Out of bounds write in TFLite implementation of segment sum

High **mihaimaruseac** published **GHSA-p2cq-cprg-frvm** on Sep 24, 2020

Package

tensorflow-lite (tensorflow)

Affected versions	Patched versions
2.2.0, 2.3.0	2.2.1, 2.3.1

Description

Impact

In TensorFlow Lite models using segment sum can trigger a write out bounds / segmentation fault if the segment ids are not sorted. Code assumes that the segment ids are in increasing order, using the last element of the tensor holding them to determine the dimensionality of output tensor:

tensorflow/tensorflow/lite/kernels/segment_sum.cc
Lines 39 to 44 in 0e68f4d

```
  if (segment_id_size > 0) {
    max_index = segment_ids->data.i32[segment_id_size - 1];
  }
  const int data_rank = NumDimensions(data);
  TfLiteIntArray* output_shape = TfLiteIntArrayCreate(NumDimensions(data));
  output_shape->data[0] = max_index + 1;
```

This results in allocating insufficient memory for the output tensor and in a write outside the bounds of the output array:

tensorflow/tensorflow/lite/kernels/internal/reference/reference_ops.h
Lines 2625 to 2631 in 0e68f4d

```
  memset(output_data, 0, sizeof(T) * output_shape.FlatSize());

  for (int i = 0; i < input_shape.Dims(0); i++) {
    int output_index = segment_ids_data[i];
    for (int j = 0; j < segment_flat_size; ++j) {
      output_data[output_index * segment_flat_size + j] +=
          input_data[i * segment_flat_size + j];
```

This usually results in a segmentation fault, but depending on runtime conditions it can provide for a write gadget to be used in future memory corruption-based exploits.

Patches

We have patched the issue in `204945b` and will release patch releases for all affected versions.

We recommend users to upgrade to TensorFlow 2.2.1, or 2.3.1.

Workarounds

A potential workaround would be to add a custom `Verifier` to the model loading code to ensure that the segment ids are sorted, although this only handles the case when the segment ids are stored statically in the model.

A similar validation could be done if the segment ids are generated at runtime between inference steps.

If the segment ids are generated as outputs of a tensor during inference steps, then there are no possible workaround and users are advised to upgrade to patched code.

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-2020-15214

Weaknesses

No CWEs