



CCcam loggen mit Boardmitteln

In dieser Anleitung werde ich euch erklären, wie ihr den Trace/Debug im laufenden Betrieb, mit Boardmitteln, überprüfen könnt. Ihr braucht dazu keine großen Änderungen an eurem PC oder Dream machen.

Voraussetzungen:

- CCcamServer
- Receiver (z.B. Dreambox) mit einem Syslog-Daemon und Logread (z.B. Gemini, sonstiges CVS)

Überprüfen der Dreambox

Als erstes müssen wir schauen, ob ein Syslogd und Logreader auf dem Receiver läuft. Benutzer vom Gemini, LT-Image, iCVS und OoZooN brauchen es nicht testen, die haben es auf jeden Fall. Dazu verbinden wir uns mittels Telnet auf unsere Box und geben folgendes ein **busybox**.

A screenshot of a PuTTY terminal window titled '192.168.178.57 - PuTTY'. The terminal shows the command 'root@dm8000:~# busybox' and the output of the 'busybox' command. The output includes the version 'BusyBox v1.01 (2009.01.27-20:17+0000) multi-call binary', usage instructions, a description of BusyBox as a multi-call binary, and a list of currently defined functions. A red circle highlights the word 'logread' in the list of functions. The terminal prompt 'root@dm8000:~#' is visible at the bottom.

```
192.168.178.57 - PuTTY
root@dm8000:~# busybox
BusyBox v1.01 (2009.01.27-20:17+0000) multi-call binary

Usage: busybox [function] [arguments]...
or: [function] [arguments]...

BusyBox is a multi-call binary that combines many common Unix
utilities into a single executable. Most people will create a
link to busybox for each function they wish to use and BusyBox
will act like whatever it was invoked as!

Currently defined functions:
[, addgroup, adduser, ar, arping, ash, awk, basename, bunzip2,
busybox, bzip, cat, chgrp, chmod, chown, chroot, chvt, clear,
cp, cpio, crond, crontab, cut, date, dc, dd, deallocvt, delgroup,
deluser, df, dirname, dmesg, dos2unix, du, dumpkmap, echo, egrep,
env, expr, false, fbset, fgrep, find, free, ftpget, getkey, getty,
grep, gunzip, gzip, hdparm, head, hexdump, hostname, httpd, id,
ifconfig, ifdown, ifup, insmod, ip, kill, killall, klogd, ln,
loadfont, loadkmap, logger, login, logname, logread, logsetup,
ls, lsmmod, md5sum, mkdir, mkfifo, mknod, mkswap, mktmp, modprobe,
more, mount, mv, nc, netstat, nslookup, od, openvt, passwd, patch,
pidof, ping, pivot_root, printf, ps, pwd, readlink, renice, reset,
rm, rmdir, rmmmod, route, run-parts, rx, sed, seq, sh, shasum,
sleep, sort, start-stop-daemon, strings, stty, su, sulogin, swapoff,
swapon, sync, sysctl, syslogd, tail, tar, tee, telnet, telnetd,
test, time, top, touch, tr, traceroute, true, tty, udhcpc, umount,
uname, uniq, unix2dos, unzip, uptime, uudecode, uuencode, vi,
vlock, watch, wc, wget, which, who, whoami, xargs, yes, zcat

root@dm8000:~#
```

Dort müsste unter "Currently defined functions" **syslogd** und **logread** stehen. Sollte das nicht der Fall sein, könnt ihr jetzt aufhören und in den Boards suchen, ob es eine Möglichkeit gibt Syslogd und logread auf eure Box zu bekommen.

Steht bei euch Syslogd und Logread geht es jetzt weiter.

Änderungen an der CCcam.cfg

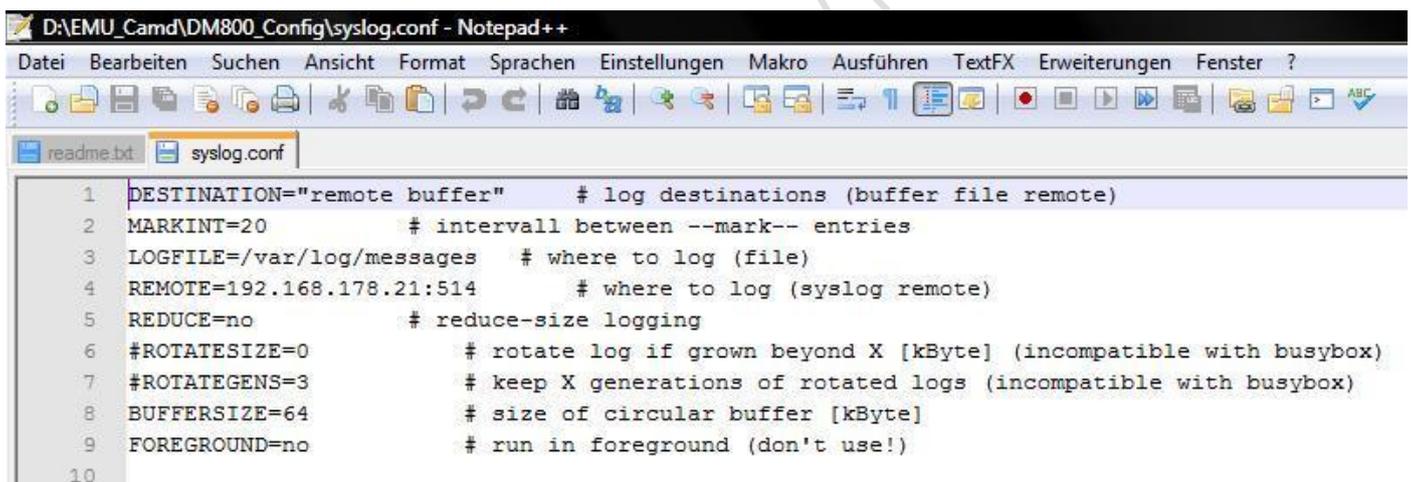
Jetzt nehmen wir uns die CCcam.cfg vor und aktivieren dort den Debug. Dazu fügen wir folgendes in die CCcam.cfg ein bzw. entfernen das Rautezeichen (#) → DEBUG : yes

Schiebt die Config wieder auf die Box und startet eure CCcam neu.

Syslog Config anpassen

Ladet euch mittels FTP die syslog.conf auf eure Box. Bei meiner DM8000 liegt diese unter /etc.

Öffnet diese mit einem linuxfähigen Editor. Wir brauchen nur in der obersten Zeile eine Änderung vornehmen.



```
D:\EMU_Camd\DM800_Config\syslog.conf - Notepad++
Datei Bearbeiten Suchen Ansicht Format Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?
readme.txt syslog.conf
1 DESTINATION="remote buffer" # log destinations (buffer file remote)
2 MARKINT=20 # intervall between --mark-- entries
3 LOGFILE=/var/log/messages # where to log (file)
4 REMOTE=192.168.178.21:514 # where to log (syslog remote)
5 REDUCE=no # reduce-size logging
6 #ROTATESIZE=0 # rotate log if grown beyond X [kByte] (incompatible with busybox)
7 #ROTATEGENS=3 # keep X generations of rotated logs (incompatible with busybox)
8 BUFFERSIZE=64 # size of circular buffer [kByte]
9 FOREGROUND=no # run in foreground (don't use!)
10
```

Und zwar muss dort zusätzlich **buffer** drin stehen. Jetzt ladet die Config wieder auf eure Box und startet euren Syslog Daemon neu.

Ich muss dazu per Telnet in das Verzeichnis /etc/init.d wechseln. Dort starte ich dann durch die Eingabe **syslog restart** den Daemon neu.

```
=====
Welcome to your
=====

root@dm8000:~# cd /etc/init.d
root@dm8000:/etc/init.d# ls
autofs          dbus-1          networking      reboot          syslog
avahi-daemon    dropbear        ppp             rmnologin      umountfs
bootup          halt            rc              sendsigs        vsftpd
busybox-cron    inetd           rcS             single
root@dm8000:/etc/init.d# syslog restart
Stopping syslogd/klogd: stopped syslogd (pid 824).
stopped klogd (pid 826).
done
Starting syslogd/klogd: done
root@dm8000:/etc/init.d# █
```

Jetzt könnt ihr durch Eingabe von **logread** den Buffer auslesen. Es sollte dann so aussehen wie auf dem Bild!

```
root@dm8000:/etc/init.d# logread
Mar 7 14:31:38 dm8000 daemon.debug CCcam: emm already handled (found in cache)
Mar 7 14:31:38 dm8000 daemon.debug CCcam: remote emm -> 192.168.178.55:17025 0x1702(0x000)
Mar 7 14:31:38 dm8000 daemon.debug CCcam: remote emm <- 192.168.178.55:17025 ok (took 0.0036 seconds)
Mar 7 14:31:39 dm8000 daemon.debug CCcam: local ecm <- card /dev/sci1 ok
Mar 7 14:31:39 dm8000 daemon.debug CCcam: client ecm request for handler 0x65 0x1702(0x0) sid 0xa ok (took 0.7911 seconds)
Mar 7 14:31:39 dm8000 daemon.debug CCcam: nagra2 emu can't decode emm (id: 3b) no RSA key
Mar 7 14:31:39 dm8000 daemon.debug CCcam: local emm -> card /dev/sci0 0x1702(0x000)
Mar 7 14:31:40 dm8000 daemon.debug CCcam: local emm <- card /dev/sci0 ok
Mar 7 14:31:40 dm8000 daemon.debug CCcam: emm already handled (found in cache)
Mar 7 14:31:40 dm8000 daemon.debug CCcam: remote emm -> 192.168.178.55:17025 0x1702(0x000)
Mar 7 14:31:40 dm8000 daemon.debug CCcam: remote emm <- 192.168.178.55:17025 ok (took 0.0036 seconds)
Mar 7 14:31:42 dm8000 daemon.debug CCcam: chid 0001
Mar 7 14:31:42 dm8000 daemon.debug CCcam: local ecm -> card /dev/sci0 0x1702(0x000) sid 0x298
Mar 7 14:31:43 dm8000 daemon.debug CCcam: local ecm -> card /dev/sci1 0x1702(0x000) sid 0x00e
Mar 7 14:31:43 dm8000 daemon.debug CCcam: local ecm <- card /dev/sci0 ok
Mar 7 14:31:43 dm8000 daemon.debug CCcam: cam[0] ecm odd ok caid:0x1702 id:0x0 pid:0x1796 Premiere/Premiere Star (19E) [tunneled Nagra] (took 0.7733 seconds)
Mar 7 14:31:43 dm8000 daemon.debug CCcam: emm already handled (found in cache)
Mar 7 14:31:43 dm8000 daemon.debug CCcam: emm already handled (found in cache)
Mar 7 14:31:43 dm8000 daemon.debug CCcam: remote emm -> 192.168.178.55:17025 0x1702(0x000)
Mar 7 14:31:43 dm8000 daemon.debug CCcam: remote emm <- 192.168.178.55:17025 ok (took 0.0025 seconds)
Mar 7 14:31:44 dm8000 daemon.debug CCcam: local ecm <- card /dev/sci1 ok
```

Wollt ihr eine permanente Ausgabe, gebt bitte als zusätzliches Parameter **-f** ein. Dadurch erhaltet ihr einen ständigen Datenstrom. Diesen könnt ihr mittels **[strg] + C** wieder beenden.

Viel Spaß wünscht Euch [Cuuiru](#)

