Segfault in `SparseCountSparseOutput`

Low **mihaimaruseac** published **GHSA-hr84-fqvp-48mm** 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

Specifying a negative dense shape in `tf.raw_ops.SparseCountSparseOutput` results in a segmentation fault being thrown out from the standard library as `std::vector` invariants are broken.

```
import tensorflow as tf

indices = tf.constant([], shape=[0, 0], dtype=tf.int64)
values = tf.constant([], shape=[0, 0], dtype=tf.int64)
dense_shape = tf.constant([-100, -100, -100], shape=[3], dtype=tf.int64)
weights = tf.constant([], shape=[0, 0], dtype=tf.int64)

tf.raw_ops.SparseCountSparseOutput(indices=indices, values=values, dense_shape=dense_shape, weights=weights, minlength=79, maxlength=96, binary_output=False)
```

This is because the implementation assumes the first element of the dense shape is always positive and uses it to initialize a `BatchedMap<T>` (i.e., `std::vector<absl::flat_hash_map<int64,T>>`) data structure.

```
bool is_1d = shape.NumElements() == 1;
int num_batches = is_1d ? 1 : shape.flat<int64>()(0);
...
auto per_batch_counts = BatchedMap<W>(num_batches);
```

If the `shape` tensor has more than one element, `num_batches` is the first value in `shape`.

Ensuring that the `dense_shape` argument is a valid tensor shape (that is, all elements are non-negative) solves this issue.

Patches

We have patched the issue in GitHub commit c57c0b9f3a4f8684f3489dd9a9ec627ad8b599f5.

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.

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 Yakun Zhang and Ying Wang of Baidu X-Team.

Severity

Low

CVE ID

CVE-2021-29521

Weaknesses

No CWEs