Heap out of bounds write in `RaggedBinCount`

Low **mihaimaruseac** published **GHSA-8h46-5m9h-7553** on May 12, 2021

Package

tensorflow, tensorflow-cpu, tensorflow-gpu (pip)

Affected versions	Patched versions
>=2.3.0, < 2.5.0	2.3.3, 2.4.2

Description

Impact

If the `splits` argument of `RaggedBincount` does not specify a valid `SparseTensor`, then an attacker can trigger a heap buffer overflow:

```
import tensorflow as tf
tf.raw_ops.RaggedBincount(splits=[7,8], values= [5, 16, 51, 76, 29, 27, 54, 95],\
                          size= 59, weights= [0, 0, 0, 0, 0, 0, 0, 0],\
                          binary_output=False)
```

This will cause a read from outside the bounds of the `splits` tensor buffer in the implementation of the `RaggedBincount` op:

```
  for (int idx = 0; idx < num_values; ++idx) {
    while (idx >= splits(batch_idx)) {
      batch_idx++;
    }
    ...
    if (bin < size) {
      if (binary_output_) {
        out(batch_idx - 1, bin) = T(1);
      } else {
        T value = (weights_size > 0) ? weights(idx) : T(1);
        out(batch_idx - 1, bin) += value;
      }
    }
  }
```

Before the `for` loop, `batch_idx` is set to 0. The attacker sets `splits(0)` to be 7, hence the `while` loop does not execute and `batch_idx` remains 0. This then results in writing to `out(-1, bin)`, which is before the heap allocated buffer for the output tensor.

Patches

We have patched the issue in GitHub commit eebb96c2830d48597d055d247c0e9aebaea94cd5.

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

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-2021-29514

Weaknesses

No CWEs