Analyzing Source Code for Bug Bounty

MNSec 2020
Ulzii Otgonbaatar,
Security Engineer, Clever Inc.

master

notes / Lets-fix-Buffer-API.md

Go to file

ChALkeR Lets-fix-Buffer-API: GitHub formatting fixes ...

2 contributors

278 lines (150 sloc) | 25.4 KB

Let's fix Node.js Buffer API.

Update 2016-08: `Buffer.alloc()` `Buffer.from()` `Buffer.allocUnsafe()` and `Bu`

new Buffer(size)

Deprecated since: v6.0.0

Stability: 0 - Deprecated: Use `Buffer.alloc()` inst

- `size` <integer> The desired length of the new

Allocates a new `Buffer` of `size` bytes. The `size` must be less than or equal to the value of `buffer`

Buffer(number) is unsafe #4660

Closed **feross** opened this issue on Jan 12, 2016 · 208 comments

feross commented on Jan 12, 2016

tl;dr

This issue proposes:

1. Change `new Buffer(number)` to return safe, zeroed-out mem
2. Create a new API for creating uninitialized Buffers, `Buffer.al`

Update: Jan 15, 2016

Upon further consideration, I think that returning zeroed out mem
allocation should be in a different API.

I now support adding two APIs:

- `Buffer.from(value)` - convert from any type to a buffer
- `Buffer.alloc(size)` - create an uninitialized buffer with give

```
// create a buffer of length 10
const buf = new Buffer(10);

console.log(buf);

// Prints:
<Buffer ff ff ff 00 00 00 00 3d>
```

Information Disclosure Vulnerability

Potential loss of secrets or PII

Hunting for bugs at scale

1. Write your analysis rule & test on small batches
2. Run at scale
3. Triage findings
4. Report to proper authorities

1. Writing analysis and testing

disallow use of the Buffer() constructor (no-buffer-constructor)

This rule was **deprecated** in ESLint v7.0.0. Please use tl

In Node.js, the behavior of the `Buffer` constructor is user input to `Buffer()` without validating its type ca denial of service. As a result, the `Buffer` constructor `Buffer.from`, `Buffer.alloc`, and `Buffer.alloc`

Rule Details

This rule disallows calling and constructing the `Buffe`

Examples of **incorrect** code for this rule:

```
new Buffer(5);
new Buffer([1, 2, 3]);

Buffer(5);
Buffer([1, 2, 3]);

new Buffer(res.body.amount);
new Buffer(res.body.values);
```

Examples of **correct** code for this rule:

```
Buffer.alloc(5);
Buffer.allocUnsafe(5);
Buffer.from([1, 2, 3]);

Buffer.alloc(res.body.amount);
Buffer.from(res.body.values);
```

```
33      create(context) {
34
35          //----------------------------------------------------------
36          // Public
37          //----------------------------------------------------------
38
39          return {
40              "CallExpression[callee.name='Buffer'], NewExpression[callee.
41                  context.report({
42                      node,
43                      messageId: "deprecated",
44                      data: { expr: node.type === "CallExpression" ? "Buff
45                  });
46              }
47          };
48      }
49  };
```

Semgrep

- Variables to extract out
- Write analysis as you write and think about code
- Post-analysis filtering
- No painful domain-specific language

https://r2c.dev/blog/2020/why-i-moved-to-semgrep-for-all-my-code-analysis/

```
// https://semgrep.dev/s/ulziibay:deprecated-buffer-api-use-detected

rules:
- id: deprecated-buffer-api-usage
 patterns:
 - pattern: |
     new Buffer($SIZE)
 - metavariable-comparison:
     metavariable: '$SIZE'
     comparison: '$SIZE > 0'
 message: |
   NodeJS `new Buffer()` API is deprecated. Please do not use it! See
https://nodejs.org/en/docs/guides/buffer-constructor-deprecation/ for instructions!
 severity: ERROR
```

Variables

```
// create a buffer of length 10
const buf = new Buffer(10);
```

Post analysis filtering

New Load Examples Tools Help

JavaScript ulziibay: Untitled rule save

SEMGREP RULE

Simple Advanced

```yaml
rules:
- id: deprecated-buffer-api-usage
  patterns:
  - pattern: |
      new Buffer($SIZE)
  - metavariable-comparison:
      metavariable: '$SIZE'
      comparison: '$SIZE > 0'
  message: |
    NodeJS `new Buffer()` API is deprecated. Please do not use it! See https://nodejs.org/en/docs/guides/buffer-constructor-deprecation/ for i
```

TEST CODE

```
1  const buf1 = new Buffer([1,2,3]);
2  // creates a buffer containing [01, 02, 03]
3  const buf2 = new Buffer('test');
4  // creates a buffer containing ASCII bytes [74, 65, 73, 74]
5  const buf3 = new Buffer(10);
6  // creates a buffer of length 10
```

```
$ semgrep --config https://semgrep.dev/s/ulziibay:deprecated-buffer-api-use-detected
using config from https://semgrep.dev/s/ulziibay:deprecated-buffer-api-use-detected. Visit https://semgrep.dev/registry to see all public rules.
downloading config...
running 1 rules...
buffer_exploit.js
severity:error rule:deprecated-buffer-api-usage: NodeJS `new Buffer()` API is deprecated. Please do not use it! See
https://nodejs.org/en/docs/guides/buffer-constructor-deprecation/ for instructions!

3:    var buf = (new Buffer(200)).toString('ascii');

example.js
severity:error rule:deprecated-buffer-api-usage: NodeJS `new Buffer()` API is deprecated. Please do not use it! See
https://nodejs.org/en/docs/guides/buffer-constructor-deprecation/ for instructions!

5:const buf3 = new Buffer(10);
ran 1 rules on 2 files: 2 findings
```

2. Running at scale

Run Job

Analyzer

dev/semgrep
0.25.0

Change

Input Set

Hacktoberfest 202 (0.0.1)

Change

Step 3:
Include your parameters (optional)

```
yamlurl: 'https://semgrep.dev/c/ulziibay:deprecated-buffer-api-use-detected'
```

Triager | Job 1245 | Scoring

Filter by repositories, commit hashes, or checks:

filter by repos...

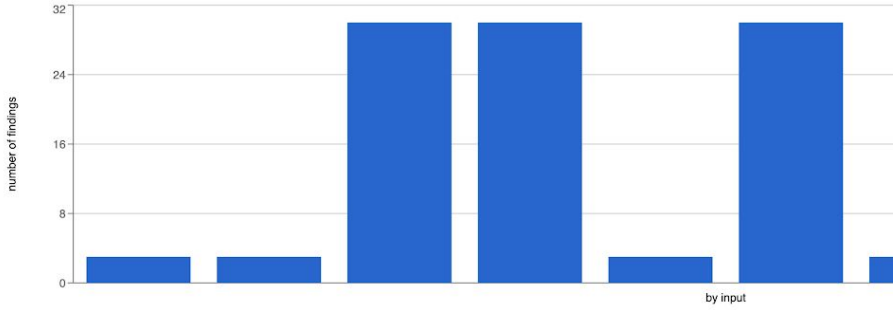
filter by commit hashes...

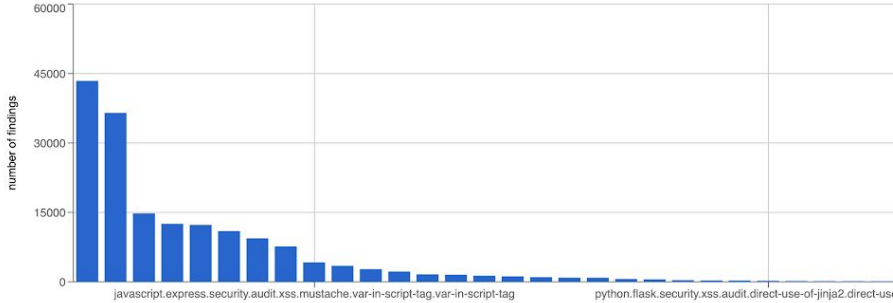
javascript.express.security.audit.xss.pug.explicit-unescape.template-explicit-unescape

filter exclude path prefixes...

Only Severity = ERROR

	Repository	Commit	Findings	Annotations	Action
1	https://github.com/GoogleCloudPlatform/nodejs-docs-samples	6b00a3c	3		results
2	https://github.com/bkimminich/juice-shop	e74095d	3		results
3	https://github.com/Requarks/wiki	6624df2	30		results
4	https://github.com/Requarks/wiki	df933f5	30		results
5	https://github.com/GoogleCloudPlatform/nodejs-docs-samples	a38120d	3		results
6	https://github.com/Requarks/wiki	a7b2b31	30		results
7	https://github.com/GoogleCloudPlatform/nodejs-docs-samples	7317547	3		results
8	https://github.com/bkimminich/juice-shop	bede3ff	3		results
9	https://github.com/GoogleCloudPlatform/nodejs-docs-samples	b50d29a	3		results
10	https://github.com/algolia/bootstrap	13bf8ae	2		results





3. Triaging

	annotations
ode!= JSON.stringify(...	

extra:


```
{
    "lines": "code!= JSON.stringify(val)",
    "message": "Detected an explicit unescape in a Pug template, using either\n'!=' or '!{...}'. If external data can reach these locations,\nyour application is exposed to a cross-site scripting (XSS)\nvulnerability. If yo
    "metadata": {
        "cwe": "CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')",
        "owasp": "A7: Cross-site Scripting (XSS)",
        "references": [
            "https://pugjs.org/language/code.html#unescaped-buffered-code",
            "https://pugjs.org/language/attributes.html#unescaped-attributes"
        ]
    },
    "severity": "WARNING"
}
```


appengine/pubsub/views/index.pug

```
1   doctype html
2   html(lang='en')
3     head
4       title PubSub
5       meta(charset='utf-8')
6     body
7       p Bearer tokens received by this instance:
8       ul
9         each val in tokens
10          li= val
11      p Claims received by this instance:
12      ul
13        each val in claims
14          li
15            code!= JSON.stringify(val)
16      p Messages received by this instance:
17      ul
18        each val in messages
19          li= val
20      p
21        small.
22          Note: because your application is likely running multiple instances,
23          each instance will have a different list of messages.
24
25      form(method='post')
26        textarea(name='payload', placeholder='Enter message here.')
27        button(type='submit') Send
28
```

javascript.express.security.audit.xss.pug.explicit-unescape.template-explicit-unescape

Positive	Negative	Unknown

(next: ctrl+n, previous: ctrl+p)

1. Proof of concept exploit
    a. Call graph analysis
    b. Construct payload
2. Risk evaluation
    a. Node module usage

4. Reporting

hackerone

bugcrowd

Node.js

The Node.js JavaScript Runtime

https://nodejs.org · @nodejs

Reports resolved	Assets in scope	Average bounty
21	1	$250

Submit report

Bug Bounty Program
Launched on Mar 2018

Bounty splitting enabled

Policy Hacktivity Thanks Updates (0)

Policy

Reporting a Bug in Node.js

All security bugs in Node.js are taken seriously. Please report any security issues here.

Your report will be acknowledged within 24 hours, and you'll receive a more detailed response within 48 hours indicating the next steps in handling your report.

After the initial reply to your report, the security team will endeavor to keep you informed of the progress being made towards a fix and full announcement, and may ask for additional information or guidance surrounding the reported issue. These updates will be sent at least every five days, in practice, this is more likely to be every 24-48 hours.

Response Efficiency

18 hrs
Average time to first response

4 days
Average time to triage

-
Average time to bounty

~10 minutes to write analysis
~10000 repos analyzed
~10 reports