Pre-Auth DOS

High **dyemanov** published **GHSA-6crx-4g37-7j49** 3 days ago

Package

Firebird server

Affected versions

All versions starting with FB3.0.0

Patched versions

6.0, 5.0.4, 4.0.7, 3.0.14

Description

Summary

Incorrect order of `CNCT_specific_data` sequence segments causes **SIGSEGV**.

Details

The `CNCT_specific_data` is a default one byte tag, limited 256 bytes of content (1 byte sequence number and 254 bytes of data). When a data is bigger than one tag cup, it'll be splitted on segments and pushed as in the next `CNCT_specific_data` tags. Every tag has his own number to recover original sequence (same as TCP protocol). In an auth process `CNCT_specific_data` tag provide some data, needed for user verification on server.

The main problem is that the segments must have a strict order, from small numbers to large, otherwise it will cause a `SIGSEGV` error, since the `grow(const size_type newCount)` method of the `Array` class tries to `memset` the data by its offset.

PoC

Imagine the server receives two segments, numbered `0xFD (offset 253 bytes)` and `0 (or any other number less than 253)`. The first segment attempts to ensure capacity (`sizeof(T) * newcapacity`) and then fills it with zeros. If the first segment is numbered `0xFD`, this means that space must be allocated for all segments preceding it. Each segment is `254 bytes in size`, meaning the total number of bytes that must be allocated is `254 * (253 + 1) = 64,516`.

When the second segment is processed, Array will not call `ensureCapacity(size_type newcapacity)` because space for the first segment was previously allocated. However, attempting to fill it with zeros will cause a `SIGSEGV` error because `grow(const size_type newCount)` is designed to append data to the end of the array. If we try to increase the array's size to a value smaller than its current value, the `sizeof(T) * (newCount - count)` formula, designed to fill the array with new data, will get negative value.

Impact

The Firebird server is vulnerable because anyone who knows only the IP and port can easily take it down.

exploit.py

Severity

High 8.2 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	None
Integrity	Low
Availability	High

Learn more about base metrics

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:H

CVE ID

No known CVE

Weaknesses

- CWE-119
- CWE-787