Division by 0 in `FractionalAvgPool`

Low **mihaimaruseac** published **GHSA-f78g-q7r4-9wcv** 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 runtime division by zero error and denial of service in `tf.raw_ops.FractionalAvgPool` :

```
import tensorflow as tf

value = tf.constant([60], shape=[1, 1, 1, 1], dtype=tf.int32)
pooling_ratio = [1.0, 1.0000014345305555, 1.0, 1.0]
pseudo_random = False
overlapping = False
deterministic = False
seed = 0
seed2 = 0

tf.raw_ops.FractionalAvgPool(
  value=value, pooling_ratio=pooling_ratio, pseudo_random=pseudo_random,
  overlapping=overlapping, deterministic=deterministic, seed=seed, seed2=seed2)
```

This is because the implementation computes a divisor quantity by dividing two user controlled values:

```
for (int i = 0; i < tensor_in_and_out_dims; ++i) {
  output_size[i] = static_cast<int>(std::floor(input_size[i] / pooling_ratio_[i]));
  DCHECK_GT(output_size[i], 0);
}
```

The user controls the values of `input_size[i]` and `pooling_ratio_[i]` (via the `value.shape()` and `pooling_ratio` arguments). If the value in `input_size[i]` is smaller than the `pooling_ratio_[i]` , then the floor operation results in `output_size[i]` being 0. The `DCHECK_GT` line is a no-op outside of debug mode, so in released versions of TF this does not trigger.

Later, these computed values are used as arguments to `GeneratePoolingSequence` . There, the first computation is a division in a modulo operation:

```
std::vector<int64> GeneratePoolingSequence(int input_length, int output_length,
                                           GuardedPhiloxRandom* generator,
                                           bool pseudo_random) {
  ...
  if (input_length % output_length == 0) {
    diff = std::vector<int64>(output_length, input_length / output_length);
  }
  ...
}
```

Since `output_length` can be 0, this results in runtime crashing.

Patches

We have patched the issue in GitHub commit 548b5eaf23685d86f722233d8fbc21d0a4aecb96.

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-2021-29550

Weaknesses

No CWEs