

# CVSS Calculation Typst Library Documentation

The CVSS Typst Library is a Typst package designed to facilitate the calculation of Common Vulnerability Scoring System (CVSS) scores for vulnerabilities across multiple versions, including CVSS 2.0, 3.0, 3.1, and 4.0. This library provides developers, security analysts, and researchers with a reliable and efficient toolset for assessing the severity of security vulnerabilities based on the CVSS standards.

## Functions

### verify

This function verifies if the input string is a valid CVSS vector string.

Example:

```
cvss.verify("CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H")
```

Output:

```
true // or false
```

### Parameters

**verify**(vec: string) -> bool: true if the input string is a valid CVSS vector string  
false otherwise.

**vec** string

the CVSS vector string.

### metrics

This function calculates breaks down the CVSS vector string into a dictionary.

Example:

```
cvss.metrics("CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H")
```

Output:

```
(  
  "version": "3.1",  
  "AV": "N",  
  "AC": "L",  
  "PR": "N",  
  "UI": "N",  
  "S": "U",  
  "C": "N",  
  "I": "N",  
  "A": "H"  
)
```

### Parameters

**metrics**(s: string) -> dictionary: the CVSS vector string as a dictionary.

**s** `string`

the CVSS vector string.

### **dict2str**

This function changes the CVSS vector dictionary into a string.

Example:

```
cvss.dict2str((
    "version": "3.1",
    "AV": "N",
    "AC": "L",
    "PR": "N",
    "UI": "N",
    "S": "U",
    "C": "N",
    "I": "N",
    "A": "H"
))
```

Output:

"CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H"

### **Parameters**

**dict2str**(d: `dictionary`) -> `string`: the CVSS vector dictionary as a string.

**d** `dictionary`

the CVSS vector dictionary.

### **str2dict**

This function changes the CVSS vector string into a dictionary.

Example:

```
cvss.str2dict("CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H")
```

Output:

```
(
    "version": "3.1",
    "AV": "N",
    "AC": "L",
    "PR": "N",
    "UI": "N",
    "S": "U",
    "C": "N",
    "I": "N",
    "A": "H"
)
```

```
"A": "H"  
)
```

### Parameters

**str2dict**(s: string) -> dictionary: the CVSS vector string as a dictionary.

**s** string

the CVSS vector string.

### score

This function calculates the CVSS severity based on the CVSS vector string or dictionary.

Example:

```
cvss.score("CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H")
```

Output:

7.5

- vec (string|dictionary): the CVSS vector string.

### Parameters

**score**(vec) -> float: the CVSS score.

### parse

This function calculates the CVSS severity, score and metrics based on the CVSS vector string or dictionary.

Example:

```
cvss.parse("CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H")
```

Output:

```
(  
  metrics: (  
    "version": "3.1",  
    "AV": "N",  
    "AC": "L",  
    "PR": "N",  
    "UI": "N",  
    "S": "U",  
    "C": "N",  
    "I": "N",  
    "A": "H"  
  ),  
  score: 7.5,  
  severity: "High"  
)
```

- `vec` (string|dictionary): the CVSS vector string.
- `dictionary`: the CVSS severity, score and metrics.

### Parameters

`parse(vec)`

### NONE

Pseudo-constant representing the None severity.

string: "None"

### LOW

Pseudo-constant representing the Low severity.

string: "Low"

### MEDIUM

Pseudo-constant representing the Medium severity.

string: "Medium"

### HIGH

Pseudo-constant representing the High severity.

string: "High"

### CRITICAL

Pseudo-constant representing the Critical severity.

string: "Critical"

**VERSIONS** `array`: the accepted CVSS versions. ("4.0"

This array contains strings of the the accepted CVSS versions.

**re** `regex: the regex pattern.`

Regex pattern for the parsing the CVSS vector string.

**does not handle invalid metrics or invalid values** E.g. "CVSS:3.1/ZZ:QQ/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H"