Using volatility and Linux to open encrypted drives

First: Find the profile of the memdump:

```
volatility -f ram.dd imageinfo
```

Create a folder for mounting the unencrypted folders:

```
sudo mkdir /mnt/decrypted
```

TrueCrypt:

To get information about Truecrypt:

```
volatility -f ram.dd --profile=Win7SP1x64 truecryptsummary
```

To extract/dump the master key to a file:

```
volatility -f ram.dd --profile=Win7SP1x64 truecryptmaster -D .
```

In Linux, you can now mount the encrypted volume in an empty directory using the master key file with the MKDecrypt python script:

```
./MKDecrypt.py -X -m /mnt/decrypted encrypted.tc ./masterkey
```

BitLocker:

To extract bitlocker keys, you need a plugin from Marcin Ulikowski. This can be used against full encrypted volumes, but NOT full disk encryption at the time of writing (https://github.com/elceef/bitlocker)

```
v.exe --plugins=. -f mem_dump.mem --profile=Win7SP1x64 bitlocker
```

This command will give you the FVEK and TWEAK keys.

In Linux, you can check the layout of the encrypted volume using mmls from Sleuthkit:

```
mmls volume.bl
```

Remember the offset to the partition with bitlocker – in mmls this offset is in sectors, but the other tools will need this in bytes. Win this example we use 128 as the offset from now on.

Check information about bitlocker using bdeinfo:

```
bdeinfo -o $((128*512)) volume.bl
```

Unlock the volume:

```
sudo bdemount -k FVEK:TWEAK -o $((128*512)) volume.bl /mnt/decrypted
```

In /mnt/decrypted you should now have a unmounted, unencrypted volume (ble1) that can be imaged. If you want to mount the image, you can do so as well:

```
mkdir mounted
mount -o,ro /mnt/decrypted/ble1 mounted
```