`CHECK`-fail in `AddManySparseToTensorsMap`

Low **mihaimaruseac** published **GHSA-2cpx-427x-q2c6** on May 12, 2021

Package

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

Affected versions	Patched versions
< 2.5.0	2.1.4, 2.2.3, 2.3.3, 2.4.2

Description

Impact

An attacker can trigger a denial of service via a `CHECK` -fail in `tf.raw_ops.AddManySparseToTensorsMap` :

```
import tensorflow as tf
import numpy as np

sparse_indices = tf.constant(530, shape=[1, 1], dtype=tf.int64)
sparse_values = tf.ones([1], dtype=tf.int64)

shape = tf.Variable(tf.ones([55], dtype=tf.int64))
shape[:8].assign(np.array([855, 901, 429, 892, 892, 852, 93, 96], dtype=np.int64))

tf.raw_ops.AddManySparseToTensorsMap(sparse_indices=sparse_indices,
                     sparse_values=sparse_values,
                     sparse_shape=shape)
```

This is because the implementation takes the values specified in `sparse_shape` as dimensions for the output shape:

```
    TensorShape tensor_input_shape(input_shape->vec<int64>());
```

The `TensorShape` constructor uses a `CHECK` operation which triggers when `InitDims` returns a non-OK status.

```
template <class Shape>
TensorShapeBase<Shape>::TensorShapeBase(gtl::ArraySlice<int64> dim_sizes) {
  set_tag(REP16);
  set_data_type(DT_INVALID);
  TF_CHECK_OK(InitDims(dim_sizes));
}
```

In our scenario, this occurs when adding a dimension from the argument results in overflow:

```
template <class Shape>
Status TensorShapeBase<Shape>::InitDims(gtl::ArraySlice<int64> dim_sizes) {
  ...
  Status status = Status::OK();
  for (int64 s : dim_sizes) {
    status.Update(AddDimWithStatus(internal::SubtleMustCopy(s)));
    if (!status.ok()) {
      return status;
    }
  }
}

template <class Shape>
Status TensorShapeBase<Shape>::AddDimWithStatus(int64 size) {
  ...
  int64 new_num_elements;
  if (kIsPartial && (num_elements() < 0 || size < 0)) {
    new_num_elements = -1;
  } else {
    new_num_elements = MultiplyWithoutOverflow(num_elements(), size);
    if (TF_PREDICT_FALSE(new_num_elements < 0)) {
        return errors::Internal("Encountered overflow when multiplying ",
                                num_elements(), " with ", size,
                                ", result: ", new_num_elements);
      }
  }
  ...
}
```

This is a legacy implementation of the constructor and operations should use `BuildTensorShapeBase` or `AddDimWithStatus` to prevent `CHECK` -failures in the presence of overflows.

Patches

We have patched the issue in GitHub commit 69c68ecbb24dff3fa0e46da0d16c821a2dd22d7c.

The fix will be included in TensorFlow 2.5.0. We will also cherrypick this commit on TensorFlow 2.4.2, TensorFlow 2.3.3, TensorFlow 2.2.3 and TensorFlow 2.1.4, as these are also affected and still in supported range.

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-2021-29523

Weaknesses

No CWEs