.txz | tar --unlink -xpJf - -C /tmp/mnt/xpool
```

Post-Installation Setup

- Chroot into the xpool:

```
chroot /tmp/mnt/xpool
```

- Copy the install bootload files over to the bootpool, then create a /boot symlink:

```
cd /
rm -r boot/zfs
mv boot/* bootpool/boot/
rm -r boot
ln -sf bootpool/boot
```

- Create an fstab file:

```
vi /etc/fstab
```

  - And add the swap partition definition:

```
/dev/ada0p8.eli         none    swap    sw    0    0
```

- Add the initial system configuration:

```
echo 'zfs_enable="YES"' >> /etc/rc.conf
echo 'sshd_enable="YES"' >> /etc/rc.conf
echo 'hostname="pcbsd.example.com"' >> /etc/rc.conf
```

- Add the bootloader config:

```
echo 'geom_eli_load="YES"' >> /boot/loader.conf
echo 'zfs_load="YES"' >> /boot/loader.conf
echo 'vfs.root.mountfrom="zfs:xpool/ROOT"' >> /boot/loader.conf
echo 'zpool_cache_load="YES"' >> /boot/loader.conf
echo 'zpool_cache_type="/boot/zfs/zpool.cache"' >> /boot/loader.conf
echo 'zpool_cache_name="/boot/zfs/zpool.cache"' >> /boot/loader.conf
```

Networking

- Show what network interfaces are available:

```
ifconfig
```

  - **NOTE**: This guide uses em0 for the ethernet interface and ath0 as the wireless interface.

Ethernet

- Add the em interface driver to the bootloader config:

```
echo 'if_em_load="YES"' >> /boot/loader.conf
```

- Setup ethernet networking using DHCP:

```
echo 'ifconfig_em0="DHCP"' >> /etc/rc.conf
echo 'hostname="freebsd.example.com"' >> /etc/rc.conf
```

- (Optional) Setup networking using a static IP address instead:

```
echo 'ifconfig_em0="inet 192.168.10.70 netmask 255.255.255.0 broadcast 198.100.10.255"' >> /et
c/rc.conf
echo 'defaultrouter="192.168.10.1"' >> /etc/rc.conf
echo 'hostname="freebsd.example.com"' >> /etc/rc.conf
echo 'nameserver 192.168.10.1' >> /etc/resolv.conf
```

Wireless

- Add the ath interface driver and the wireless cryptographic modules to the bootloader config:

```
echo 'if_ath_load="YES"' >> /boot/loader.conf
echo 'wlan_ccmp_load="YES"' >> /boot/loader.conf
echo 'wlan_tkip_load="YES"' >> /boot/loader.conf
```

- Setup wireless networking using WPA and DHCP:

```
echo 'wlans_ath0="wlan0"' >> /etc/rc.conf
echo 'ifconfig_wlan0="WPA SYNCDHCP"' >> /etc/rc.conf
```

- Create a wpa_supplicant.conf file:

```
vi /etc/wpa_supplicant.conf
```

  - And add the following, modifying accordingly:

```
network={
    ssid="HomeWifi"
    psk="SuperSecretPassword"
}
```

- Then restart the network interface service:

```
service netif restart
```

Finish the Installation

- Exit from the chroot environment:

```
exit
```

- Setup the ZFS mountpoints

```
zfs set mountpoint=legacy xpool/ROOT
zfs set mountpoint=/tmp xpool/tmp
zfs set mountpoint=/usr xpool/usr
zfs set mountpoint=/var xpool/var
zfs set mountpoint=/bootpool bootpool
```

- Unmount the filesystems:

```
umount /boot/zfs
zfs unmount -a
zpool export xpool
zpool export bootpool
```

- Reboot the system and eject the FreeBSD install disc:

```
reboot
```

Setup PCBSD

- Then, disable the FreeBSD package repository:

```
mv /etc/pkg/FreeBSD.conf /root/FreeBSD.conf-old
```

- Create the pkg repos directory:

```
mkdir -p /usr/local/etc/pkg/repos
```

- Then, create the PCBSD repo file:

```
vi /usr/local/etc/pkg/repos/pcbsd.conf
```

  - And add the following:

```
pcbsd: {
       url: "http://pkg.cdn.pcbsd.org/10.0-RELEASE/amd64",
       signature_type: "fingerprints",
       fingerprints: "/usr/local/etc/pkg/fingerprints/pcbsd",
       enabled: true
      }
```

- Next, create the pkg fingerprints directories:

```
mkdir -p /usr/local/etc/pkg/fingerprints/pcbsd/{revoked,trusted}
```

- Then, download the PCBSD repository fingerprint file:

```
cd /usr/local/etc/pkg/fingerprints/pcbsd/trusted/
fetch https://raw.githubusercontent.com/pcbsd/pcbsd/master/src-sh/pcbsd-utils/pc-extractoverla
y/ports-overlay/usr/local/etc/pkg/fingerprints/pcbsd/trusted/pkg.cdn.pcbsd.org.20131209
```

- Update the package database and any installed packages:

```
pkg update
pkg upgrade -fy
```

- Once the repository configuration is complete install the base components:

```
fetch --no-verify-peer -o /etc/freebsd-update.conf 'https://github.com/pcbsd/freebsd/raw/maste
r/etc/freebsd-update.conf'
freebsd-update fetch
freebsd-update install
```

- Then setup the installation to be a PC-BSD desktop

```
pkg install -fy pcbsd-base
rehash
pbreg set /PC-BSD/SysType PCBSD
pc-extractoverlay ports
pc-extractoverlay desktop
```

Setup Desktop Environment

- Install the xfce desktop environment:

```
pkg install pcbsd-meta-xfce
```

- Set the first boot scripts to run:

```
sh /usr/local/share/pcbsd/scripts/sys-init.sh desktop en_US
touch /var/.runxsetup
touch /var/.pcbsd-firstboot
touch /var/.pcbsd-firstgui
```

NOTE: If you are using NVIDIA video hardware, load the driver before rebooting into the display wizard:

```
pkg install pcbsd-meta-nvidia
```

Resources

- http://www.schmidp.com/2014/01/07/zfs-full-disk-encryption-with-freebsd-10-part-2/
- http://web.pcbsd.org/doc-archive/10.2/html/preinstall.html
- http://web.pcbsd.org/doc-archive/10.2/html/advanced.html
- https://srobb.net/fbsdquickwireless.html
- https://www.freebsd.org/doc/handbook/network-wireless.html
- https://forums.pcbsd.org/thread-20411.html

History

#1 - 04/16/2016 12:08 AM - Daniel Curtis

- Category set to Workstation

- Assignee set to Daniel Curtis

- Target version set to PCBSD

- Start date changed from 04/15/2016 to 04/17/2016

- % Done changed from 0 to 20

- Estimated time set to 3.00 h

#2 - 04/16/2016 01:13 PM - Daniel Curtis

- Description updated

- Status changed from New to In Progress

- % Done changed from 20 to 50

#3 - 04/16/2016 08:02 PM - Daniel Curtis

- Description updated

#4 - 04/16/2016 11:24 PM - Daniel Curtis

- Description updated

#5 - 04/16/2016 11:51 PM - Daniel Curtis

- Status changed from In Progress to Resolved

- % Done changed from 50 to 100

#6 - 04/22/2016 05:06 PM - Daniel Curtis

- Status changed from Resolved to Closed