DCMTK - Patch #847

Fix possible buffer overflows when parsing an A-ASSOCIATE packet

2018-08-30 09:25 - Marco Eichelberg

Status:	Closed	**Start date:**	2018-08-30
Priority:	Normal	**Due date:**	
Assignee:	Marco Eichelberg	**% Done:**	100%
Category:	Library	**Estimated time:**	0:00 hour
Target version:	3.6.4		
Module:	dcmnet	**Compiler:**	
Operating System:			

Description

Buffer overflows have been detected by means of fuzz testing in DCMTK's routines that process A-ASSOCIATE packets.
The parse methods return the number of bytes processed, which is subtracted from the number of bytes left.
There is no check whether the number of bytes left is actually larger than the number of bytes processed
(which may not be the case if malformed packets are processed). In that case, an integer underflow occurs,
resulting in a number of bytes left that is much too high, which in turn causes a buffer overflow.

In order to reproduce the problem, apply the following patch to DCMTK 3.6.3
and then run echoscu many times against an SCP like storescp. This should trigger a crash.

```
--- dcmtk-3.6.3/dcmnet/libsrc/dulconst.cc        2018-02-05 18:58:13.000000000 +0100
+++ dcmtk-3.6.3-patched/dcmnet/libsrc/dulconst.cc    2018-08-23 08:03:38.979760526 +0200
@@ -936,7 +936,7 @@
     unsigned long compatMode = dcmEnableBackwardCompatibility.get();
     max->type = DUL_TYPEMAXLENGTH;
     max->rsv1 = 0;
-    max->length = 4;
+    max->length = 63412;
     if (compatMode & 0x8000) max->maxLength = DUL_DULCOMPAT | DUL_DIMSECOMPAT | compatMode;
     else max->maxLength = maxPDU;
     *rtnLen = 8;
```

The attached patch fixes the problem.

This patch was submitted by Peter Klotz <Peter.Klotz@ith-icoserve.com>.

History

#1 - 2018-08-30 09:29 - Marco Eichelberg

- Status changed from New to Closed

Closed by commit #f6f40f639.

#2 - 2018-08-30 09:29 - Marco Eichelberg

- % Done changed from 0 to 100

Files

dcmtk-3.6.3-Parsing.patch	5.46 KB	2018-08-30	Marco Eichelberg