Format-string vulnerability in TensorFlow's `as_string`

High **mihaimaruseac** published **GHSA-xmq7-7fxm-rr79** on Sep 24, 2020

Package

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

Affected versions	Patched versions
< 2.3.0	1.15.4, 2.0.3, 2.1.2, 2.2.1, 2.3.1

Description

Impact

By controlling the `fill` argument of `tf.strings.as_string`, a malicious attacker is able to trigger a format string vulnerability due to the way the internal format use in a `printf` call is constructed:

tensorflow/tensorflow/core/kernels/as_string_op.cc
Lines 68 to 74 in `0e68f4d`

```
68      format_ = "%";
69      if (width > -1) {
70        strings::Appendf(&format_, "%s%d", fill_string.c_str(), width);
71      }
72      if (precision > -1) {
73        strings::Appendf(&format_, ".%d", precision);
74      }
```

This can result in unexpected output:

```
In [1]: tf.strings.as_string(input=[1234], width=6, fill='-')
Out[1]: <tf.Tensor: shape=(1,), dtype=string, numpy=array(['1234  '], dtype=object)>
In [2]: tf.strings.as_string(input=[1234], width=6, fill='+')
Out[2]: <tf.Tensor: shape=(1,), dtype=string, numpy=array([' +1234'], dtype=object)>
In [3]: tf.strings.as_string(input=[1234], width=6, fill="h")
Out[3]: <tf.Tensor: shape=(1,), dtype=string, numpy=array(['%6d'], dtype=object)>
In [4]: tf.strings.as_string(input=[1234], width=6, fill="d")
Out[4]: <tf.Tensor: shape=(1,), dtype=string, numpy=array(['12346d'], dtype=object)>
In [5]: tf.strings.as_string(input=[1234], width=6, fill="o")
Out[5]: <tf.Tensor: shape=(1,), dtype=string, numpy=array(['23226d'], dtype=object)>
In [6]: tf.strings.as_string(input=[1234], width=6, fill="x")
Out[6]: <tf.Tensor: shape=(1,), dtype=string, numpy=array(['4d26d'], dtype=object)>
In [7]: tf.strings.as_string(input=[1234], width=6, fill="g")
Out[7]: <tf.Tensor: shape=(1,), dtype=string, numpy=array(['8.67458e-3116d'], dtype=object)>
In [8]: tf.strings.as_string(input=[1234], width=6, fill="a")
Out[8]: <tf.Tensor: shape=(1,), dtype=string, numpy=array(['0x0.00ff7eebb4d4p-10226d'], dtype=object)>
In [9]: tf.strings.as_string(input=[1234], width=6, fill="c")
Out[9]: <tf.Tensor: shape=(1,), dtype=string, numpy=array(['\xd26d'], dtype=object)>
In [10]: tf.strings.as_string(input=[1234], width=6, fill="p")
Out[10]: <tf.Tensor: shape=(1,), dtype=string, numpy=array(['0x4d26d'], dtype=object)>
In [11]: tf.strings.as_string(input=[1234], width=6, fill='m')
Out[11]: <tf.Tensor: shape=(1,), dtype=string, numpy=array(['Success6d'], dtype=object)>
```

However, passing in `n` or `s` results in segmentation fault.

Patches

We have patched the issue in `33be22c` and will release patch releases for all versions between 1.15 and 2.3.

We recommend users to upgrade to TensorFlow 1.15.4, 2.0.3, 2.1.2, 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-2020-15203

Weaknesses

No CWEs