!!! ACHTUNG - evtl. veraltet - ACHTUNG !!!

Diese Seite wurde zuletzt am 9. Juli 2014 um 10:48 Uhr geändert.

unallocated area / HPA und loop-aes

Voraussetzungen:

- 1GB großer USB-Stick
- nach dem Einstecken ist der USB-Stick (in meinem beispiel) durch „/dev/sdc“ verfügbar
- primäre NTFS-Partition (ID = 7) ab „0“ mit einer Größe von „+900M“ (auch per „mkfs.ntfs“ als solche formatiert)
- per „losetup“ aus den „loop-aes-utils“ wurde ein entsprechender bereich (im meinem beispiel ab 921600000 für eine länge von 51200000) nach „/dev/loop0“ gemoutet
- und schließlich mit „mkfs.xfs“ formatiert

Wie bekomme ich mehr Informationen über die „unallocated area“?

```
 user@debian:~$ sudo cfdisk -P s /dev/sdc
 Partition Table for /dev/sdc
                First       Last
  # Type       Sector      Sector   Offset    Length   Filesystem Type (ID)
Flag
 -- ------- ----------- ----------- ------ ----------- --------------------
----
  1 Primary           0     1760551     62     1760552 HPFS/NTFS (07)
None
    Pri/Log     1760552     1956595      0      196044 Free Space
None
```

Im Falle einer HPA würde man sich „disk_stat“ aus Sleuthkit bedienen.

Zum einfachen Mounten habe ich mir ein kleines Shell-Skript (mymount in /usr/local/bin) geschrieben:

```
 #!/bin/bash
 if [[|! -f /etc/debian_version ]] ; then echo -e "\nsorry, this script was
written for debian gnu/linux-systems!\n" ; exit ; fi
 if [[|$# -ne 2 ]] ; then echo -e "\nusage: $0 <device> <mountpoint>\n" ;
exit ; fi
 if [[|! $( echo $1 | grep "^/dev/" ) ]] ; then echo -e "\n<device> is not
in /dev!\n" ; exit ; fi
 if [[|! -w $( dirname $2 ) ]] ; then echo -e "\nyou need write permisson to
the parent directory of the <mountpoint>!\n" ; exit ; fi
 DEVICE=$1
 MOUNTPOINT=$2
 OFFSET=$(( 1800000 * 512 ))
 SIZE=$(( 100000 * 512 ))
 dpkg -l | grep loop-aes-utils > /dev/null
 if [[|$? -ne 0 ]] ; then sudo aptitude install loop-aes-utils -y; fi
 LOOPDEV=$( sudo losetup -f )
```

```
 sudo losetup -o $OFFSET -s $SIZE $LOOPDEV $DEVICE
 mkdir $MOUNTPOINT
 sudo mount $LOOPDEV $MOUNTPOINT
```

Wendet man das ganze auf einen anderen USB-Stick an, so sind OFFSET und SIZE entsprechend anzupassen (ja, in meinem Fall wären noch präzisere Werte möglich gewesen, aber die kann ich mir immer so schwer merken )!

Zum einfachen Mounten habe ich mir ein kleines Shell-Skript (myumount in /usr/local/bin) geschrieben:

```
 #!/bin/bash
 if [[|! -f /etc/debian_version ]] ; then echo -e "\nsorry, this script was
written for debian gnu/linux-systems!\n" ; exit ; fi
 if [[|$# -ne 1 ]] ; then echo -e "\nusage: $0 <mountpoint>\n" ; exit ; fi
 if [[|! -d $1 ]] ; then echo -e "\nthe given mountpoint is not a
directory!\n" ; exit ; fi
 MOUNTPOINT=$1
 if [[|! "$( grep $MOUNTPOINT <( mount ) )" ]] ; then echo -e "\nnothing is
mounted to given mountpoint!\n" ; exit ; fi
 if [[|! "$( grep $MOUNTPOINT <( mount ) | grep /dev/loop )" ]] ; then
   echo -e "\nthere is no loopback-device like /dev/loop0 mounted to the
mountmount!\n" ; exit
 fi
 LOOPDEV=$( grep $MOUNTPOINT <( mount ) | awk '{ print $1; }' )
 echo -e -n "\ndelete mountpoint after unmounting? (y for yes... everything
else for no): "
 read DELETE
 if [[|"$DELETE" = "y" -a ! -w $( dirname $MOUNTPOINT ) ]] ; then
   echo -e "\nyou need write permisson to the parent directory of the
<mountpoint>!\n" ; exit
 fi
 sudo umount $LOOPDEV
 if [[|"$DELETE" = "y" ]] ; then rmdir $MOUNTPOINT ; fi
 sudo losetup -d $LOOPDEV
 echo
```

... nun sollten (bei eingabe eines „y“) alle „Spuren“ wieder etwas verwischt worden sein!?

Installation (von Verschlüsselung)

Debian-Pakete installieren:

```
 linux:~# aptitude install loop-aes-modules-2.6.18-6-686 loop-aes-utils
```

Das Standard-loop-Kernel-Modul entladen und anschließend das loop-aes-Kernel-Modul laden:

```
 linux:~# modprobe -v -r loop
```

```
linux:~# modprobe loop-aes
```

Hinweis:

Das loop-aes-Kernel-Modul kann u. U. Probleme mit gängigen und durchaus öfter benutzten Linuxbefehlen verursachen! Das ist u. a. der Grund, weshalb ich bei mir nicht zum Einsatz kommt.

Unverschlüsseltes virtuelles Gerät über loop-aes initiieren:

```
linux:~# losetup -e AES256 /dev/loop0 /dev/sda1
```

Ein Dateisystem nach Wahl im entschlüsselten, virtuellen Gerät erstellen:

```
linux:~# mkfs.xfs /dev/loop0
```

entschlüsseltes Gerät einbinden:

```
linux:~# mkdir /mnt/la_decrypted
linux:~# mount /dev/loop0 /mnt/la_decrypted
```

... ... Daten im Verzeichnis /mnt/la_decrypted erstellen...

entschlüsseltes Gerät trennen:

```
linux:~# umount /mnt/la_decrypted
linux:~# losetup -d /dev/loop0
```

Identifizierung

...