Bug #1208 CLOSED

Heap Buffer Overflow in XMLNode::parseFile() - ofxml.cc

Added by Jörg Riesmeier about 1 month ago. Updated 15 days ago.

Status:	Closed	**Start date:**	2026-05-19
Priority:	High	**Due date:**	
Assignee:	Marco Eichelberg	**% Done:**	100%
Category:	Library	**Estimated time:**	1:00 h
Target version:	3.7.1		
Module:	ofstd	**Compiler:**	
Operating System:			

Description

On 2026-05-16, Cristhian Daniel Rivas Zúñiga and Sebastian Andres Muñoz Morera reported the following:

=== *CUT* ===

REPORTERS

Cristhian Daniel Rivas Zúñiga and Sebastian Andres Muñoz Morera
Insituto Tecnológico de Costa Rica

SUMMARY

A heap buffer overflow exists in XMLNode::parseFile() in ofstd/libsrc/ofxml.cc. When the function is called with a FIFO (named pipe) as input — which is a supported and documented use case via cda2dcm — the ftell() call returns -1 to signal an error. The code does not check for this error condition (it only checks for l == 0), causing malloc(3) to be called followed by fread() with a size_t-casted -1 value, resulting in an attempt to read up to SIZE_MAX bytes into a 3-byte heap buffer.

CWE: CWE-122 (Heap-based Buffer Overflow)
CVSS (estimated): 8.1 HIGH — AV:L/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

AFFECTED VERSIONS

DCMTK 3.6.7, 3.6.8, 3.6.9, 3.7.0, and current master branch.
The vulnerable lines (1962–1969) are identical across all versions checked.

ROOT CAUSE

File: ofstd/libsrc/ofxml.cc
Function: XMLNode::parseFile()
Lines: ~1962–1969

The vulnerability chain:

1. fseek(f, 0, SEEK_END) succeeds on a FIFO but positions nothing meaningful.
2. int l = OFstatic_cast(int, ftell(f)) → ftell() returns -1 on a FIFO (POSIX: "undefined for non-seekable files").
3. if (!l) { ... return emptyXMLNode; } → This check only catches l 0. It does NOT catch l -1, so execution continues.
4. malloc(l + 4) = malloc(-1 + 4) = malloc(3) → A 3-byte heap buffer is allocated.
5. fread(buf, 1, l, f): the value -1 is implicitly cast to size_t, becoming SIZE_MAX (~18 exabytes). fread attempts to read SIZE_MAX bytes into a 3-byte buffer.
6. buf[l]=0; buf[l+1]=0; buf[l+2]=0; buf[l+3]=0; → Writes to buf[-1] through buf[2], corrupting adjacent heap memory.

CALL CHAIN
---
cda2dcm (main)
→ OFstub_main() with --filetype-cda flag
→ DcmEncapsulatedDocument::parseArguments()
→ DcmEncapsulatedDocument::insertEncapsulatedDocument() [sets ftype_ = DT_cdaDocument]
→ DcmEncapsulatedDocument::formatSpecificProcessing()
→ DcmEncapsulatedDocument::getCDAData()
→ XMLNode::parseFile() ← VULNERABLE

PROOF OF CONCEPT

Prerequisites: DCMTK built with AddressSanitizer (-fsanitize=address).

Terminal 1 (reader):
export DCMDICTPATH=/path/to/dcmtk/dcmdata/data/dicom.dic
mkfifo /tmp/exploit.xml
./build-asan/bin/cda2dcm /tmp/exploit.xml out.dcm

Terminal 2 (writer, after ~1 second):
echo "<ClinicalDocument></ClinicalDocument>" > /tmp/exploit.xml

Expected output:
==ERROR: AddressSanitizer: heap-buffer-overflow
READ of size 18446744073709551615 at 0x...

The crash is 100% reproducible across all tested versions.

[Screenshot removed]

SUGGESTED FIX

In ofstd/libsrc/ofxml.cc, change line ~1964 from:

if (!l) { if (pResults) pResults->error=eXMLErrorEmpty; fclose(f); return emptyXMLNode; }

To:

if (l <= 0) { if (pResults) pResults->error=eXMLErrorEmpty; fclose(f); return emptyXMLNode; }

This single character change causes the function to correctly handle the ftell() error return value of -1.

[...]

History | Notes | Property changes

Updated by Jörg Riesmeier about 1 month ago

- **Subject** changed from *Heap Buffer Overflow in XMLNode::parseFile() — ofxml.cc* to *Heap Buffer Overflow in XMLNode::parseFile() - ofxml.cc*
- **Description** updated (diff)

Updated by Jörg Riesmeier about 1 month ago

- **Description** updated (diff)

Updated by Marco Eichelberg about 1 month ago

- **Status** changed from *New* to *Closed*
- **Assignee** set to *Marco Eichelberg*
- **% Done** changed from *0* to *100*
- **Estimated time** set to *1:00 h*

Closed by commit #1d4b3815c.

Updated by Marco Eichelberg 15 days ago

- **Private** changed from *Yes* to *No*