

[NT] Visual Studio Buffer Overflow

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2006-03/msg00004.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxxxx>
 - *Date:* 5 Mar 2006 18:52:09 +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>

Visual Studio Buffer Overflow

SUMMARY

A Buffer Overflow Vulnerability exists with Visual Studio parsing of some files.

DETAILS

Vulnerable Systems:

- * Microsoft Visual Studio 6.0 (with latest Service Pack 6)
- * Microsoft Development Environment 6.0 (SP6) (Microsoft Visual InterDev 6.0)

A Buffer Overflow Vulnerability exists for the following file formats of affected products:

- * Visual Studio Database Project File (.dbp)
- * Visual Studio Solution (.sln)

The vulnerability is caused due to a boundary error within the handling of a ".dbp" file (.sln files are also affected) that contains an overly long string in the "DataProject" field. This can be exploited to cause a stack-based buffer overflow and allows arbitrary code execution when a

[NT] Visual Studio Buffer Overflow

malicious ".dbp" file is opened.

A specially crafted project file can overwrite a stack based buffer allowing for fully EIP register control resulting in code execution and compromising the user's system.

Example:

```
# Microsoft Developer Studio Project File – Database Project
Begin DataProject = "ProjectName"
End
```

Carriage return and line feed (0x0d and 0x0a) characters and some others (0x00 ...) can not be used in project name variable.

An example .dbp file which overwrites EIP register:

```
# Microsoft Developer Studio Project File – Database Project
Begin DataProject =
"Project1AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA123456AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
End
```

The length must be 384 bytes long. Otherwise other registers will be overwritten differently and exploitation method will be changed. So 384 bytes long length is the most suitable way.

In this example when file is opened:

XXXX (0x58585858) characters will overwrite EIP.
And 123456AAAA... (3132333435364141... in hex) bytes will be on ESP.

So an attacker could create a malicious .dbp project file which includes a payload which on ESP and EIP should point to this shellcode with a loaded moduls jmp esp or call esp opcodes.

Proof of Concept:

The local path length of the dbp file changes the arrangement of malformed data. So, exploit has to re-align the data for total path length.

Copy the following file as c:\deneme\Project1.dbp

<<http://www.spyinstructors.com/kozan/poc/vuln.dbp>>

<http://www.spyinstructors.com/kozan/poc/vuln.dbp>

ADDITIONAL INFORMATION

The information has been provided by <<mailto:kozan@xxxxxxxxxxxxxxxxxxxxxx>>
kozan.

The original article can be found at:

<<http://www.spyinstructors.com/show.php?name=Advisories&pa=showpage&pid=73>>

[NT] Visual Studio Buffer Overflow

<http://www.spyinstructors.com/show.php?name=Advisories&pa=showpage&pid=73>

=====

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@xxxxxxxxxxxxxxxx

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

=====

=====

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.