Segfault and data corruption caused by negative indexing in TFLite

High **mihaimaruseac** published **GHSA-q4qf-3fc6-8x34** on Sep 24, 2020

Package

tensorflow-lite (tensorflow)

Affected versions	Patched versions
< 2.3.0	1.15.4, 2.0.3, 2.1.2, 2.2.1, 2.3.1

Description

Impact

To mimic Python's indexing with negative values, TFLite uses `ResolveAxis` to convert negative values to positive indices. However, the only check that the converted index is now valid is only present in debug builds:

tensorflow/tensorflow/lite/kernels/internal/reference/reduce.h
Lines 68 to 72 in `0e68f4d`

```
68    // Handle negative index. A positive index 'p_idx' can be represented as a
69    // negative index 'n_idx' as: n_idx = p_idx-num_dims
70    // eg: For num_dims=3, [0, 1, 2] is the same as [-3, -2, -1]  */
71    int current = axis[idx] < 0 ? (axis[idx] + num_dims) : axis[idx];
72    TFLITE_DCHECK(current >= 0 && current < num_dims);
```

If the `DCHECK` does not trigger, then code execution moves ahead with a negative index. This, in turn, results in accessing data out of bounds which results in segfaults and/or data corruption.

Patches

We have patched the issue in `2d88f47` and will release patch releases for all versions between 1.15 and 2.3.

We recommend users to upgrade to TensorFlow 1.15.4, 2.0.3, 2.1.2, 2.2.1, or 2.3.1.

For more information

Please consult our security guide for more information regarding the security model and how to contact us with issues and questions.

Attribution

This vulnerability has been reported by members of the Aivul Team from Qihoo 360.

Severity

High

CVE ID

CVE-2020-15207

Weaknesses

No CWEs