ShwethaKalyanaraman/atom:script/package.json

Snapshot taken 17 hours ago.

Retest now

Vulnerabilities	24 via 96 paths
Dependencies	1122
Source	GitHub
Taken by	Recurring
Tested with	package-lock.json,package.json
Repository	atom
Branch	master
Manifest	script/package.json

MEDIUM SEVERITY

Uninitialized Memory Exposure

Vulnerable module: tunnel-agent
Introduced through: webdriverio@2.4.5

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › request@2.34.0 › tunnel-agent@0.3.0

Remediation: Upgrade to webdriverio@4.7.0.

Overview

`tunnel-agent` is HTTP proxy tunneling agent. Affected versions of the package are vulnerable to Uninitialized Memory Exposure.

A possible memory disclosure vulnerability exists when a value of type `number` is used to set the *proxy.auth* option of a request `request` and results in a possible uninitialized memory exposures in the request body.

This is a result of unobstructed use of the `Buffer` constructor, whose insecure default constructor increases the odds of memory leakage.

HIGH SEVERITY

Arbitrary Code Injection

Arbitrary Code Injection

Vulnerable module: growl
Introduced through: fs-admin@0.1.6

Detailed paths and remediation

Introduced through: atom-build-scripts@* › fs-admin@0.1.6 › mocha@3.5.3 › growl@1.9.2

Remediation: No remediation path available.

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.

HIGH SEVERITY

Denial of Service (Memory Exhaustion)

Vulnerable module: qs
Introduced through: webdriverio@2.4.5

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › request@2.34.0 › qs@0.6.6

Remediation: Upgrade to webdriverio@3.0.0.

Overview

`qs` is a querystring parser that supports nesting and arrays, with a depth limit.

Affected versions of this package are vulnerable to Denial of Service (Dos) attacks. During parsing, the `qs` module may create a sparse area (an array where no elements are filled), and grow that array to the necessary size based on the indices used on it. An attacker can specify a high index value in a query string, thus making the server allocate a respectively big array. Truly large values can cause the server to run out of memory and cause it to crash - thus enabling a Denial-of-Service attack.

HIGH SEVERITY

Prototype Override Protection Bypass

Vulnerable module: qs
Introduced through: webdriverio@2.4.5

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › request@2.34.0 › qs@0.6.6

Remediation: Upgrade to webdriverio@4.2.0.

Overview

`qs` is a querystring parser that supports nesting and arrays, with a depth limit.

By default `qs` protects against attacks that attempt to overwrite an object's existing prototype properties, such as `toString()`, `hasOwnProperty()` ,etc.

From `qs` documentation:

> *By default parameters that would overwrite properties on the object prototype are ignored, if you wish to keep the data from those fields either use plainObjects as mentioned above, or set allowPrototypes to true which will allow user input to overwrite those properties. WARNING It is generally a bad idea to enable this option as it can cause problems when attempting to use the properties that have been overwritten. Always be careful with this option.*

Overwriting these properties can impact application logic, potentially allowing attackers to work around security controls, modify data, make the application unstable and more.

In versions of the package affected by this vulnerability, it is possible to circumvent this protection and overwrite prototype properties and functions by prefixing the name of the parameter with `[` or `]`. e.g. `qs.parse("]=toString")` will return `{toString = true}`, as a result, calling `toString()` on the object will throw an exception.

Example:

```
qs.parse('toString=foo', { allowPrototypes: false })
// {}

qs.parse("]=toString", { allowPrototypes: false })
// {toString = true} <== prototype overwritten
```

For more information, you can check out our blog.

HIGH SEVERITY

Regular Expression Denial of Service (DoS)

Vulnerable module: minimatch
Introduced through: webdriverio@2.4.5, standard@8.4.0 and others

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › archiver@0.6.1 › file-utils@0.1.5 › findup-sync@0.1.3 › glob@3.2.11 › minimatch@0.3.0

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

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › archiver@0.6.1 › file-utils@0.1.5 › minimatch@0.2.14

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

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › archiver@0.6.1 › file-utils@0.1.5 › glob@3.2.11 › minimatch@0.3.0

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

…and 9 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 implementat may reach edge cases that causes them to work very slowly (exponentially related to input size), allowing an attacker to expl this and can cause the program to enter these extreme situations by using a specially crafted input and 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.

HIGH SEVERITY

Regular Expression Denial of Service (ReDoS)

Vulnerable module: underscore.string
Introduced through: donna@1.0.16

Detailed paths and remediation

Introduced through: atom-build-scripts@* › donna@1.0.16 › underscore.string@3.3.4

Remediation: Your dependencies are out of date, otherwise you would be using a newer underscore.string than underscore.string@3.3.4. Try reinstalling your dependencies. If the problem persists, one of your dependencies may be bundling outdated modules.

Overview

underscore.string is a String manipulation helpers for javascript.

Affected versions of this package are vulnerable to Regular Expression Denial of Service (ReDoS). It parses dates using regex strings, which may cause a slowdown of 2 seconds per 50k characters.

MEDIUM SEVERITY

Denial of Service (DoS)

Vulnerable module: mem
Introduced through: npm@6.2.0

Detailed paths and remediation

Introduced through: atom-build-scripts@* › npm@6.2.0 › libnpx@10.2.0 › yargs@11.0.0 › os-locale@2.1.0 › mem@1.1.0

Remediation: No remediation path available.

Overview

mem is an optimization technique used to speed up consecutive function calls by caching the result of calls with identical input.

Affected versions of this package are vulnerable to Denial of Service (DoS) attacks. Old results are not deleted from the cache and could cause a memory leak.

MEDIUM SEVERITY

Denial of Service (Event Loop Blocking)

Vulnerable module: qs
Introduced through: webdriverio@2.4.5

Fix this vulnerability

Detailed paths and remediation

Detailed paths and remediation

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › request@2.34.0 › qs@0.6.6

Remediation: Upgrade to webdriverio@3.0.0.

Overview

qs is a querystring parser that supports nesting and arrays, with a depth limit.

Affected versions of this package are vulnerable to Denial of Service (DoS). When parsing a string representing a deeply nested object, qs will block the event loop for long periods of time. Such a delay may hold up the server's resources, keeping it from processing other requests in the meantime, thus enabling a Denial-of-Service attack.

MEDIUM SEVERITY

Insecure Randomness

Vulnerable module: cryptiles
Introduced through: webdriverio@2.4.5 and npm@6.2.0

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › request@2.34.0 › hawk@1.0.0 › cryptiles@0.2.2

Remediation: Upgrade to webdriverio@4.13.0.

Introduced through: atom-build-scripts@* › npm@6.2.0 › libcipm@2.0.0 › npm-lifecycle@2.0.3 › node-gyp@3.7.0 › request@2.81.0 › hawk@3.1.3 › cryptiles@2.0.5

Remediation: Your dependencies are out of date, otherwise you would be using a newer cryptiles than cryptiles@2.0.5. Try reinstalling your dependencies. If the problem persists, one of your dependencies may be bundling outdated modules.

Introduced through: atom-build-scripts@* › npm@6.2.0 › npm-lifecycle@2.0.3 › node-gyp@3.7.0 › request@2.81.0 › hawk@3.1.3 › cryptiles@2.0.5

Remediation: Your dependencies are out of date, otherwise you would be using a newer cryptiles than cryptiles@2.0.5. Try reinstalling your dependencies. If the problem persists, one of your dependencies may be bundling outdated modules.

…and 3 more

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

Remote Memory Exposure

Vulnerable module: request
Introduced through: webdriverio@2.4.5

Fix this vulnerability

Detailed paths and remediation

Detailed paths and remediation

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › request@2.34.0

Remediation: Upgrade to webdriverio@4.2.0.

Overview

`request` is a simplified http request client. A potential remote memory exposure vulnerability exists in `request`. If a `request` uses a multipart attachment and the *body type* option is `number` with value X, then X bytes of uninitialized memory will be sent in the body of the request.

Note that while the impact of this vulnerability is high (memory exposure), exploiting it is likely difficult, as the attacker needs to somehow control the body type of the request. One potential exploit scenario is when a request is composed based on JSON input, including the body type, allowing a malicious JSON to trigger the memory leak.

MEDIUM SEVERITY

Time of Check Time of Use (TOCTOU)

Vulnerable module: chownr
Introduced through: npm@6.2.0 and electron-link@0.2.2

Detailed paths and remediation

Introduced through: atom-build-scripts@* › npm@6.2.0 › libnpmhook@4.0.1 › npm-registry-fetch@3.1.1 › make-fetch-happen@4.0.1 › cacache@11.0.2 › chownr@1.0.1

Remediation: No remediation path available.

Introduced through: atom-build-scripts@* › npm@6.2.0 › libcipm@2.0.0 › pacote@8.1.6 › make-fetch-happen@4.0.1 › cacache@11.0.2 › chownr@1.0.1

Remediation: No remediation path available.

Introduced through: atom-build-scripts@* › electron-link@0.2.2 › leveldown@4.0.1 › prebuild-install@4.0.0 › tar-fs@1.16.3 › chownr@1.0.1

Remediation: No remediation path available.

…and 10 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.

MEDIUM SEVERITY

Timing Attack

Vulnerable module: http-signature
Introduced through: webdriverio@2.4.5

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › request@2.34.0 › http-signature@0.10.1

Remediation: Upgrade to webdriverio@4.2.0.

Overview

`http-signature` is a reference implementation of Joyent's HTTP Signature scheme.

Affected versions of the package are vulnerable to Timing Attacks due to time-variable comparison of signatures.

The library implemented a character to character comparison, similar to the built-in string comparison mechanism, `===`, and not a time constant string comparison. As a result, the comparison will fail faster when the first characters in the signature are incorrect. An attacker can use this difference to perform a timing attack, essentially allowing them to guess the signature one character at a time.

You can read more about timing attacks in Node.js on the Snyk blog.

HIGH SEVERITY

Command Injection

Vulnerable module: shelljs
Introduced through: standard@8.4.0

Detailed paths and remediation

Introduced through: atom-build-scripts@* › standard@8.4.0 › eslint@3.7.1 › shelljs@0.6.1

Remediation: No remediation path available.

Overview

`shelljs` is a portable Unix shell commands for Node.js.

Affected version of this package are vulnerable to Command Injection. It is possible to invoke commands from `shell.exec()` from external sources, allowing an attacker to inject arbitrary commands.

LOW SEVERITY

Prototype Pollution

Vulnerable module: lodash
Introduced through: webdriverio@2.4.5, electron-packager@7.3.0 and others

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › archiver@0.6.1 › file-utils@0.1.5 › lodash@2.1.0

Remediation: No remediation path available.

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › archiver@0.6.1 › file-utils@0.1.5 › findup-sync@0.1.3 › lodash@2.4.2

Remediation: No remediation path available.

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › archiver@0.6.1 › lodash@2.4.2

Remediation: Upgrade to webdriverio@4.2.0.

...and 5 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);
```

LOW SEVERITY

Prototype Pollution

Vulnerable module: hoek
Introduced through: webdriverio@2.4.5 and npm@6.2.0

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › request@2.34.0 › hawk@1.0.0 › cryptiles@0.2.2 › boom@0.4.2 › hoek@0.9.1

Remediation: Upgrade to webdriverio@4.9.0.

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › request@2.34.0 › hawk@1.0.0 › sntp@0.2.4 › hoek@0.9.1

Remediation: Upgrade to webdriverio@4.9.0.

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › request@2.34.0 › hawk@1.0.0 › boom@0.4.2 › hoek@0.9.1

Remediation: Upgrade to webdriverio@4.9.0.

…and 21 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

Prototype Pollution

Vulnerable module: extend

Introduced through: npm@6.2.0, electron-packager@7.3.0 and others

Detailed paths and remediation

Introduced through: atom-build-scripts@* › npm@6.2.0 › libcipm@2.0.0 › npm-lifecycle@2.0.3 › node-gyp@3.7.0 › 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.

Introduced through: atom-build-scripts@* › electron-packager@7.3.0 › electron-download@2.2.1 › nugget@1.6.2 › request@2.87.0 › 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.

Introduced through: atom-build-scripts@* › electron-winstaller@2.6.4 › asar@0.11.0 › mksnapshot@0.3.1 › request@2.87.0 › 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.

…and 9 more

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 (DoS)

Vulnerable module: hawk

Introduced through: webdriverio@2.4.5

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › request@2.34.0 › hawk@1.0.0

Remediation: Upgrade to webdriverio@4.2.0.

Overview

hawk is an HTTP authentication scheme using a message authentication code (MAC) algorithm to provide partial HTTP request cryptographic verification.

Affected versions of this package are vulnerable to Regular Expression Denial of Service (ReDoS) attacks.

LOW SEVERITY

Regular Expression Denial of Service (ReDoS)

Vulnerable module: plist

Introduced through: electron-packager@7.3.0

Detailed paths and remediation

Introduced through: atom-build-scripts@* › electron-packager@7.3.0 › plist@1.2.0

Remediation: No remediation path available.

Overview

`plist` is a Mac OS X Plist parser/builder for Node.js and browsers

Affected versions of this package are vulnerable to Regular Expression Denial of Service (ReDoS) attacks due to bundling a vulnerable version of the XMLBuilder package. This can cause an impact of about 10 seconds matching time for data 60 characters long.

LOW SEVERITY

Regular Expression Denial of Service (ReDoS)

Vulnerable module: mime
Introduced through: webdriverio@2.4.5

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › request@2.34.0 › form-data@0.1.4 › mime@1.2.11

Remediation: Upgrade to webdriverio@3.0.0.

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › request@2.34.0 › mime@1.2.11

Remediation: Run `snyk wizard` to patch mime@1.2.11.

Overview

`mime` is a comprehensive, compact MIME type module.

Affected versions of this package are vulnerable to Regular expression Denial of Service (ReDoS). It uses regex the following regex `/.*[\.\/\\]/` in its lookup, which can cause a slowdown of 2 seconds for 50k characters.

The Regular expression Denial of Service (ReDoS) is a type of Denial of Service attack. Many Regular Expression implementations may reach extreme situations that cause them to work very slowly (exponentially related to input size), allowing an attacker to exploit this and can cause the program to enter these extreme situations by using a specially crafted input and cause the service to excessively consume CPU, resulting in a Denial of Service.

LOW SEVERITY

Regular Expression Denial of Service (ReDoS)

Vulnerable module: github-url-to-object
Introduced through: publish-release@1.6.0

Detailed paths and remediation

Introduced through: atom-build-scripts@* › publish-release@1.6.0 › github-url-to-object@1.6.0

Remediation: No remediation path available.

Overview

`github-url-to-object` is a module for node.js and browsers that extracts useful properties like user, repo, and branch from various flavors of GitHub URLs.

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 40K characters long.

LOW SEVERITY

Regular Expression Denial of Service (ReDoS)

Vulnerable module: diff
Introduced through: fs-admin@0.1.6

Detailed paths and remediation

Introduced through: atom-build-scripts@* › fs-admin@0.1.6 › 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: eslint
Introduced through: standard@8.4.0

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom-build-scripts@* › standard@8.4.0 › eslint@3.7.1

Remediation: Upgrade to standard@11.0.0.

Overview

`eslint` is an AST-based pattern checker for JavaScript.

Affected versions of the package are vulnerable to Regular Expression Denial of Service (ReDoS) attacks. This can cause an impact of about 10 seconds matching time for data 100k characters long.

LOW SEVERITY

Regular Expression Denial of Service (ReDoS)

Vulnerable module: debug
Introduced through: webdriverio@2.4.5 and fs-admin@0.1.6

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom-build-scripts@* › webdriverio@2.4.5 › archiver@0.6.1 › zip-stream@0.2.3 › debug@0.7.4

Remediation: No remediation path available.

Introduced through: atom-build-scripts@* › fs-admin@0.1.6 › 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.

LOW SEVERITY

Regular Expression Denial of Service (ReDoS)

Vulnerable module: braces
Introduced through: klaw-sync@1.1.2 and stylelint@9.3.0

Fix this vulnerability

Detailed paths and remediation

Introduced through: atom-build-scripts@* › klaw-sync@1.1.2 › micromatch@2.3.11 › braces@1.8.5

Remediation: Upgrade to klaw-sync@2.0.0.

Introduced through: atom-build-scripts@* › stylelint@9.3.0 › micromatch@2.3.11 › braces@1.8.5

Remediation: Upgrade to stylelint@9.8.0.

Overview

`braces` is a Bash-like brace expansion, implemented in JavaScript.

Affected versions of this package are vulnerable to Regular Expression Denial of Service (ReDoS) attacks. It used a regular expression (`^\{(,+(?:(\{,+\})*),*|,*(?:(\{,+\})*),+)\}`) in order to detects empty braces. This can cause an impact of about 10 seconds matching time for data 50K characters long.

© 2018 Snyk Ltd.