

AqBanking - Bug #255

Hbci mit chipkarte (Volksbank): Fehler in libchipcard: "Unexpected tag length modifier"

07.01.2022 14:05 - ErwinRieger

Status:	Closed	Beginn:	07.01.2022
Priorität:	Normal	Abgabedatum:	
Kategorie:	Libchipcard	Anwendung:	andere
Betriebssystem:	Linux	Version der Anwendung:	aqhbc-tool4 (aq: 6.4.1.0, gwen: 5.8.1.0)
AqBanking-Version:	6.4.1.0		

Beschreibung

Hallo,

ich versuche grad, mit aqbanking einen "HBCI mit chipkarte" zugang zur volksbank (laupheim) einzurichten (von der kommandozeile aus).

Dabei tritt leider der folgende fehler auf und das hbci-tool beendet sich darauf hin:

```
...
3:2022/01/05 12-38-46:libchipcard(32643):msgengine.c: 513: Unexpected tag length modifier b0
...
```

Es wird ein cyberjack kartenleser unter debian buster verwendet. Ich habe zwei hbci chipkarten von der volksbank (eine inzwischen gesperrte von 2019 und eine aktuelle von 2021), und bei beiden tritt der fehler auf (obige medung mit der 2019-er karte, die 2021-er karte meldet einen anderen 'length modifier' wert von 85).

Das verwendete kommando lautet:

```
aqhbc-tool4 adduser -t zkacard --hbciversion=300 -N "&lt;name&gt;";
```

Zunächst läuft alles ganz gut, und ich werde zur passwort-eingabe am kartenleser aufgefordert. Nach dem eingeben der richtigen pin bricht dann aqhbc-tool4 mit der oben genannten fehlermeldung ab.

Versionen:

```
aqbanking-cli versions
Versions:
  AqBanking-CLI: 6.4.1
  Gwenhywfar   : 5.8.1.0
  AqBanking    : 6.4.1.0
```

Selbst kompiliert aus den quellen:

```
gwenhywfar-5.8.1
libchipcard-5.99.1beta
aqbanking-6.4.1
```

Ich habe eine logdatei und mitschnitte zweier gdb-sessions, ich werde die an martin schicken... Ausserdem kann ich euch die 2019-er chipkarte zur verfügung stellen zum testen, einfach melden :-)

Ich habe bereits etwas genauer hingeschaut und habe folgendes herausgefunden:

In libchipcard:LC_MsgEngine_TypeRead() wird versucht, ein "BER-TLV data object" einzulesen und das geht richtig schief :-(

Zunächst wird ein "tag field" mit dem wert 0xf0 gelesen, laut <https://cardwerk.com/iso7816-4-annex-d-use-of-basic-encoding-rules-asn-1/> ein gültiger wert.

Danach kommt das "Length field" byte mit dem wert 0xb0 (bzw. 0x85 bei der zweiten karte). Das ist voll daneben, weil darauf 0x30 (bzw. 0x5) längen-bytes folgen müssten - es sind aber wohl maximal weitere vier bytes erlaubt. Libchipcard kann nur zwei weitere längenbytes, so daß nur ein "Length field" von

0x80, 0x81 und 0x82 an dieser stelle erlaubt sind...

Hier ein auszug aus einer gdb session (memory dump), gezeigt werden die ersten 100 bytes der eingelesenen daten (pointer "p"):

```
(gdb) x/100xb p
0x555555594220: 0xf0 0xb0 0xc0 0x03 0x30 0x30 0x32 0xe1
0x555555594228: 0x51 0xc1 0x14 0x56 0x52 0x20 0x4c 0x61
0x555555594230: 0x75 0x70 0x68 0x65 0x69 0x6d 0x2d 0x49
0x555555594238: 0x6c 0x6c 0x65 0x72 0x74 0x61 0x6c 0xc2
0x555555594240: 0x03 0x32 0x38 0x30 0xc3 0x08 0x36 0x35
0x555555594248: 0x34 0x39 0x31 0x33 0x32 0x30 0xc4 0x27
0x555555594250: 0x30 0x30 0x37 0x30 0x30 0x31 0x03 0xd4
0x555555594258: 0x95 0x7a 0x38 0x75 0x17 0xfa 0x8e 0xd2
0x555555594260: 0x32 0x46 0x58 0xf4 0x9f 0xa7 0xdd 0x63
0x555555594268: 0x6d 0xb9 0x62 0x72 0x00 0x50 0x90 0x87
0x555555594270: 0x0f 0xab 0x82 0x32 0xd4 0x23 0x8a 0xc5
0x555555594278: 0x01 0x04 0xe2 0x16 0xc6 0x01 0x02 0xc7
0x555555594280: 0x11 0x68 0x62 0x63
(gdb)
```

Gut zu sehen: "tag field" 0xf0 und "length modifier" 0xb0.

Wenn man sich den speicherbereich als string ausgeben lässt, so kann man darin den bank-namen (VR Laupheim-Illertal) erkennen:

```
(gdb) p p
$11 = 0x555555594220 "\360\260\300\003\060\060\062\341Q\301\024VR Laupheim-Illertal\302\003\062\070\060\303\b65491320\304'007001\003 z8u\027\372\216\322\062FX\364\237\247\335cm\271br"
```

Es scheint also so zu sein, daß hier an der falschen stelle gelesen wird oder daß der inhalt des daten-puffers nicht stimmt, "memory problem".

Um mit der hbcu einrichtung weiterzukommen, habe ich testweise die fehlerbehandlung in LC_MsgEngine_TypeRead() (zeile 513) ausgeschaltet und die länge des datenblocks einfach auf die anzahl der verbleibenden bytes gesetzt:

```
...
else {
DBG_ERROR(LC_LOGDOMAIN, "Unexpected tag length modifier %02x", j);
// return -1;
j = size-pos-1;
DBG_ERROR(LC_LOGDOMAIN, "Hack, setting len to %02x, pos: 0x%x, size: 0x%x", j, pos, size);
}
...
```

Mit dieser "gehackten" libchipcard war es mir dann möglich, einen hbcu benutzer anzulegen und sogar die "bank synchronisation" durchzuführen! :-)

Soweit, so gut :-)

Historie

#1 - 07.01.2022 14:26 - ErwinRieger

Vorgehensweise wie unter https://www.aquamaniac.de/rdm/projects/aqbanking/wiki/RDH7_einrichten beschrieben.

#2 - 07.01.2022 15:34 - ErwinRieger

Na super, heute nochmal probiert und es tritt kein fehler mehr auf!? :-o

Gleiche software, zwei verschiedene kartenleser und zwei verschiedene chipkarten getestet, alles ok, kein fehler mehr...

Sporadischer fehler oder problem im BS oder treiber/pcsc?

Werde hier berichten falls problem wieder auftaucht...

#3 - 16.02.2022 09:30 - martin

- Status wurde von New zu Feedback geändert

Interessanter Bugreport... Ist das noch mal aufgetreten? Hattest Du bei "make" vorher "make clean" gemacht? Scheinbar gibt es ein Problem in der Build-Kette, wobei aqbanking nicht komplett neugebaut wird bei Änderungen. Ein Grund, warum ich gwbuild geschrieben habe,

#4 - 17.02.2022 16:35 - ErwinRieger

Hallo, nein, problem ist nicht mehr aufgetreten.

Make clean habe ich zunächst nicht gemacht, da ich ja aus den frischen sourcen heraus gebaut habe.

In der folge hab ich dann mehrmals compiliert und installiert und dabei auch mal "make clean" gemacht und das ziel-installations vezeichnis gelöscht, aber die genaue abfolge kann ich nicht mehr nachvollziehen.

Schon denkbar, daß es mit dem buildvorgang zu tun hat.

#5 - 17.02.2022 16:39 - martin

- Status wurde von Feedback zu Closed geändert