Data corruption due to dimension mismatch in TFLite

High **mihaimaruseac** published **GHSA-mxjj-953w-2c2v** 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

When determining the common dimension size of two tensors, TFLite uses a `DCHECK` which is no-op outside of debug compilation modes:

tensorflow/tensorflow/lite/kernels/internal/types.h
Lines 437 to 442 in `0e68f4d`

```
437   // Get common shape dim, DCHECKing that they all agree.
438   inline int MatchingDim(const RuntimeShape& shape1, int index1,
439                          const RuntimeShape& shape2, int index2) {
440     TFLITE_DCHECK_EQ(shape1.Dims(index1), shape2.Dims(index2));
441     return shape1.Dims(index1);
442   }
```

Since the function always returns the dimension of the first tensor, malicious attackers can craft cases where this is larger than that of the second tensor. In turn, this would result in reads/writes outside of bounds since the interpreter will wrongly assume that there is enough data in both tensors.

Patches

We have patched the issue in `8ee24e7` 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-2020-15208

Weaknesses

No CWEs