

[NT] MODPlug Tracker/OpenMPT/Libmodplug Stack And Heap Overflows

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2006-08/msg00078.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxx>
 - *Date:* 21 Aug 2006 14:08:18 +0200
-

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>
-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.
<http://www.securiteam.com/maillinglist.html>

MODPlug Tracker/OpenMPT/Libmodplug Stack And Heap Overflows

SUMMARY

MODPlug Tracker, and naturally its more recent open source version OpenMPT, is one of the coolest music trackers which supports many music module types too. libmodplug instead is a Linux library created from the OpenMPT source and mainly used for the ModPlug-XMMS plugin.

Due to lacking of input validation there are exploitable stack and heap overflows in MODPlug.

DETAILS

Global buffer overflows in ReadITProject:

All the text fields in the ITP files are not sanitized so is possible to overflow the global variables through this function and possibly executing malicious code (confirmed in my tests). Note: ITP files are not supported in libmodplug

From soundlib/Load_it.cpp:

[NT] MODPlug Tracker/OpenMPT/Libmodplug Stack And Heap Overflows

```
BOOL CSoundFile::ReadITProject(LPCBYTE lpStream, DWORD dwMemLength)
{
...
// Song name

// name string length
memcpy(&id,lpStream+streamPos,sizeof(DWORD));
len = id;
streamPos += sizeof(DWORD);

// name string
memcpy(&m_szNames[0],lpStream+streamPos,len);
streamPos += len;
...
(other overflows)
...
```

Heap overflow in ReadSample:

In some modules the ReadSample function can be used to cause a heap overflow through an invalid nLength value. As visible by the code below, nLength is incremented of 6 bytes (mem) and in some cases its value is multiplied by two, the final value is then used to allocate pIns->pSample (FYI AllocateSample allocates "(nbytes + 39) & ~7" and returns the pointer plus 16). An attacker, after having forced the program to allocate 0 bytes, will be able to overflow the memory through the memcpy instructions which will copy (depending by nFlags) all the remaining bytes in the file. The best type of module for exploiting this vulnerability seems to be AMF.

From soundlib/Sndfile.cpp:

```
UINT CSoundFile::ReadSample(MODINSTRUMENT *pIns, UINT nFlags, LPCSTR
lpMemFile, DWORD dwMemLength)
//-----
{
UINT len = 0, mem = pIns->nLength+6;

if ((!pIns) || (pIns->nLength < 4) || (!lpMemFile)) return 0;
if (pIns->nLength > MAX_SAMPLE_LENGTH) pIns->nLength =
MAX_SAMPLE_LENGTH;
...
if ((pIns->pSample = AllocateSample(mem)) == NULL)
...
default:
len = pIns->nLength;
if (len > dwMemLength) len = pIns->nLength = dwMemLength;
memcpy(pIns->pSample, lpMemFile, len);
}
...
```

[NT] MODPlug Tracker/OpenMPT/Libmodplug Stack And Heap Overflows

Proof of concept:

<<http://aluigi.org/poc/mptho.zip>> <http://aluigi.org/poc/mptho.zip>

Fix:

A new version will be released soon.

ADDITIONAL INFORMATION

The information has been provided by <<mailto:aluigi@xxxxxxxxxxxxxx>> Luigi Auriemma.

The original article can be found at:

<<http://aluigi.org>> <http://aluigi.org>

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@xxxxxxxxxxxxxx

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxx

=====
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.