Undefined behavior in dlpack

Low **mihaimaruseac** published **GHSA-q8qj-fc9q-cphr** 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

If a user passes an invalid argument to `dlpack.to_dlpack` the expected validations will cause variables to bind to `nullptr` while setting a `status` variable to the error condition.

However, this `status` argument is not properly checked:

tensorflow/tensorflow/c/eager/dlpack.cc
Lines 265 to 267 in `0e68f4d`

```
265     dlm_tensor->dl_tensor.data = TFE_TensorHandleDevicePointer(h, status);
266     dlm_tensor->dl_tensor.dtype = GetDlDataType(data_type, status);
267
```

Hence, code following these methods will bind references to null pointers:

tensorflow/tensorflow/c/eager/dlpack.cc
Lines 279 to 285 in `0e68f4d`

```
279     dlm_tensor->dl_tensor.shape = &(*shape_arr)[0];
280     // There are two ways to represent compact row-major data
281     // 1) nullptr indicates tensor is compact and row-majored.
282     // 2) fill in the strides array as the real case for compact row-major data.
283     // Here we choose option 2, since some frameworks didn't handle the strides
284     // argument properly.
285     dlm_tensor->dl_tensor.strides = &(*stride_arr)[0];
```

This is undefined behavior and reported as an error if compiling with `-fsanitize=null` .

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 discovered during variant analysis of GHSA-rjjg-hgv6-h69v.

Severity

Low

CVE ID

CVE-2020-15191

Weaknesses

No CWEs