Segfault due to invalid splits in RaggedCountSparseOutput

Moderate **mihaimaruseac** published **GHSA-x7rp-74x2-mjf3** on Sep 24, 2020

Package

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

Affected versions	Patched versions
2.3.0	2.3.1

Description

Impact

The `RaggedCountSparseOutput` implementation does not validate that the input arguments form a valid ragged tensor. In particular, there is no validation that the values in the `splits` tensor generate a valid partitioning of the `values` tensor. Thus, the following code sets up conditions to cause a heap buffer overflow:

```
auto per_batch_counts = BatchedMap<W>(num_batches);
int batch_idx = 0;
for (int idx = 0; idx < num_values; ++idx) {
  while (idx >= splits_values(batch_idx)) {
    batch_idx++;
  }
  const auto& value = values_values(idx);
  if (value >= 0 && (maxlength_ <= 0 || value < maxlength_)) {
    per_batch_counts[batch_idx - 1][value] = 1;
  }
}
```

A `BatchedMap` is equivalent to a vector where each element is a hashmap. However, if the first element of `splits_values` is not 0, `batch_idx` will never be 1, hence there will be no hashmap at index 0 in `per_batch_counts`. Trying to access that in the user code results in a segmentation fault.

Patches

We have patched the issue in `3cbb917` and will release a patch release.

We recommend users to upgrade to TensorFlow 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 is a variant of GHSA-p5f8-gfw5-33w4

Severity

Moderate

CVE ID

CVE-2020-15200

Weaknesses

No CWEs