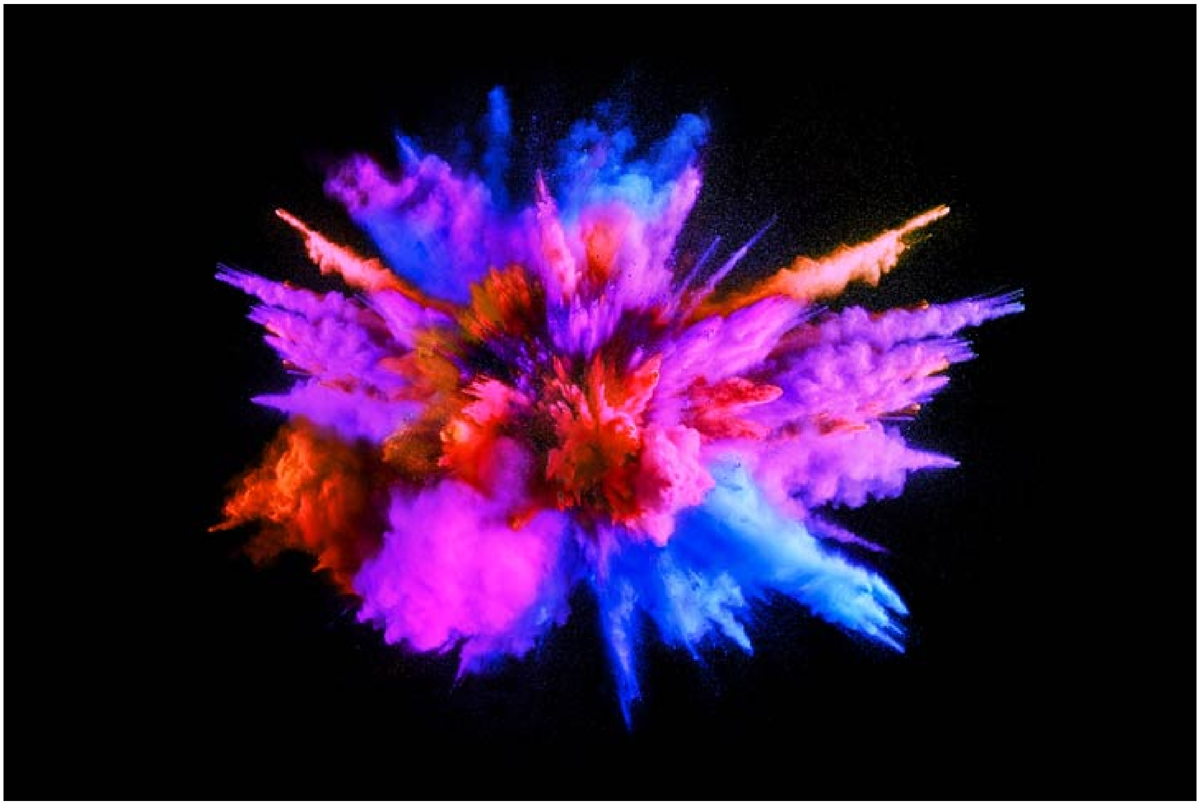
Sabotaged in Protest by their Maintainer—What to do Now?

January 10, 2022 By **Ax Sharma**

7 minute read time

SHARE:



In what can only be described as one of the most bizarre events in the history of open source, we find that the massively popular open source libraries, *colors.js*, and *faker.js* were sabotaged by their very own maintainer, as I **first reported** on over the weekend.

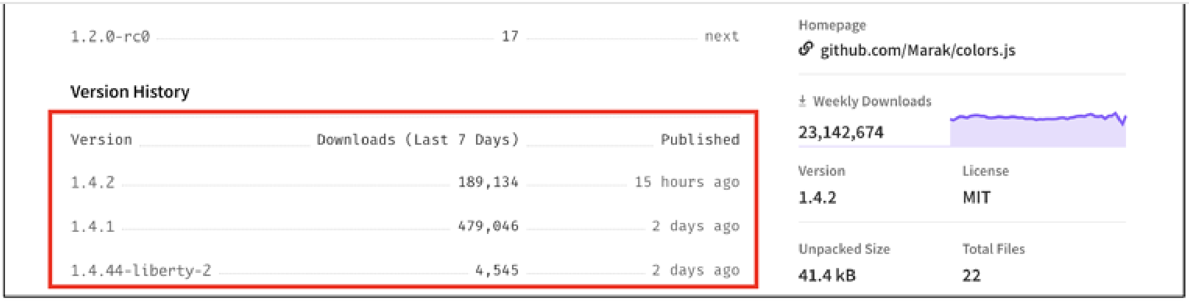
projects are developed and maintained by the same author, Marak Squires.

Note, hijacked versions of both projects were recorded as “malicious” in Sonatype’s data under identifiers: sonatype-2022-0215 and sonatype-2022-0216. The security data was made available to our customers the same day.

The immense download rate of these two components can be attributed to the basic, but essential, functionality they provide to JavaScript developers: ‘Colors’ lets you print colorful text messages on the console, whereas ‘faker’ helps devs generate fake data for their applications, for testing or staging purposes.

‘Colors’ and ‘faker’ Debacle Hits Thousands of Applications

Many open source developers found themselves at unease yesterday after suspecting that these popular dependencies–’colors’ and ‘faker’ had been compromised. Given the recent ua-parser-js and coa/rc npm library hijacking incidents, such an assumption would have been reasonable, after mysterious ‘colors’ versions 1.4.1, 1.4.2, and 1.4.44-liberty-2 appeared on npm:



1.2.0-rc0 17 next

Version History

Version	Downloads (Last 7 Days)	Published
1.4.2	189,134	15 hours ago
1.4.1	479,046	2 days ago
1.4.44-liberty-2	4,545	2 days ago

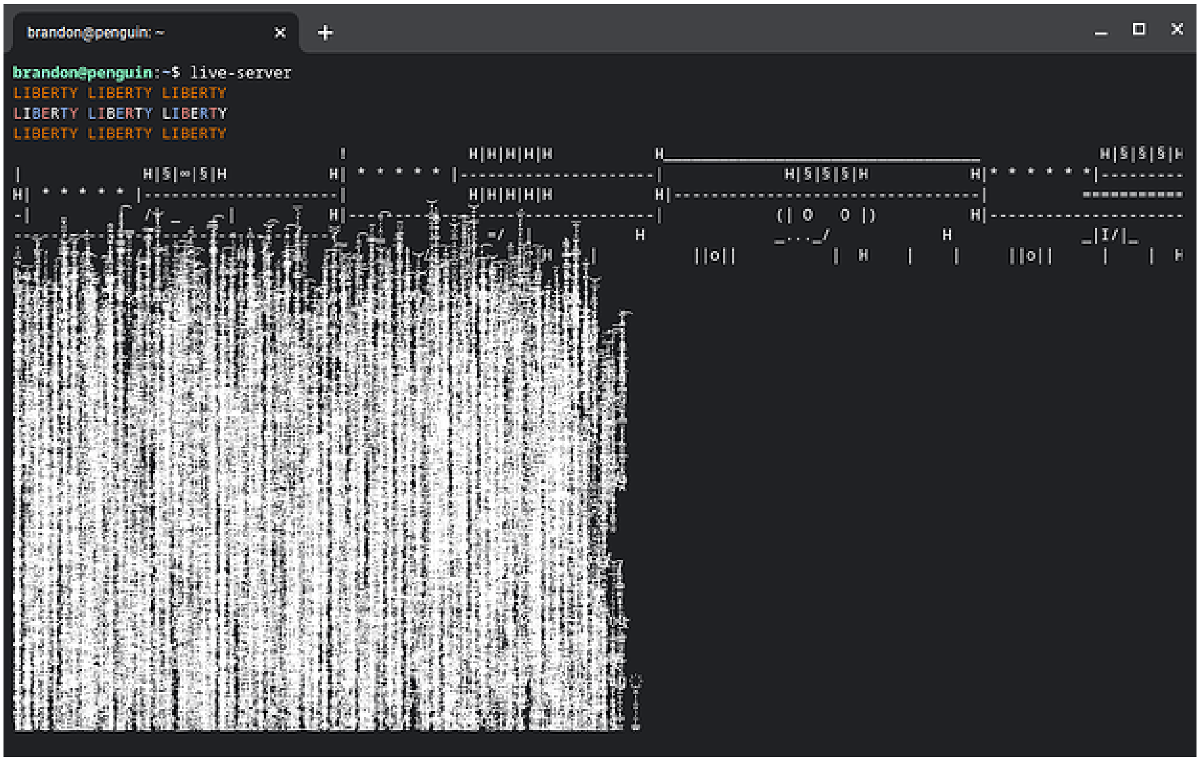
Homepage
github.com/Marak/colors.js

Weekly Downloads
23,142,674

Version	License
1.4.2	MIT

Unpacked Size	Total Files
41.4 kB	22

Those whose applications pulled this recently published ‘colors’ version found their applications caught in an infinite loop, printing ‘LIBERTY ‘LIBERTY LIBERTY’ followed by a sequence of gibberish non-ASCII characters aka **Zalgo text**:

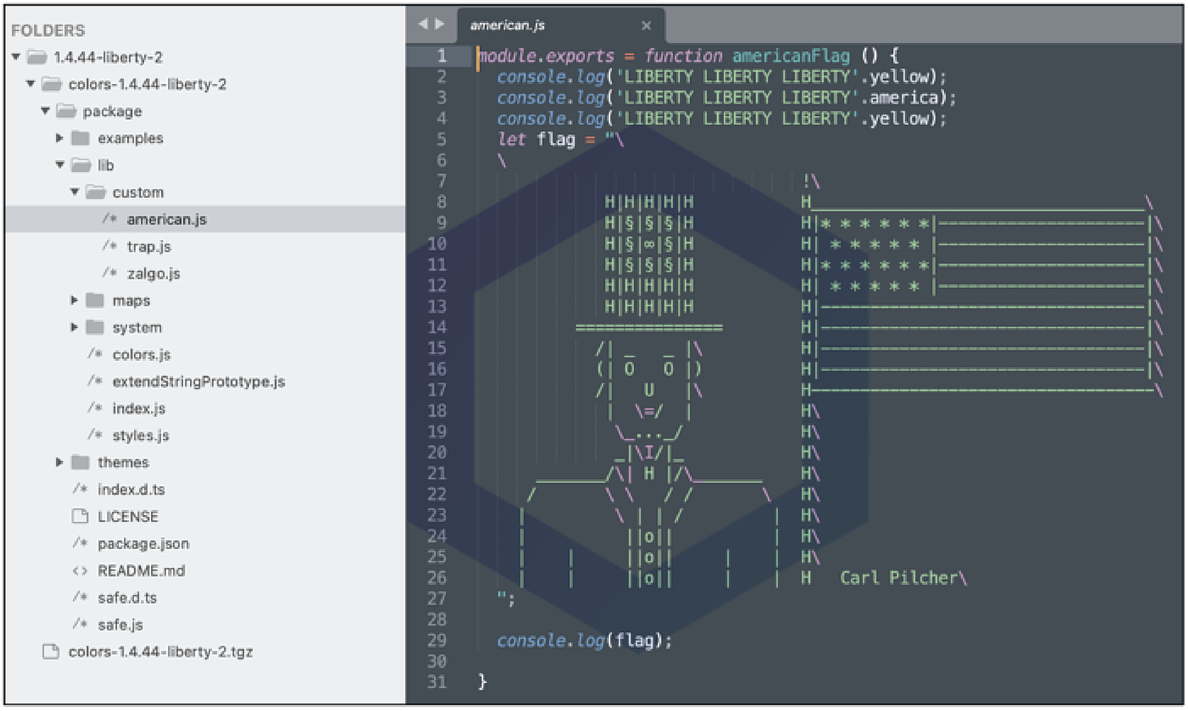


```
brandon@penguin:~$ live-server
LIBERTY LIBERTY LIBERTY
LIBERTY LIBERTY LIBERTY
LIBERTY LIBERTY LIBERTY
```

Developers contributing to well-known OSS projects, including **Amazon’s Cloud Development Kit** (aws-cdk), **Facebook’s Jest**, **Javascripting** project, “It's come to our attention that there is a zalgo bug in the v1.4.44-liberty-2 release of colors.

Please know we are working right now to fix the situation and will have a resolution shortly,” wrote the developer.

The culprit here, as verified by Sonatype, is the “**American flag module**” added by the developer in the aforementioned versions to print ‘LIBERTY’ text followed by the American flag:



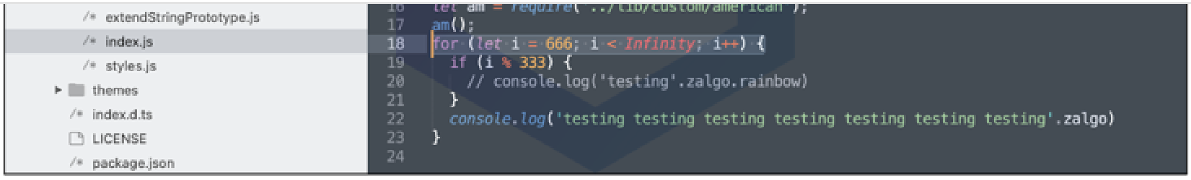
```
FOLDERS
▼ 1.4.44-liberty-2
  ▼ colors-1.4.44-liberty-2
    ▼ package
      ▶ examples
      ▼ lib
        ▼ custom
            american.js
            trap.js
            zalgo.js
        ▶ maps
        ▶ system
          colors.js
          extendStringPrototype.js
          index.js
          styles.js
      ▶ themes
        index.d.ts
        LICENSE
        package.json
        README.md
        safe.d.ts
        safe.js
    colors-1.4.44-liberty-2.tgz
```

```
module.exports = function americanFlag () {
  console.log('LIBERTY LIBERTY LIBERTY'.yellow);
  console.log('LIBERTY LIBERTY LIBERTY'.america);
  console.log('LIBERTY LIBERTY LIBERTY'.yellow);
  let flag = "\
  \
                               !\
            H|H|H|H|H          H________________________________\
            H|§|§|§|H          H|* * * * * *|-------------------|\
            H|§|∞|§|H          H| * * * * * |-------------------|\
            H|§|§|§|H          H|* * * * * *|-------------------|\
            H|H|H|H|H          H| * * * * * |-------------------|\
            H|H|H|H|H          H|-------------------------------|\
         ===============       H|-------------------------------|\
           /| _   _ |\         H|-------------------------------|\
           (| O   O |)         H|-------------------------------|\
           /|   U   |\         H---------------------------------\
            |  \=/  |          H\
             \_..._/           H\
             _|\I/|_           H\
      ______/\| H |/\_______   H\
     /      \ \   / /      \   H\
    |        \ | | /         |  H\
    |         ||o||         |  H\
    |    |    ||o||     |   |  H\
    |    |    ||o||     |   |  H   Carl Pilcher\
  ";

  console.log(flag);

}
```

Whereas the infinite loop is introduced in the lib/index.js file (line 18) and keeps rendering the Zalgo text in the user’s console indefinitely:



```
let am = require('../lib/custom/american');
am();
for (let i = 666; i < Infinity; i++) {
  if (i % 333) {
    // console.log('testing'.zalgo.rainbow)
  }
  console.log('testing testing testing testing testing testing testing'.zalgo)
}
```

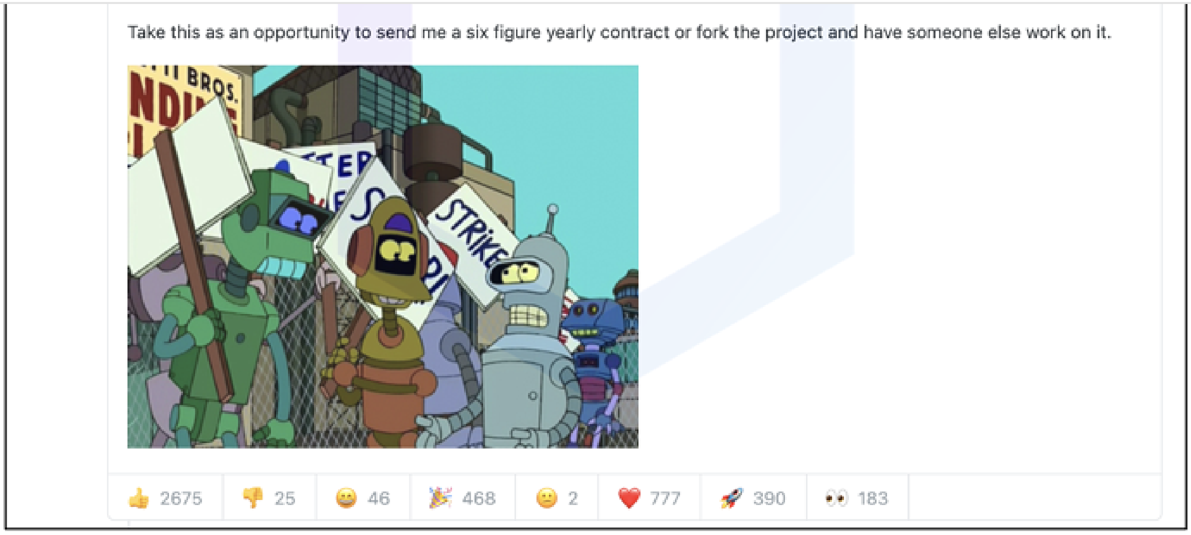
Self-sabotage in Protest Speaks to the Open Source Sustainability Problem

Upon digging deeper, it turns out that the developer himself introduced an infinite loop in colors, thereby sabotaging its functionality, and purged the functional code from the **‘faker’ package in version 6.6.6**. As of today, the front pages of both the latest npm version of ‘faker’ and its **modified GitHub repository** pose a provocative question: “What really happened with Aaron Swartz?”

For this protest, and since initially there was some confusion if the source of the problematic code was a project compromise, GitHub **allegedly suspended** the developer’s account. Although judging by the fact that GitHub account **‘Marak’** is accessible at the time of writing, the suspension appears to have been temporary.

On reaching out to Squires, I couldn’t get a definite answer as to what his motives were, but did come across a 2020 GitHub issue **published by the developer**.

In November 2020, the developer explicitly expressed an intention of no longer supporting big companies with their “free work” and that businesses should either fork the developer’s projects or compensate him handsomely with a six-figure annual salary:



Take this as an opportunity to send me a six figure yearly contract or fork the project and have someone else work on it.

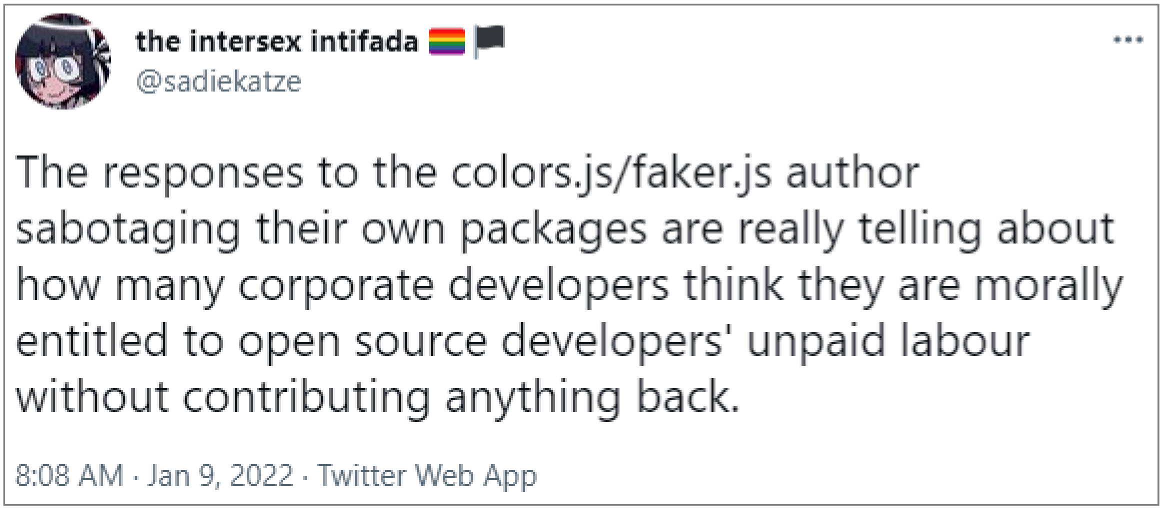
Moreover, the commit history of ‘colors’ and ‘faker’ seen by us also indicates these changes are intentionally pushed by the developer to push people to either fork the projects, resolve the bugs themselves, or support the open source developers that are largely volunteers, sustaining critical projects in their free time.

The incident draws attention to the issue of the Open Source Sustainability problem that **GitHub has previously written about**. Moreover, the incident follows the notorious Apache Log4j debacle that led to Log4j’s already understaffed development team of volunteers to work extra hours over the holidays as **many more CVEs kept coming up**, despite **not all of them** being worthy of attention.

The log4shell mass exploitation and the Log4j maintainers having been burdened with the responsibility of delivering urgent patches raised concerns as to how **mega corporations extensively depended-on open source projects** but did not give back enough to support the volunteers behind these projects. Subsequently, some casual bug bounty hunters and

> “Log4j maintainers have been working sleeplessly on mitigation measures; fixes, docs, CVE, replies to inquiries, etc. Yet nothing is stopping people to bash us, for work we aren't paid for, for a feature we all dislike yet needed to keep due to backward compatibility concerns.”

Interestingly, despite many not agreeing with Marak Squires’ bold move to sabotage his own projects, many have sided with him with one **tweeting**:



the intersex intifada 🏳️‍🌈🏴
@sadiekatze

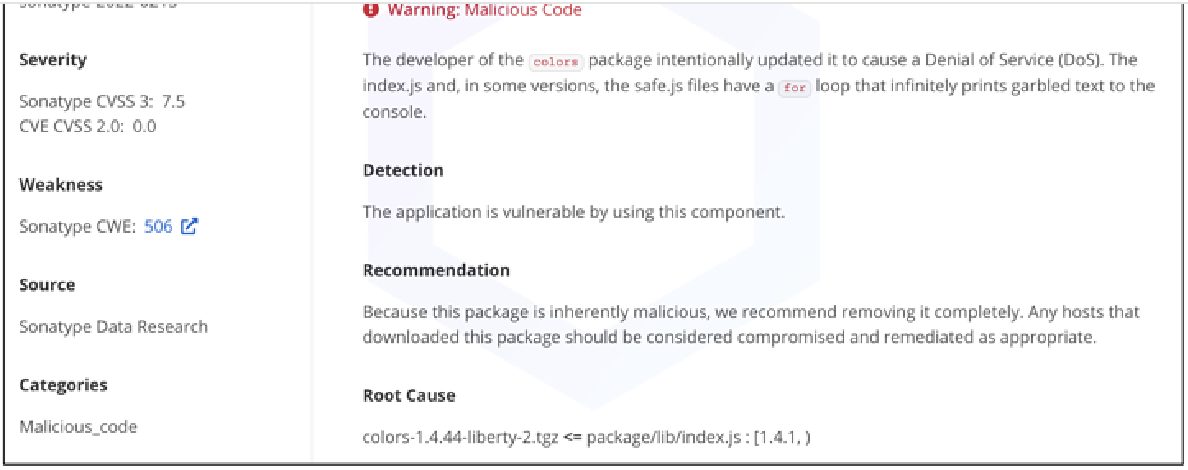
The responses to the colors.js/faker.js author sabotaging their own packages are really telling about how many corporate developers think they are morally entitled to open source developers' unpaid labour without contributing anything back.

8:08 AM · Jan 9, 2022 · Twitter Web App

Developers using 'colors' and 'faker' npm projects should ensure they are not using an unsafe version.

Sonatype customers using **Nexus Intelligence**-powered products, such as Nexus Lifecycle, need not worry as our security research and vulnerability data continues to protect their software development pipelines.

Thankfully, within hours of GitHub reports on the anomalous behavior exhibited by ‘colors’ and ‘faker,’ Sonatype’s security products began



Severity

Sonatype CVSS 3: 7.5
CVE CVSS 2.0: 0.0

Weakness

Sonatype CWE: 506

Source

Sonatype Data Research

Categories

Malicious_code

Warning: Malicious Code

The developer of the `colors` package intentionally updated it to cause a Denial of Service (DoS). The index.js and, in some versions, the safe.js files have a `for` loop that infinitely prints garbled text to the console.

Detection

The application is vulnerable by using this component.

Recommendation

Because this package is inherently malicious, we recommend removing it completely. Any hosts that downloaded this package should be considered compromised and remediated as appropriate.

Root Cause

colors-1.4.44-liberty-2.tgz <= package/lib/index.js : [1.4.1,)

For those not using Sonatype’s products powered by next-generation **Nexus Intelligence**, downgrading to an earlier version of colors (e.g. 1.4.0) and faker (e.g. 5.5.3) remains a solution.

It's also a good practice to pin your dependency versions to a specific, trusted version rather than automatically pulling in the latest version, without automated security checks in place to catch malicious releases. For example, **here's how** Amazon's *aws-cdk* project has achieved this.

If you are using Sonatype Lift, you can log into the web console and click Analyze Now on any repositories of concern. Lift will report usage of all vulnerable versions of Log4j. **Install now**.

Finally, even if you aren’t using any of Sonatype’s products, Sonatype offers a **free vulnerability scanner** scanner you can download or use online. The report will detail usage of all vulnerable versions of Log4j in your repositories.

Tags: **vulnerabilities**, **featured**, **Nexus Intelligence Insights**, **DevZone**