# Heap buffer overflow in weighted sparse count ops

High **mihaimaruseac** published **GHSA-pg59-2f92-5cph** on Sep 24, 2020

Package

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

| Affected versions | Patched versions |
| --- | --- |
| 2.3.0 | 2.3.1 |

## Description

### Impact

The `SparseCountSparseOutput` and `RaggedCountSparseOutput` implementations don't validate that the `weights` tensor has the same shape as the data. The check exists for `DenseCountSparseOutput`, where both tensors are fully specified:

tensorflow/tensorflow/core/kernels/count_ops.cc
Lines 110 to 117 in `0e68f4d`

```
110     if (use_weights) {
111       OP_REQUIRES(
112           context, weights.shape() == data.shape(),
113           errors::InvalidArgument(
114               "Weights and data must have the same shape. Weight shape: ",
115               weights.shape().DebugString(),
116               "; data shape: ", data.shape().DebugString()));
117     }
```

In the sparse and ragged count weights are still accessed in parallel with the data:

tensorflow/tensorflow/core/kernels/count_ops.cc
Lines 199 to 201 in `0e68f4d`

```
199     } else if (use_weights) {
200       per_batch_counts[batch][value] += weight_values(idx);
201     } else {
```

But, since there is no validation, a user passing fewer weights than the values for the tensors can generate a read from outside the bounds of the heap buffer allocated for the weights.

### 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**

High

**CVE ID**

CVE-2020-15196

**Weaknesses**

No CWEs