

Erledigt

ZFS on OSX, nutzt das jemand außer mir)?

Beitrag von „Superjeff“ vom 20. November 2017, 18:48

Also dann los, ich beschreibe hier mal kurz und knackig wie ich ein verschlüsseltes Raid (Mirror) mit Cache eingerichtet hab.

Also ein verschlüsseltes, komprimierendes, gespiegeltes "Fusion-Drive".

Fetter Text bedeutet Eingabe im Terminal.

WARNUNG: Backup ist wichtig! Wir formatieren hier Festplatten, das heißt, alle Daten darauf sind dann weg!

Als erstes installieren wir uns ZFS von hier:

<https://openzfsonosx.org>

Dann brauchen wir noch einen Key, mit dem der Pool entschlüsselt wird.

Option ist die Eingabe eines Passworts im Terminal bei jedem Start oder ein Keyfile.

Aus Bequemlichkeit habe ich ein Keyfile genommen, daß auf die per FileVault verschlüsselte OSX-SSD gelegt wird.

Das heißt, man loggt sich beim Systemstart normal ein und erst dann ist das Keyfile lesbar.

Ich habe eins mit Zufallsdaten erstellt:

```
sudo head -c 32 /dev/urandom > /sbin/enc4zfs
```

Dann schauen wir uns an wie die Festplatten heißen

diskutil list

Hier im Beispiel sind das:

disk0 -> Boot-SSD mit High Sierra

disk1 -> 1. 4TB-HD

disk2 -> 2. 4TB-HD

disk3 -> 3. 4TB-HD

disk4 -> 4. 4TB-HD

disk5 -> 120 GB-SSD

Nun legen mir den Pool an. Dabei sind vorhandene Partitionen auf den Platten egal, wir nutzen die kompletten Laufwerke.

```
sudo zpool create -f -o ashift=9 -O casesensitivity=insensitive -O  
normalization=formD -O encryption=on -O keylocation=file:///sbin/enc4zfs -O  
keyformat=raw Daten disk1 disk2 mirror disk3 disk4 cache disk5
```

Damit haben wir den Pool angelegt (der Name ist dann "Daten") incl. der Cache-SSD
Nun konfigurieren wir noch Kompression, Checksummen etc.

```
sudo zfs set checksum=fletcher4 atime=off compression=lz4 Daten
```

Dann die Berechtigungen setzen:

```
sudo chown -R username:admin /Volumes/Daten
```

Bei 'username' dann den eigenen Login-Name einsetzen.

Damit der Pool beim Start automatisch gemountet wird muss ein wenig getrickst werden. Ist nicht sehr elegant, aber funktioniert.

Wir legen ein Script /sbin/mount-ZFS.sh an:

```
sudo nano /sbin/mount-ZFS.sh
```

Inhalt wie folgt:

```
#!/bin/bash  
sleep 2  
/usr/local/bin/zfs load-key Daten  
sleep 2  
/usr/local/bin/zfs mount Daten
```

Mit Strg-X speichern wir das Script und machen es startfähig:

```
sudo chmod +x /sbin/mount-ZFS.sh
```

Dieses Skript lädt den Key und bindet den Pool ein.

Aufgerufen wird es über ein Applescript welches in die Anmeldeobjekte gepackt wird.

Hier das Script:

delay 35

do shell script "sudo /sbin/mount-ZFS.sh"

Den Delay hab ich eingesetzt, damit OSX nach öffnen des Desktops genug Zeit hat die nötigen Treiber zu laden.

Da wir das Script aber mit 'sudo' aufrufen müssten wir nun bei jedem Start das Admin-Passwort eingeben.

Das können wir ändern wenn wir unserem User für das Mount-Script dauerhaft root-Rechte geben:

sudo visudo

Damit öffnen wir die sudoers-Datei und fügen folgende Zeile ein:

username ALL=NOPASSWD: /sbin/mount-ZFS.sh

wobei 'username' wieder unser Login-Name ist.

Wichtig ist, daß unser Script in /sbin liegt wo nur root Zugriff hat!

Liegt sie woanders könnte irgendwer der Zugriff auf den Mac hat die Datei ändern und somit beim Systemstart irgendwas mit Root-Rechten starten!

Das wars eigentlich schon...

Klingt ziemlich kompliziert, ist es aber nicht wenn man mal verstanden hat was da passiert. Auf Openzfs.org gibt es auch ein Wiki welches das auch nochmal beschreibt.