Division by zero in padding computation in TFLite

Low **mihaimaruseac** published **GHSA-mv78-g7wq-mhp4** on May 12, 2021

Package

tensorflow-lite (pip)

Affected versions	Patched versions
< 2.5.0	2.1.4, 2.2.3, 2.3.3, 2.4.2

Description

Impact

The TFLite computation for size of output after padding, `ComputeOutSize`, does not check that the `stride` argument is not 0 before doing the division.

```
inline int ComputeOutSize(TfLitePadding padding, int image_size,
                          int filter_size, int stride, int dilation_rate = 1) {
  int effective_filter_size = (filter_size - 1) * dilation_rate + 1;
  switch (padding) {
    case kTfLitePaddingSame:
      return (image_size + stride - 1) / stride;
    case kTfLitePaddingValid:
      return (image_size + stride - effective_filter_size) / stride;
    default:
      return 0;
  }
}
```

Users can craft special models such that `ComputeOutSize` is called with `stride` set to 0.

Patches

We have patched the issue in GitHub commit 49847ae69a4e1a97ae7f2db5e217c77721e37948.

The fix will be included in TensorFlow 2.5.0. We will also cherrypick this commit on TensorFlow 2.4.2, TensorFlow 2.3.3, TensorFlow 2.2.3 and TensorFlow 2.1.4, as these are also affected and still in supported range.

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

Low

CVE ID

CVE-2021-29585

Weaknesses

No CWEs