FasterXML / **jackson-databind** Public

Code Issues 457 Pull requests 19 Actions Projects Wiki ···

New issue Jump to bottom

Add check in primitive value deserializers to avoid deep wrapper array nesting wrt `UNWRAP_SINGLE_VALUE_ARRAYS` [CVE-2022-42003] #3590

Closed **cowtowncoder** opened this issue on Sep 5 · 44 comments

Labels 2.14 CVE

Milestone 2.14.0-rc1

cowtowncoder commented on Sep 5 • edited Member

TL;DNR:

Fix included in:

- 2.14.0 once released (until then, 2.14.0-rc1 and rc2)
- 2.13.4.2 micro-patch (jackson-bom 2.13.4.20221013). (NOTE: 2.13.4.1/2.13.4.20221012 have an issue that affects Gradle users)
- 2.12.7.1 micro-patch (jackson-bom 2.12.7.20221012)

(note: similar to #3582)
(note: originally found via oss-fuzz https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=51020)

Implementation of methods like `_parseBooleanPrimitive` (in `StdDeserializer`) uses idiom:

```
        if (ctxt.isEnabled(DeserializationFeature.UNWRAP_SINGLE_VALUE_ARRAYS)) {
            p.nextToken();
            final boolean parsed = _parseBooleanPrimitive(p, ctxt);
            _verifyEndArrayForSingle(p, ctxt);
            return parsed;
        }
```

to handle unwrapping. While simple this exposes possibility of "too deep" nesting and possible problem with resource exhaustion in some cases. We should change this similar to how #3582 was handled.

cowtowncoder added the **to-evaluate** label on Sep 5

cowtowncoder commented on Sep 5 • edited Member Author

Methods to update (and regression test):

- `_parseBooleanPrimitive()`
- `_parseBytePrimitive()`
- `_parseShortPrimitive()`
- `_parseIntPrimitive()`
- `_parseLongPrimitive()`
- `_parseFloatPrimitive()`
- `_parseDoublePrimitive()`
- `_parseDateFromArray()` (maybe?)

Also note that method `_deserializeWrappedValue()` implements checks although cannot quite be called as-is.
Similarly, `BeanDeserializer._deserializeFromArray()` at around line 632 has similar checks.

cowtowncoder added 2.14 and removed **to-evaluate** labels on Sep 5

cowtowncoder changed the title ~~Add check in primitive value deserializers to avoid wrapper array nesting wrt `UNWRAP_SINGLE_VALUE_ARRAYS`~~ **Add check in primitive value deserializers to avoid deep wrapper array nesting wrt `UNWRAP_SINGLE_VALUE_ARRAYS`** on Sep 6

cowtowncoder closed this as completed in `d78d00e` on Sep 6

---

henryrneh commented on Sep 9

Hello dear **@cowtowncoder**,

I am Henry from Code Intelligence. First of all thank you for your quick fixes of this issue!

https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=51020

Is this issue regarded as a security issue? If yes, we are thinking about applying CVE for it, so the community knows about it and will update to the latest version of jackson-databind.

Thank you for your fixes and support for OSS-Fuzz!

Best regards,
Henry

cowtowncoder commented on Sep 9 Member Author

@henryrneh Same as #3582 (see my comments there).

👍 1

henryrneh commented on Sep 9

@cowtowncoder thank you 👍

DavidKorczynski commented on Sep 12

This issue was found by a fuzzer written by the Ada Logics team and is part of an ongoing security assessment. **@henryrneh** can you please ensure the issues you report are found by the fuzzers written by your team (https://github.com/google/oss-fuzz/blob/master/projects/jackson-core/JsonFuzzer.java and https://github.com/google/oss-fuzz/blob/master/projects/jackson-databind/ObjectReaderFuzzer.java) then we'll take care of those from our fuzzers.

henryrneh commented on Sep 12 • edited ▾

I already canceled the application. We will do our best to try not to apply CVEs for fuzz targets written by AdaLogics, however we will need some assitance or notification by you to know who wrote which fuzz target, because OSS-Fuzz is not designed to support this, maybe you can use some special prefix in the fuzz target name so it's more obvious for us so we can filter it out?

This was referenced on Oct 2

Denial of Service (DoS) SNYK-JAVA-COMFASTERXMLJACKSONCORE-3038426 RADAR-base/radar-jersey#66

✓ Closed

Denial of Service (DoS) SNYK-JAVA-COMFASTERXMLJACKSONCORE-3038426 RADAR-base/radar-output-restructure#532

Closed

chadlwilson mentioned this issue on Oct 3

Temporarily suppress CVE-2022-42003 Jackson vulnerability gocd/gocd#10890

Merged

pjfanning commented on Oct 5 Member

@henryrneh the disclosure in https://nvd.nist.gov/vuln/detail/CVE-2022-42003 should really have been delayed until jackson-databind v2.14.0 was released. The RC should have been given a good chance to be tested and an orderly release of 2.14.0 done when it is ready. Now we risk having people clamouring for an early release of 2.14.0.

👍 4

DavidKorczynski commented on Oct 5

@pjfanning -- **@AdamKorcz** and I applied for this CVE and was done in coordination with the current audit of jackson. We coordinated this with Tatu in terms of agreeing we should apply for CVEs for these issues.

👎 2

jgoeres commented on Oct 5

Our OWASP scan just ran into this finding for our use of jackson-databind 2.13.3. We are very hesitant to go to an RC of 2.14 (and of course fully understand that this issue cannot change the release plans for 2.14, so we will *not* join the crowd that is pushing for an "early release of 2.14" **@pjfanning** :-)).
But since 2.13 is an actively maintained branch according to https://github.com/FasterXML/jackson#active-developed-jackson-2x-branches, I assume a fix will be made available there as well? Any activities or maybe even an ETA for that?

👍 9

pjfanning commented on Oct 5 • edited ▾ Member

@jgoeres **@alzimmermsft** are you even using the UNWRAP_SINGLE_VALUE_ARRAYS option?

alzimmermsft commented on Oct 5 • edited ▾

I'll second what **@jgoeres**, that we're getting security alerts around usage of Jackson 2.13.4 and won't be able to upgrade to the 2.14 RC any time soon and downstream consumers of our SDKs will also get alerts as well. We aren't using `UNWRAP_SINGLE_VALUE_ARRAYS` but a lot of noise will still be caused.

Edit:

I know that the fix for this issue now results in an exception being thrown, which is a runtime breaking change within a patch release, but there has been prior art on this being accepted with the fix for #2816. There was also API changes as well, though looking at the fix for this issue the shared method could be made non-protected if it's backported and shouldn't be exposed.

👍 6

pjfanning commented on Oct 5 Member

> **@pjfanning** -- **@AdamKorcz** and I applied for this CVE and was done in coordination with the current audit of jackson. We coordinated this with Tatu in terms of agreeing we should apply for CVEs for these issues.

Just watch the comments flow in looking for special attention now this change is made public. I've already had swagger-akka-http/swagger-akka-http#342

jgoeres commented on Oct 5 • edited ▾

@pjfanning
We have roughly dozen microservices spread across as many teams, we haven't checked for all of them. But even if none of them used that setting, scans done in various places (both within our organization, but also by customers that run this on-prem) will find the lib and this will lead to irritation and demands to resolve it by updating even if we claim that it is not exploitable.

👍 3

bovy89 mentioned this issue on Oct 5

system wide vulnerability management DependencyTrack/dependency-track#1495

⊙ Open

alzimmermsft mentioned this issue on Oct 5

CVE-2022-42003 Jackson vulnerability Azure/azure-sdk-for-java#31277

⊙ Open

cowtowncoder changed the title ~~Add check in primitive value deserializers to avoid deep wrapper array nesting wrt `UNWRAP_SINGLE_VALUE_ARRAYS`~~ Add check in primitive value deserializers to avoid deep wrapper array nesting wrt `UNWRAP_SINGLE_VALUE_ARRAYS` [CVE-2022-42003] on Oct 5

cowtowncoder added the CVE label on Oct 5

cowtowncoder commented on Oct 5 Member Author

@DavidKorczynski I think we need coordinate better in future: my understanding was that a CVE would be filed, I'd be notified with id allocated (so I can update this issue, release notes), but CVE only made public once official release was out.
But perhaps I am assuming there is a mechanism that does not exist between allocating an id and making it available for the public.

What is needed at minimum, at any rate, is the linkage between CVE and issues. The problem right now is that users email me or file issues asking "what's up with CVE-2022-xxxx" and then I have to go digging. It's not a huge effort of course but is an interruption, and something we can avoid by collaboration.

We may also need to discuss some of finer details: in this case I hope I mentioned (but perhaps I didn't?) that this setting is disabled by default -- that should have an effect in ranking.

I realize I did not quite realize everything that needs to be discussed before I say it's fine to proceed with CVE filing.
Live and learn. Thank you everyone.

❤️ 8

cowtowncoder commented on Oct 5 Member Author

Note: also added the "sibling" CVE marker (2022-42004) for #3582.

cowtowncoder added a commit that referenced this issue on Oct 5

`Add CVE markers for #3582, #3590` 450a2d9

cowtowncoder added this to the **2.14.0-rc1** milestone on Oct 5

graemerocher mentioned this issue on Oct 6

Upgrade to Jackson 2.14.0 micronaut-projects/micronaut-core#8124
Closed

34 hidden items
Load more...

ppatierno mentioned this issue on Oct 14

Bumped Jackson Databind dependency to fix CVE-2022-42003 strimzi/strimzi-kafka-bridge#670
Merged

vdotjansen commented on Oct 14 • edited

@chadlwilson

> Thanks a lot **@cowtowncoder** . I have contacted NVD/NIST with the relevant information and am seeking to get https://nvd.nist.gov/vuln/detail/CVE-2022-42003 updated to reflect these micro-patch releases so people's scanner noise goes away if they update to these (or via relevant BOMs).
>
> Will follow up for the separate CVE fix backported to `2.12.x` if I get a positive response/outcome.

They updated the CVE as the description changed from
"In FasterXML jackson-databind before 2.14.0-rc1, resource exhaustion can occur because of a lack of a check in primitive value deserializers to avoid deep wrapper array nesting, when the UNWRAP_SINGLE_VALUE_ARRAYS feature is enabled."
to
"In FasterXML jackson-databind before 2.14.0-rc1, resource exhaustion can occur because of a lack of a check in primitive value deserializers to avoid deep wrapper array nesting, when the UNWRAP_SINGLE_VALUE_ARRAYS feature is enabled. Additional fix version in 2.13.4.1 and 2.12.17.1".

But they forgot to update the CPE, therefor it will still match the new versions.

I e-mailed them because I mostly look at the CPE, did not notice the change in the description at first.

🚀 1

radarsh commented on Oct 14

> did not notice the change in the description at first.

You are not alone.

Have they been informed about 2.13.4.2 as well?

chadlwilson commented on Oct 14 • edited

> They updated the CVE as the description changed from "In FasterXML jackson-databind before 2.14.0-rc1, resource exhaustion can occur because of a lack of a check in primitive value deserializers to avoid deep wrapper array nesting, when the UNWRAP_SINGLE_VALUE_ARRAYS feature is enabled." to "In FasterXML jackson-databind before 2.14.0-rc1, resource exhaustion can occur because of a lack of a check in primitive value deserializers to avoid deep wrapper array nesting, when the UNWRAP_SINGLE_VALUE_ARRAYS feature is enabled. Additional fix version in 2.13.4.1 and 2.12.17.1".
>
> But they forgot to update the CPE, therefor it will still match the new versions.

@vdotjansen What you possibly need to know (but may not?) is that descriptions as they appear in NVD come from and are mastered in MITRE rather than the NVD -- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-42003 . Likely someone has updated it there; possibly via the CNA/submitter or with a form submission to MITRE itself and the description has flowed through to the NVD in feeds. That won't do anything for CVSS scores, CPEs, affected versions etc. These parts are mastered in the NVD and MITRE knows nothing about them. When changes flow through from MITRE, they tend to get marked as

> This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

So they didnt "forget" to update the CPE, they just haven't updated it yet.

> I e-mailed them because I mostly look at the CPE, did not notice the change in the description at first.

NVD haven't actually replied to my original email, and it's still marked as being under analysis due to this description change - so they will probably get to it. I'd advise being patient. I've included the details already about the CPEs needing adding as well as the adjustment to the versions. I've done this a few times before, and it has always worked ... eventually.

> Have they been informed about 2.13.4.2 as well?

@radarsh It's not *strictly* necessary for them to have every version listed unless it's at the boundary of a fix or vuln, but since it is still in their analysis queue, they will likely add that version at the same time.

The process is slow and manual and non-transparent, but I suspect 100 of us emailing them isn't going to make it go any faster....

👍 4

vdotjansen commented on Oct 14 • edited

> **@vdotjansen** What you possibly need to know (but may not?) is that descriptions as they appear in NVD come from and are mastered in MITRE rather than the NVD -- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-42003 . Likely someone has updated it there; possibly via the CNA/submitter or with a form submission to MITRE itself and the description has flowed through to the NVD in feeds. That won't do anything for CVSS scores, CPEs, affected versions etc. These parts are mastered in the NVD and MITRE knows nothing about them. When changes flow through from MITRE, they tend to get marked as

I knew that there was data shared between MITRE and NVD, but did not know which part came from MITRE although I did know at least the CPE's are managed by NVD.

> NVD haven't actually replied to my original email, and it's still marked as being under analysis due to this description change - so they will probably get to it. I'd advise being patient. I've included the details already about the CPEs needing adding as well as the adjustment to the versions. I've done this a few times before, and it has always worked ... eventually.

Like I said I only checked the CPE.

> The process is slow and manual and non-transparent, but I suspect 100 of us emailing them isn't going to make it go any faster....

You are correct and I should have checked here first. My e-mail was send before I noticed your message here. We should not all send an e-mail you are correct. I should have checked here to see, but I think it is better to send mails by 2 different people then 0 mails, as they need to know this. Next time I will check the issues to see if someone already send a mail. Me posting here that I also send an e-mail is hoping there will not be more e-mails towards them about this :)

👍 2

snuyanzin mentioned this issue on Oct 14

[FLINK-29631] Update Jackson-bom to 2.13.4.20221013 apache/flink-shaded#113

Merged

chadlwilson commented on Oct 14

NIST/NVD have got back to me after reviewing the publicly available info and are updating the affected versions right now.

I am working with the assumption that all 2.11.x and earlier versions are also affected, but if the UNWRAP_SINGLE_VALUE_ARRAYS feature in question was only added some time along the way in a 2.x version the CVE mapping could be more specific. Possibly a bit moot given the other vulnerabilities likely to exist in those earlier versions anyway...

🚀 3

davalos-adrian mentioned this issue on Oct 14

CVE-2022-42003 reported for jackson-databind 2.13.2.1 oracle/oci-java-sdk#449

Closed

cxronen mentioned this issue on Oct 12

CVE-2022-42003 @ Maven-com.fasterxml.jackson.core:jackson-databind-2.0.4 cxronen/BookStore_VSCode#231

Open

github-actions bot mentioned this issue on Oct 17

Denial of Service (DoS) SNYK-JAVA-COMFASTERXMLJACKSONCORE-3038426 RADAR-base/RADAR-Rest-Source-Auth#204

Closed

natedanner mentioned this issue on Oct 17

Update jackson bom for CVE-2022-42003 openrewrite/rewrite#2310

Merged

robander mentioned this issue on Oct 17

Security problem in Jackson libraries dita-ot/dita-ot#4016

Open

frantuma mentioned this issue on Oct 18

Update pom.xml swagger-api/swagger-core#4276

Closed

chadlwilson commented on Oct 18 • edited

The CVE affected versions for this issue are finally updated and should be correct now. There was a bit of back-and-forth with the NIST/NVD folks and a weekend delay, but got there eventually...

👍 1 🎉 2 ❤️ 1 🚀 1

RetroDreams commented on Oct 19

@chadlwilson Maybe I just need to wait for another database update but 26 hours ago my Anchore vuln scanner reported this was fixed via 2.13.4.1 and now my scanner is reporting that NVD says CVE-2022-42003 has no fix available. This is directly contradictory to what is displayed on NIST? I am thoroughly confused.

```
{
            "feed": "vulnerabilities",
            "feed_group": "nvd",
            "fix": "None",
            "nvd_data": [
                {
                    "cvss_v2": {
                        "base_score": -1.0,
                        "exploitability_score": -1.0,
                        "impact_score": -1.0
                    },
                    "cvss_v3": {
                        "base_score": 7.5,
                        "exploitability_score": 3.9,
                        "impact_score": 3.6
                    },
                    "id": "CVE-2022-42003"
                }
            ],
            "package": "jackson-databind-2.13.4.1",
            "package_cpe": "None",
            "package_cpe23": "cpe:2.3:a:fasterxml:jackson-databind:2.13.4.1:*:*:*:*:*:*:*",
            "package_name": "jackson-databind",
            "package_path": "/root/.gradle/wrapper/dists/gradle-7.5.1-
bin/7jzzequgds1hbszbhq3npc5ng/gradle-7.5.1/lib/plugins/jackson-databind-2.13.4.1.jar",
            "package_type": "java",
            "package_version": "2.13.4.1",
            "severity": "High",
            "url": "https://nvd.nist.gov/vuln/detail/CVE-2022-42003",
            "vendor_data": [],
            "vuln": "CVE-2022-42003",
            "will_not_fix": false
}
```

chadlwilson commented on Oct 19

@RetroDreams not sure. Seems like something you'd need to check in your Anchore deployment.

github-actions bot mentioned this issue on Oct 24

Denial of Service (DoS) SNYK-JAVA-COMFASTERXMLJACKSONCORE-3038426 RADAR-base/radar-app-config#32

Closed

dlvenable mentioned this issue last month

Jackson 2.13.4.2 opensearch-project/data-prepper#1925

Merged

4 tasks

tfmorris mentioned this issue 26 days ago

fix(sec): upgrade com.fasterxml.jackson.core:jackson-databind to 2.14.0-rc1 OpenRefine/OpenRefine#5395

Closed

HugoVeillette mentioned this issue 25 days ago

Why micro-patch in open branch 2.13? #3648

Closed

cxronen mentioned this issue 23 days ago

CVE-2022-42003 @ Maven-com.fasterxml.jackson.core:jackson-databind-2.9.10.7 cxronen/AST_BookStore#247

Open

This was referenced 21 days ago

Fix Jackson databind vulnerability CVE-2022-42003 redisson/redisson#4654

Closed

Bump jackson-bom to 2.13.4.20221013 (upgrade to jackson-databind 2.13.4.2) redisson/redisson#4618

Merged

ws-ghe bot mentioned this issue 16 days ago

s3-2.13.10.jar: 15 vulnerabilities (highest severity is: 7.5) kyalwss/Test-00101604#2

Open

sonnyhcl mentioned this issue 12 days ago

[GHSA-jjjh-jjxp-wpff] Uncontrolled Resource Consumption in Jackson-databind github/advisory-database#821

Merged

J3173 mentioned this issue 9 days ago

Update Jackson Databind to version 2.13.4.2 openEHR/archie#449

Merged

maxgrossman mentioned this issue 3 days ago

update fasterxml to 2.14.1 ngageoint/hootenanny#5505

Merged

odl-github pushed a commit to opendaylight/odlparent that referenced this issue yesterday

```
Bump jackson-databind to 2.13.4.2   …                                    36012dd
```

odl-github pushed a commit to opendaylight/odlparent that referenced this issue yesterday

```
Bump jackson-databind to 2.13.4.2   …                                    2754281
```

odl-github pushed a commit to opendaylight/odlparent that referenced this issue yesterday

```
Bump jackson-databind to 2.13.4.2   …                                    cb0ec8f
```

Assignees

No one assigned

Labels

2.14 CVE

Projects

None yet

Milestone

2.14.0-rc1

Development

No branches or pull requests

13 participants