

Fixing "Buffer without new" deprecation warnings

This article was originally published at

<https://gist.github.com/joepie91/a0848a06b4733d8c95c95236d16765aa>. Newer Node.js versions no longer behave in this exact way, but the information is kept here for posterity. If you have code that still uses `new Buffer`, you should still update it.

If you're using Node.js, you might run into a warning like this:

```
DeprecationWarning: Using Buffer without `new` will soon stop working.
```

The reason for this warning is that the Buffer creation API was changed to require the use of `new`. However, contrary to what the warning says, you should *not* use `new Buffer` either, [for security reasons](#). Any usage of it must be converted *as soon as possible* to `Buffer.from`, `Buffer.alloc`, or `Buffer.allocUnsafe`, depending on what it's being used for. Not changing it could mean a **security vulnerability** in your code.

Where is it coming from?

Unfortunately, the warning doesn't indicate *where* the issue comes from. If you've verified that *your own code* doesn't use `Buffer` without `new` anymore, but you're still getting the warning, then you are probably using an (outdated) dependency that still uses the old API.

The following command (for Linux and Cygwin) will list all the affected modules:

```
grep -rP '(?!new |[a-zA-Z])Buffer\(' node_modules | grep "\.js" | grep -Eo  
'^(node_modules/[^/:]*)*' | sort | uniq -c | sort -h
```

If you're on OS X, your `sort` tool will not have the `-h` flag. Therefore, you'll want to run this instead (but the result won't be sorted by frequency):

```
grep -rP '(?!new |[a-zA-Z])Buffer\(' node_modules | grep "\.js" | grep -Eo  
'^(node_modules/[^/:]*)*' | sort | uniq -c | sort
```

How do I fix it?

If the issue is in your own code, [this documentation](#) will explain how to migrate. If you're targeting older Node.js versions, you may want to use the `safe-buffer` [shim](#) to maintain compatibility.

If the issue is in a third-party library:

1. Run `npm ls <package name here>` to determine where in your dependency tree it is installed, and look at the top-most dependency (that isn't your project itself) that it originates from.
2. If that top-most dependency is out of date, try updating the dependency first, to see if the warning goes away.
3. If the dependency is *up-to-date*, that means it's an unfixed issue in the dependency. You should create an issue ticket (or, even better, a pull request) on the dependency's repository, asking for it to be fixed.

Revision #1

Created 2024-12-11 12:16:23 UTC by joepie91

Updated 2024-12-11 18:44:19 UTC by joepie91