Memory corruption in dlpack.to_dlpack

High mihaimaruseac published GHSA-rjjg-hgv6-h69v on Sep 24, 2020

Package

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

Affected versions	Patched versions
2.2.0, 2.3.0	2.2.1, 2.3.1

Description

Impact

The implementation of `dlpack.to_dlpack` can be made to use uninitialized memory resulting in further memory corruption. This is because the pybind11 glue code assumes that the argument is a tensor:

tensorflow/tensorflow/python/tfe_wrapper.cc
Line 1361 in 0e68f4d

```
1361      TFE_TensorHandle* thandle = EagerTensor_Handle(eager_tensor_pyobject_ptr);
```

However, there is nothing stopping users from passing in a Python object instead of a tensor.

```
In [2]: tf.experimental.dlpack.to_dlpack([2])
==1720623==WARNING: MemorySanitizer: use-of-uninitialized-value
    #0 0x55b0ba5c410a in tensorflow::(anonymous namespace)::GetTensorFromHandle(TFE_TensorHandle*, TF_Status*) third_party/tensorflow/c/eager/dlpack.cc:46:7
    #1 0x55b0ba5c38f4 in tensorflow::TFE_HandleToDLPack(TFE_TensorHandle*, TF_Status*) third_party/tensorflow/c/eager/dlpack.cc:252:26
...
```

The uninitialized memory address is due to a `reinterpret_cast`

tensorflow/tensorflow/python/eager/pywrap_tensor.cc
Lines 848 to 850 in 0e68f4d

```
848     TFE_TensorHandle* EagerTensor_Handle(const PyObject* o) {
849       return reinterpret_cast<const EagerTensor*>(o)->handle;
850     }
```

Since the `PyObject` is a Python object, not a TensorFlow Tensor, the cast to `EagerTensor` fails.

Patches

We have patched the issue in `22e07fb` and will release a patch release for all affected versions.

We recommend users to upgrade to TensorFlow 2.2.1 or 2.3.1.

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

High

CVE ID

CVE-2020-15193

Weaknesses

No CWEs