Incorrect handling of invalid surrogate pair characters

Moderate **hugovk** published **GHSA-wpqr-jcpx-745r** on Jul 2

Package

ujson (pip)

Affected versions	Patched versions
< 5.4.0	5.4.0

Description

Impact

What kind of vulnerability is it? Who is impacted?

Anyone parsing JSON from an untrusted source is vulnerable.

JSON strings that contain escaped surrogate characters not part of a proper surrogate pair were decoded incorrectly. Besides corrupting strings, this allowed for potential key confusion and value overwriting in dictionaries.

Examples:

```
# An unpaired high surrogate character is ignored.
>>> ujson.loads(r'"\uD800"')
''
>>> ujson.loads(r'"\uD800hello"')
'hello'

# An unpaired low surrogate character is preserved.
>>> ujson.loads(r'"\uDC00"')
'\udc00'

# A pair of surrogates with additional non surrogate characters pair up in spite of being invali
>>> ujson.loads(r'"\uD800foo bar\uDC00"')
'foo barᵸ'
```

Patches

Has the problem been patched? What versions should users upgrade to?

Users should upgrade to UltraJSON 5.4.0.

From version 5.4.0, UltraJSON decodes lone surrogates in the same way as the standard library's `json` module does, preserving them in the parsed output:

```
>>> ujson.loads(r'"\uD800"')
'\ud800'
>>> ujson.loads(r'"\uD800hello"')
'\ud800hello'
>>> ujson.loads(r'"\uDC00"')
'\udc00'
>>> ujson.loads(r'"\uD800foo bar\uDC00"')
'\ud800foo bar\udc00'
```

Workarounds

Is there a way for users to fix or remediate the vulnerability without upgrading?

Short of switching to an entirely different JSON library, there are no safe alternatives to upgrading.

For more information

If you have any questions or comments about this advisory:

- Open an issue in UltraJSON

Severity

Moderate

CVE ID

CVE-2022-31116

Weaknesses

CWE-228

Credits

JustAnotherArchivist

the-bumble