Write to immutable memory region

Low **mihaimaruseac** published **GHSA-hhvc-g5hv-48c6** on Dec 9, 2020

Package

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

Affected versions	Patched versions
< 2.4.0	1.15.5, 2.0.4, 2.1.3, 2.2.2, 2.3.2, 2.4.0

Description

Impact

The `tf.raw_ops.ImmutableConst` operation returns a constant tensor created from a memory mapped file which is assumed immutable. However, if the type of the tensor is not an integral type, the operation crashes the Python interpreter as it tries to write to the memory area:

```
>>> import tensorflow as tf
>>> with open('/tmp/test.txt','w') as f: f.write('a'*128)
>>> tf.raw_ops.ImmutableConst(dtype=tf.string,shape=2,
                              memory_region_name='/tmp/test.txt')
```

If the file is too small, TensorFlow properly returns an error as the memory area has fewer bytes than what is needed for the tensor it creates. However, as soon as there are enough bytes, the above snippet causes a segmentation fault.

This is because the alocator used to return the buffer data is not marked as returning an opaque handle since the needed virtual method is not overriden.

Patches

We have patched the issue in GitHub commit c1e1fc899ad5f8c725dcbb6470069890b5060bc7 and will release TensorFlow 2.4.0 containing the patch. TensorFlow nightly packages after this commit will also have the issue resolved.

Since this issue also impacts TF versions before 2.4, we will patch all releases between 1.15 and 2.3 inclusive.

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 members of the Aivul Team from Qihoo 360.

Severity

Low

CVE ID

CVE-2020-26268

Weaknesses

No CWEs