HowTo Firmware Netcenter editieren

(Der gesamte Vorgang lies sich nur unter Linux erfolgreich durchführen,
unter Cygwin war z.B. das Mounten des cramfs nicht möglich)

1. Bereiche für Kernel und Filesstemin der Original-FW finden

1.1 dazu die Originalfirmware mit einem Hexeditor öffnen und nach der Textstelle: „piggy" suchen,
-> der zu extrahierende Bereich beginnt einige Bytes vor dieser Textstelle mit der Bytefolge „*1F 8B 08 08* .."

- das Extrahieren darf also erst ab Byte *1F* erfolgen, alles vorher soll übersprungen werden,
daher nun zählen, wieviel Bytes davor liegen (ab Anfang der Datei), diesen Wert aufschreiben
(wird später für die Option skip= benötigt – siehe 2.1)

1.2 weitersuchen nach der Textstelle „Decompressed",
-> der zu extrahierende Bereich beginnt einige Bytes vor dieser Textstelle mit der Bytefolge „*45 3D CD 28* .."

- das Extrahieren darf also erst ab Byte *45* erfolgen, alles vorher soll übersprungen werden,
daher nun zählen, wieviel Bytes davor liegen (ab Anfang der Datei), diesen Wert aufschreiben
(wird später für die Option skip= benötigt – siehe 2.2)

2. Dateien extrahieren

- die Firmware in das Verzeichnis \opt kopieren

2.1 Kernel extrahieren (piggy.gz)

```
# cd /opt

# dd if=Firmware.wdf of=piggy.gz bs=1 skip=112
```
(bei skip= den unter 1.1 ermittelten Wert eintragen, 112 bezieht sich auf die Netcenter-Original-FW)
-> piggy.gz wird nun erstellt und sollte ca. 6,5 MB groß sein

- um die korrekte Größe zu erhalten, muß der Kernel noch nachbehandelt werden

```
# gunzip piggy.gz
```
-> Rückmeldung: gunzip: piggy.gz: decompression OK, trailing garbage ignored

(erfolgt eine Rückmeldung, wie „not gunzip-format" oder so, war der skip-Wert nicht korrekt)

```
# gzip -9 piggy
```
-> die Datei piggy.gz, sollt nun 1158029 Bytes groß sein

2.2 Filesystem extrahieren (root.cramfs)

```
# dd if=Firmware.wdf of=root.cramfs bs=1 skip=1151648
```
(bei skip= den unter 1.2 ermittelten Wert eintragen, 1158144 bezieht sich auf die Netcenter-Original-FW)
-> root.cramfs wird nun erstellt und sollte ca. 5,7 MB groß

2.2.1 Filesystem mounten (root.cramfs)

```
# mount -o loop,ro ./root.cramfs /mnt
```

- dies mounted das Verzeichnis /root der Firmware nach /mnt, es ist aber dort nicht editierbar
daher nun das gemountete Verzeichnis an eine andere Stelle kopieren, um es bearbeiten zu können

cp -Rp /mnt root

- nun befindet sich das Verzeichnis /root auch in /opt und ist editiertbar

- beim Mounten werden mehrere Fehlermeldungen über nicht kopierbare Verzeichnisse erfolgen,
diese müssen manuell erstellt werden (das folgende Beispiel bezieht sich auf die Netcenter-FW)

cd root
mkdir foreign_shares mnt proc shares tmp
cd usr/share
mkdir empty

- letzendlich kann das unter /mnt gemountete Verzeichnis /root unmounted werden
(wenn es dabei zu einer busy-Meldung kommt, den Vorgang später wiederholen)

umount /mnt

Nun kann das Filesystem der Firmware editiert werden

4. Firmware zusammensetzen

Dazu können z.B. die Tools WRT54GS_v4.70.6 verwendet werden (downloadbar bei Netgear -> Development). Nach dem Download der ca. 170 MB großen Datei werden die Tools entpackt, z.B. in /opt, so daß es dann ein Verzeichnis /opt/WRTGS54…. gibt. Eine spezielle Zusatzbehandlung benötigen die WRTGS-Tools für diese Zwecke nicht.

In der nachfolgenden Beschreibung wird von einem Verzeichnis /opt/WRT…/WRTGS… ausgegangen und die benötigten Tools liegen dabei in

mkcramfs	in	/WRT…/WRTGS…/realease/src/linux/linux/scripts/cramfs
trx	in	/WRT…/WRTGS…/realease/tools

(dies bitte überprüfen)

4.1 Filesystem erstellen (cramfs)

cd /opt/WRT…/WRTGS…/realease/src/linux/linux/scripts/cramfs
mkcramfs /opt/root root.cramfs

- die Datei cramfs und die Datei piggy.gz (in /opt) in das Verzeichnis
/opt/WRT…/WRTGS …/realease/tools kopieren

4.2 Kernel und Filesystem kombinieren

cd /opt/WRT…/WRTGS…/realease/tools
trx -o myfirmware.wdf piggy.gz root.cramfs

- die neue Firmware-Datei wird erstellt

(eine Fehlermeldung, wie warning: increasing offset (wert1) (wert2) kann wohl vernachlässigt werden, solange der Wert2 nicht um mehr als 3 Stellen größer ist, als der Wert1)

Die neue Firmware ist fertig!