ShwethaKalyanaraman/atom:package.json

Snapshot taken 2 days ago.

Retest now

Vulnerabilities	11 via 36 paths
Dependencies	724
Source	GitHub
Taken by	Recurring
Tested with	package-lock.json,package.json
Repository	atom
Branch	master
Manifest	package.json

LOW SEVERITY

Prototype Pollution

Vulnerable module: lodash

Introduced through: settings-view@https://www.atom.io/api/packages/settings-view/versions/0.256.1/tarball, markdown-preview@https://www.atom.io/api/packages/markdown-preview/versions/0.159.25/tarball and others

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom@1.33.0-dev › settings-view@https://www.atom.io/api/packages/settings-view/versions/0.256.1/tarball › roaster@1.2.1 › task-lists@0.2.0 › cheerio@0.15.0 › lodash@2.4.2

Remediation: No remediation path available.

Introduced through: atom@1.33.0-dev › markdown-preview@https://www.atom.io/api/packages/markdown-preview/versions/0.159.25/tarball › roaster@1.2.1 › task-lists@0.2.0 › cheerio@0.15.0 › lodash@2.4.2

Remediation: No remediation path available.

Introduced through: atom@1.33.0-dev › settings-view@https://www.atom.io/api/packages/settings-view/versions/0.256.1/tarball › roaster@1.2.1 › cheerio@0.15.0 › lodash@2.4.2

Remediation: No remediation path available.

…and 3 more

Overview

lodash is a javaScript utility library delivering modularity, performance & extras.

Affected versions of this package are vulnerable to Prototype Pollution. The utilities function allow modification of the `Object` prototype. If an attacker can control part of the structure passed to this function, they could add or modify an existing property.

PoC by Olivier Arteau (HoLyVieR)

```
var _= require('lodash');
var malicious_payload = '{"__proto__":{"oops":"It works !"}}' ;

var a = {};
console.log("Before : " + a.oops);
_.merge({}, JSON.parse(malicious_payload));
console.log("After : " + a.oops);
```

HIGH SEVERITY

Arbitrary Code Injection

Vulnerable module: growl
Introduced through: text-buffer@13.14.9, fs-admin@0.1.7 and others

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom@1.33.0-dev › text-buffer@13.14.9 › fs-admin@0.1.7 › mocha@3.5.3 › growl@1.9.2

Remediation: No remediation path available.

Introduced through: atom@1.33.0-dev › fs-admin@0.1.7 › mocha@3.5.3 › growl@1.9.2

Remediation: No remediation path available.

Introduced through: atom@1.33.0-dev › mocha@2.5.1 › growl@1.9.2

Remediation: Upgrade to mocha@4.0.0.

Overview

`growl` is a package adding Growl support for Nodejs.

Affected versions of the package are vulnerable to Arbitrary Code Injection due to unsafe use of the `eval()` function. Node.js provides the `eval()` function by default, and is used to translate strings into Javascript code. An attacker can craft a malicious payload to inject arbitrary commands.

MEDIUM SEVERITY

Cross-site Scripting (XSS)

Vulnerable module: react-dom
Introduced through: github@https://www.atom.io/api/packages/github/versions/0.20.0/tarball

Detailed paths and remediation

Introduced through: atom@1.33.0-dev › github@https://www.atom.io/api/packages/github/versions/0.20.0/tarball › react-dom@16.4.0

Remediation: No remediation path available.

Overview

react-dom serves as the entry point of the DOM-related rendering paths.

Affected versions of this package are vulnerable to Cross-site Scripting (XSS) attacks.

This attack is possible only only if the following two conditions are true:

The application is being rendered to HTML using `ReactDOMServer` API, and

The app includes a user-supplied attribute name in an HTML tag.

Server-rendered React apps, which contain the following pattern may by affected:

```
let props = {};
props[userProvidedData] = "hello";
let element = <div {...props} />;
let html = ReactDOMServer.renderToString(element);
```

Given the attacker could influence the `userProvidedData` , they could craft an attribute name that would triggler an XSS vulnerability, like:

```
></div><script>alert("hi")</script>`
```

Allowing the attacker to inject arbitrary markup:

```
<div ></div><script>alert("hi")</script>
```

Note: This vulnerability affects only some server-rendered React apps, which contain a the pattern. Purely client-rendered apps are not affected.

MEDIUM SEVERITY

Insecure Randomness

Vulnerable module: cryptiles
Introduced through: less-cache@1.1.0

Detailed paths and remediation

Introduced through: atom@1.33.0-dev › less-cache@1.1.0 › less@2.7.3 › request@2.81.0 › hawk@3.1.3 › cryptiles@2.0.5

Remediation: No remediation path available.

Overview

cryptiles is a package for general crypto utilities.

Affected versions of this package are vulnerable to Insecure Randomness. The `randomDigits()` method is supposed to return a cryptographically strong pseudo-random data string, but it was biased to certain digits. An attacker could be able to guess the created digits.

MEDIUM SEVERITY

Time of Check Time of Use (TOCTOU)

Vulnerable module: chownr
Introduced through: language-shellscript@https://www.atom.io/api/packages/language-shellscript/versions/0.27.5/tarball,

language-ruby@https://www.atom.io/api/packages/language-ruby/versions/0.72.9/tarball and others

Detailed paths and remediation

Introduced through: atom@1.33.0-dev › language-shellscript@https://www.atom.io/api/packages/language-shellscript/versions/0.27.5/tarball › tree-sitter-bash@0.13.2 › prebuild-install@5.1.0 › tar-fs@1.16.3 › chownr@1.0.1

Remediation: No remediation path available.

Introduced through: atom@1.33.0-dev › language-ruby@https://www.atom.io/api/packages/language-ruby/versions/0.72.9/tarball › tree-sitter-ruby@0.13.10 › prebuild-install@5.1.0 › tar-fs@1.16.3 › chownr@1.0.1

Remediation: No remediation path available.

Introduced through: atom@1.33.0-dev › github@https://www.atom.io/api/packages/github/versions/0.20.0/tarball › keytar@4.2.1 › prebuild-install@2.5.3 › tar-fs@1.16.3 › chownr@1.0.1

Remediation: No remediation path available.

…and 3 more

Overview

Affected versions of chownr are vulnerable to Time of Check Time of Use (TOCTOU). It does not dereference symbolic links and changes the owner of the link.

HIGH SEVERITY

Regular Expression Denial of Service (DoS)

Vulnerable module: minimatch
Introduced through: jasmine-tagged@1.1.4, mocha@2.5.1 and others

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom@1.33.0-dev › jasmine-tagged@1.1.4 › jasmine-focused@1.0.7 › jasmine-node@git+https://github.com/kevinsawicki/jasmine-node.git#81af4f953a2b7dfb5bde8331c05362a4b464c5ef › gaze@0.3.4 › minimatch@0.2.14

Remediation: Run `snyk wizard` to patch minimatch@0.2.14.

Introduced through: atom@1.33.0-dev › jasmine-tagged@1.1.4 › jasmine-focused@1.0.7 › jasmine-node@git+https://github.com/kevinsawicki/jasmine-node.git#81af4f953a2b7dfb5bde8331c05362a4b464c5ef › gaze@0.3.4 › fileset@0.1.8 › minimatch@0.4.0

Remediation: Run `snyk wizard` to patch minimatch@0.4.0.

Introduced through: atom@1.33.0-dev › jasmine-tagged@1.1.4 › jasmine-focused@1.0.7 › jasmine-node@git+https://github.com/kevinsawicki/jasmine-node.git#81af4f953a2b7dfb5bde8331c05362a4b464c5ef › gaze@0.3.4 › fileset@0.1.8 › glob@3.2.11 › minimatch@0.3.0

Remediation: Run `snyk wizard` to patch minimatch@0.3.0.

…and 5 more

Overview

`minimatch` is a minimalistic matching library used for converting glob expressions into JavaScript RegExp objects. Affected versions of this package are vulnerable to Regular Expression Denial of Service (ReDoS) attacks.
The Regular expression Denial of Service (ReDoS) is a type of Denial of Service attack. Many Regular Expression implementations may reach edge cases that causes them to work very slowly (exponentially related to input size), allo an attacker to exploit this and can cause the program to enter these extreme situations by using a specially crafted inpu cause the service to excessively consume CPU, resulting in a Denial of Service.

An attacker can provide a long value to the `minimatch` function, which nearly matches the pattern being matched. This will cause the regular expression matching to take a long time, all the while occupying the event loop and preventing it from processing other requests and making the server unavailable (a Denial of Service attack).

You can read more about `Regular Expression Denial of Service (ReDoS)` on our blog.

LOW SEVERITY

Prototype Pollution

Vulnerable module: hoek
Introduced through: less-cache@1.1.0

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom@1.33.0-dev › less-cache@1.1.0 › less@2.7.3 › request@2.81.0 › hawk@3.1.3 › cryptiles@2.0.5 › boom@2.10.1 › hoek@2.16.3

Remediation: Run `snyk wizard` to patch hoek@2.16.3.

Introduced through: atom@1.33.0-dev › less-cache@1.1.0 › less@2.7.3 › request@2.81.0 › hawk@3.1.3 › sntp@1.0.9 › hoek@2.16.3

Remediation: Run `snyk wizard` to patch hoek@2.16.3.

Introduced through: atom@1.33.0-dev › less-cache@1.1.0 › less@2.7.3 › request@2.81.0 › hawk@3.1.3 › boom@2.10.1 › hoek@2.16.3

Remediation: Run `snyk wizard` to patch hoek@2.16.3.

…and 1 more

Overview

hoek is a Utility methods for the hapi ecosystem.

Affected versions of this package are vulnerable to Prototype Pollution. The utilities function allow modification of the `Object` prototype. If an attacker can control part of the structure passed to this function, they could add or modify an existing property.

PoC by Olivier Arteau (HoLyVieR)

```
var Hoek = require('hoek');
var malicious_payload = '{"__proto__":{"oops":"It works !"}}' ;

var a = {};
console.log("Before : " + a.oops);
Hoek.merge({}, JSON.parse(malicious_payload));
console.log("After : " + a.oops);
```

LOW SEVERITY

Prototype Pollution

Vulnerable module: extend
Introduced through: less-cache@1.1.0

Detailed paths and remediation

Introduced through: atom@1.33.0-dev › less-cache@1.1.0 › less@2.7.3 › request@2.81.0 › extend@3.0.1

Remediation: Your dependencies are out of date, otherwise you would be using a newer extend than extend@3.0.1. Try reinstalling your dependencies. If the problem persists, one of your dependencies may be bundling outdated modules.

Overview

extend is a port of the classic extend() method from jQuery.

Affected versions of this package are vulnerable to Prototype Pollution. Utilities function can be tricked into modifying the prototype of "Object" when the attacker control part of the structure passed to these function. This can let an attacker add or modify existing property that will exist on all object.

LOW SEVERITY

Regular Expression Denial of Service (ReDoS)

Vulnerable module: ms
Introduced through: mocha@2.5.1

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom@1.33.0-dev › mocha@2.5.1 › debug@2.2.0 › ms@0.7.1

Remediation: Upgrade to mocha@3.5.0.

Overview

`ms` is a tiny millisecond conversion utility.

Affected versions of this package are vulnerable to Regular Expression Denial of Service (ReDoS) due to an incomplete fix for previously reported vulnerability npm:ms:20151024. The fix limited the length of accepted input string to 10,000 characters, and turned to be insufficient making it possible to block the event loop for 0.3 seconds (on a typical laptop) with a specially crafted string passed to `ms()` function.

Proof of concept

```
ms = require('ms');
ms('1'.repeat(9998) + 'Q') // Takes about ~0.3s
```

Note: Snyk's patch for this vulnerability limits input length to 100 characters. This new limit was deemed to be a breaking change by the author. Based on user feedback, we believe the risk of breakage is *very* low, while the value to your security is much greater, and therefore opted to still capture this change in a patch for earlier versions as well. Whenever patching security issues, we always suggest to run tests on your code to validate that nothing has been broken.

For more information on `Regular Expression Denial of Service (ReDoS)` attacks, go to our blog.

LOW SEVERITY

Regular Expression Denial of Service (ReDoS)

Vulnerable module: diff
Introduced through: text-buffer@13.14.9 and fs-admin@0.1.7

Detailed paths and remediation

Introduced through: atom@1.33.0-dev › text-buffer@13.14.9 › fs-admin@0.1.7 › mocha@3.5.3 › diff@3.2.0

Remediation: No remediation path available.

Introduced through: atom@1.33.0-dev › fs-admin@0.1.7 › mocha@3.5.3 › diff@3.2.0

Remediation: No remediation path available.

Overview

`diff` is a javascript text differencing implementation.

Affected versions of this package are vulnerable to Regular Expression Denial of Service (ReDoS) attacks. This can cause an impact of about 10 seconds matching time for data 48K characters long.

LOW SEVERITY

Regular Expression Denial of Service (ReDoS)

Vulnerable module: debug
Introduced through: mocha@2.5.1, text-buffer@13.14.9 and others

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom@1.33.0-dev › mocha@2.5.1 › debug@2.2.0

Remediation: Upgrade to mocha@4.0.0.

Introduced through: atom@1.33.0-dev › text-buffer@13.14.9 › fs-admin@0.1.7 › mocha@3.5.3 › debug@2.6.8

Remediation: Run `snyk wizard` to patch debug@2.6.8.

Introduced through: atom@1.33.0-dev › fs-admin@0.1.7 › mocha@3.5.3 › debug@2.6.8

Remediation: Run `snyk wizard` to patch debug@2.6.8.

Overview

`debug` is a JavaScript debugging utility modelled after Node.js core's debugging technique..

`debug` uses printf-style formatting. Affected versions of this package are vulnerable to Regular expression Denial of Service (ReDoS) attacks via the the `%o` formatter (Pretty-print an Object all on a single line). It used a regular expression (`/\s*\n\s*/g`) in order to strip whitespaces and replace newlines with spaces, in order to join the data into a single line. This can cause a very low impact of about 2 seconds matching time for data 50k characters long.

API Status Vulnerability DB Blog Documentation