Heap buffer overflow in `RaggedBinCount`

Low **mihaimaruseac** published **GHSA-4278-2v5v-65r4** 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=[0], values=[1,1,1,1,1], size=5, weights=[1,2,3,4], 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++;
    }
    ...
  }
```

Before the `for` loop, `batch_idx` is set to 0. The user controls the `splits` array, making it contain only one element, 0. Thus, the code in the `while` loop would increment `batch_idx` and then try to read `splits(1)`, which is outside of bounds.

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-2021-29512

Weaknesses

No CWEs