# Resource exhaustion via specifically crafted JWE

Moderate **panva** published **GHSA-jv3g-j58f-9mq9** on Sep 6

Package

| Package | Affected versions | Patched versions |
|---|---|---|
| **jose** (npm) | 1 <=1.28.1 \|\| 2 <=2.0.5 \|\| 3 <=3.20.3 \|\| 4 <=4.9.1 | ^1.28.2 \|\| ^2.0.6 \|\| ^3.20.4 \|\| >=4.9.2 |
| **jose-browser-runtime** (npm) | 3 <=3.20.3 \|\| 4 <=4.9.1 | ^3.20.4 \|\| >=4.9.2 |
| **jose-node-cjs-runtime** (npm) | 3 <=3.20.3 \|\| 4 <=4.9.1 | ^3.20.4 \|\| >=4.9.2 |
| **jose-node-esm-runtime** (npm) | 3 <=3.20.3 \|\| 4 <=4.9.1 | ^3.20.4 \|\| >=4.9.2 |

Description

The PBKDF2-based JWE key management algorithms expect a JOSE Header Parameter named `p2c` (PBES2 Count), which determines how many PBKDF2 iterations must be executed in order to derive a CEK wrapping key. The purpose of this parameter is to intentionally slow down the key derivation function in order to make password brute-force and dictionary attacks more expensive.

This makes the PBES2 algorithms unsuitable for situations where the JWE is coming from an untrusted source: an adversary can intentionally pick an extremely high PBES2 Count value, that will initiate a CPU-bound computation that may take an unreasonable amount of time to finish.

## Impact

Under certain conditions (see below) it is possible to have the user's environment consume unreasonable amount of CPU time.

## Affected users

The impact is limited only to users utilizing the JWE decryption APIs with symmetric secrets to decrypt JWEs from untrusted parties who do not limit the accepted JWE Key Management Algorithms ( `alg` Header Parameter) using the `keyManagementAlgorithms` (or `algorithms` in v1.x) decryption option or through other means.

The PBKDF2-based JWE Key Management Algorithm Identifiers are

- `PBES2-HS256+A128KW`
- `PBES2-HS384+A192KW`
- `PBES2-HS512+A256KW`

e.g.

```
const secret = new Uint8Array(16)
const jwe = '...' // JWE from an untrusted party

await jose.compactDecrypt(jwe, secret)
```

You are NOT affected if any of the following applies to you

- Your code does not use the JWE APIs
- Your code only produces JWE tokens
- Your code only decrypts JWEs using an asymmetric JWE Key Management Algorithm (this means you're providing an asymmetric key object to the JWE decryption API)
- Your code only accepts JWEs produced by trusted sources
- Your code limits the accepted JWE Key Management Algorithms using the `keyManagementAlgorithms` decryption option not including any of the PBKDF2-based JWE key management algorithms

## Patches

`v1.28.2` , `v2.0.6` , `v3.20.4` , and `v4.9.2` releases limit the maximum PBKDF2 iteration count to `10000` by default. It is possible to adjust this limit with a newly introduced `maxPBES2Count` decryption option.

## Workarounds

All users should be able to upgrade given all stable semver major release lines have had new a patch release introduced which limits the PBKDF2 iteration count to `10000` by default. This removes the ability to craft JWEs that would consume unreasonable amount of CPU time.

If users are unable to upgrade their required library version they have two options depending on whether they expect to receive JWEs using any of the three PBKDF2-based JWE key management algorithms.

- they can use the `keyManagementAlgorithms` decryption option to disable accepting PBKDF2 altogether
- they can inspect the JOSE Header prior to using the decryption API and limit the PBKDF2 iteration count ( `p2c` Header Parameter)

### For more information

If you have any questions or comments about this advisory:

- Open an discussion in the project's repository
- Email me at panva.ip@gmail.com

**Severity**

Moderate 5.3 / 10

**CVSS base metrics**

| | |
|---|---|
| Attack vector | Network |
| Attack complexity | Low |
| Privileges required | None |
| User interaction | None |
| Scope | Unchanged |
| Confidentiality | None |
| Integrity | None |
| Availability | Low |

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

**CVE ID**

CVE-2022-36083

**Weaknesses**

No CWEs

**Credits**

TomTervoort

panva