Heap buffer overflow in `MaxPool3DGradGrad`

Low **mihaimaruseac** published **GHSA-7cqx-92hp-x6wh** 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.MaxPool3DGradGrad` is vulnerable to a heap buffer overflow:

```
import tensorflow as tf

values = [0.01] * 11
orig_input = tf.constant(values, shape=[11, 1, 1, 1, 1], dtype=tf.float32)
orig_output = tf.constant([0.01], shape=[1, 1, 1, 1, 1], dtype=tf.float32)
grad = tf.constant([0.01], shape=[1, 1, 1, 1, 1], dtype=tf.float32)
ksize = [1, 1, 1, 1, 1]
strides = [1, 1, 1, 1, 1]
padding = "SAME"

tf.raw_ops.MaxPool3DGradGrad(
    orig_input=orig_input, orig_output=orig_output, grad=grad, ksize=ksize,
    strides=strides, padding=padding)
```

The implementation does not check that the initialization of `Pool3dParameters` completes successfully:

```
Pool3dParameters params{context,  ksize_,       stride_,
                        padding_, data_format_, tensor_in.shape()};
```

Since the constructor uses `OP_REQUIRES` to validate conditions, the first assertion that fails interrupts the initialization of `params`, making it contain invalid data. In turn, this might cause a heap buffer overflow, depending on default initialized values.

Patches

We have patched the issue in GitHub commit 63c6a29d0f2d692b247f7bf81f8732d6442fad09.

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-2021-29576

Weaknesses

No CWEs