Widely used vulnerable APIs

7 widely used vulnerable APIs are Introduced here.

tf.constant. This API function is employed to generate a constant tensor from a tensor-like object. It accepts parameters such as *value* and *dtype*, where 'value' typically consists of a number or a list of numbers, and *dtype* indicates the data type. However, developers may encounter a segmentation error if they assign a string to the *value* parameter and specify *tf.float16* as the *dtype* (CVE-2020-5215). This vulnerability, once exploited, could result in a denial-of-service scenario during model inference or training. To mitigate this risk, developers utilizing TensorFlow versions affected by this vulnerability should implement validation checks for the *value* parameter type, thereby thwarting potential attacks involving malicious string data.

tf.concat. The function of this API is to concatenate tensors along a specified dimension. It accepts *values*, typically a list of two tensors, and *axis*, which dictates the concatenation axis. This API might be susceptible to CVE-2022-23580. The vulnerability arises in the *shape_inference.cc* file of *core/framework* due to the lack of limitation on the dimension size of the input tensor, leading to a memory leak that triggers system interrupts. This vulnerability could impact certain Python APIs that involve tensor operations in TensorFlow, such as *tf.concat*. While it is uncommon to encounter tensors exceeding predefined limits in practice, developers using the relevant TensorFlow versions should validate tensor dimensions to mitigate potential exploitation by malicious actors.

tf.convert_to_tensor. This function converts Python objects of various types to Tensor objects. It accepts Tensor objects, numpy arrays, Python lists, and Python scalars. However, in certain versions of TensorFlow, users may encounter a null pointer error when interacting with this API. CVE-2021-29513 manifests in the *ndarray_tensor.cc* file within *python/lib*, resulting in null-pointer dereferencing when non-numeric tensors are supplied instead of numeric ones. This vulnerability can be triggered by several Python APIs, including *tf.convert_to_tensor*, *tf.range*, *tf.one_hot*, and others. Users of affected TensorFlow versions are strongly advised to verify the data type of tensors before executing tensor operations.

tf.reshape.This API accepts a *tensor* and a *shape* parameter, where *shape* typically comprises either a single number or a list of values. Leveraging the provided *shape* value, this API enables the transformation of the *tensor* into a new tensor object with the specified shape. Similar to *tf.concat*, this API has the potential to trigger CVE-2022-23580, leading to an Abort error. Moreover, it is also affected by CVE-2022-35934. This latter vulnerability, CVE-2022-35934, is located in the *reshape_op.h* file within the *core/kernels* module. As the maximum dimensionality of *TensorShape* is not constrained, users can inadvertently trigger overflow errors, resulting in denial of service. Users employing relevant versions of TensorFlow should validate that the *shape* parameter is restricted to its maximum length before employing *tf.reshape*.

tf.gather. This API accepts *params*, typically a tensor object or Python list, and *indices*, which can be a number or an integer tensor. It can gather slices from *params* axis according to *indices*. This API could be impacted by CVE-2021-37687, stemming from the absence of proper validation for negative values within the *indices* parameter. Exploiting this vulnerability, attackers can potentially access expected data from the heap by setting negative values in *indices*, resulting in a data leak.

tf.range. This API functions similarly to Python's built-in *range()* function, generating sequences of numbers. However, unlike *range()*, *tf.range* returns a tensor object instead of a Python list. It is worth noting that this API is affected by multiple TensorFlow vulnerabilities. Firstly, similar to *tf.convert_to_tensor*, *tf.range* can also trigger CVE-2021-29513, leading to a null pointer exception. Alternatively, it may trigger CVE-2021-41202, found in the *sequence_ops.cc* file within the *core/kernels* module. This can result in crashes if the *start* or *end* parameter values of *tf.range* are excessively large. Additionally, CVE-2022-36015 manifests in the *RangeSize* function of the *core/ops/math_ops.cc* file, causing a crash when accepting a value unsuitable for *int64_t*. This vulnerability may also affect *tf.range*.

tf.transpose. This API is utilized for transposing or matrix transforming tensor objects. It accepts a tensor object along with arguments such as *perm*, which can either be a *None* value or an integer list. Depending on the value of *perm*, this API rearranges the dimensions accordingly. When *perm* is empty, the operation conducts a transpose of a standard two-dimensional matrix. Additionally, this API features the parameter *conjugate*. Setting *conjugate* to *true* results in the conjugation and transposition of tensor values. However, if a user supplies a complex tensor to this API and sets *conjugate* to true, it may trigger CVE-2021-29618, leading to a crash.

Vulnerable APIs	CVEs	Affected TensorFlow Versions
tf.constant	CVE-2020-5215	2.1.0, 2.0.0, <=1.15.0
tf.concat	CVE-2022-23580	>=0.10.0, <=2.6.2, 2.7.0, !=2.5.3
tf.convert_to_tensor	CVE-2021-29513	>=0.6.0, <=2.4.1, !=2.1.4, !=2.2.3, !=2.3.3, !=2.4.2
tf.reshape	CVE-2022-23580	>=0.10.0, <=2.6.2, 2.7.0, !=2.5.3
	CVE-2022-35934	>=0.6.0, <=2.7.1, 2.8.0, 2.9.0
tf.gather	CVE-2021-37687	>=1.13.1, <=2.4.2, 2.5.0, !=2.3.4
tf.range	CVE-2022-36015	1.15.0, >=2.0.0, <=2.7.1, 2.8.0, 2.9.0
	CVE-2022-36013	>=2.0.0, <=2.7.1, 2.8.0, 2.9.0
	CVE-2021-41202	>=0.11.0, <=2.4.3, 2.5.0, 2.5.1, 2.6.0
	CVE-2021-29513	>=0.6.0, <=2.4.1, !=2.1.4, !=2.2.3, !=2.3.3, !=2.4.2
tf.transpose	CVE-2021-29618	>=1.4.0, <=2.3.2, 2.4.0, 2.4.1

Table 1. Widely-used APIs and their corresponding vulnerabilities, and the impacted TensorFlow versions are highlighted. Notably, the symbol '!=' denotes this version is not affected.