New issue   **Jump to bottom**

Add check in `BeanDeserializer._deserializeFromArray()` to prevent use of deeply nested arrays [CVE-2022-42004] #3582

⊘ **Closed**   cowtowncoder opened this issue on Aug 23 · 23 comments

Labels   2.14   **CVE**

Milestone   ⇥ **2.13.4**

---

cowtowncoder commented on Aug 23 · **edited** ▾   Member

Fix included in

- 2.13.4
- 2.12.7.1 micro-patch (jackson-bom 2.12.7.20221012)

(note: found by oss-fuzz, see: https://bugs.chromium.org/p/oss-fuzz/issues)

Currently feature `DeserializationFeature.UNWRAP_SINGLE_VALUE_ARRAYS` is supported by most types, and deserializers tend to implement support using recursion, effectively allowing multiple nested layers of JSON Arrays to be unwrapped.
This is not a feature to support but just an implementation detail; ideally we should only allow a single JSON Array to wrap a value.

I think I have removed ability for deeper nesting from some other types so there may be some prior art.

🏷 🧑 cowtowncoder added **to-evaluate** 2.14 and removed **to-evaluate** labels on Aug 23

⌁ cowtowncoder added a commit that referenced this issue on Aug 23

🧑 `Add a (failing) test for` **#3582**   ✕ 35de19e

cowtowncoder added the `has-failing-test` label on Aug 23

cowtowncoder changed the title ~~Add check in `BeanDeserializer._deserializeFromArray()` to try to prevent use of deeply nested arrays~~ Add check in `BeanDeserializer._deserializeFromArray()` to prevent use of deeply nested arrays on Aug 23

cowtowncoder closed this as completed in `0631835` on Aug 23

---

cowtowncoder added this to the **2.14.0** milestone on Sep 3

cowtowncoder mentioned this issue on Sep 5

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

⊘ **Closed**

---

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=50490

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 I think it is reasonable to file a CVE for this, although one caveat is that it is only applicable if users enable specific `DeserializationFeature` and not with vanilla (default) setting of `ObjectMapper`. So that should probably at least be reflect in applicability -- I do not have any statistics of how common enabling this feature is but it probably is minority of usage.

👍 4

henryrneh commented on Sep 9

@cowtowncoder thanks for your reply and info! We will communicate the information and handle the application to Google. Many thanks!

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 13

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?

DavidKorczynski commented on Sep 13

> We will do our best to try not to apply CVEs for fuzz targets written by AdaLogics

Great, thanks!

> we will need some assitance or notification by you to know who wrote which fuzz target

Do the links I provided above suffice?

henryrneh commented on Sep 14 • edited ▾

Thank you that works! In the future when AdaLogics add a new fuzz target please let us know or add some prefix to the name, so this will not happen again

👍 1

DavidKorczynski commented on Sep 14

> Thank you that works! In the future when AdaLogics add a new fuzz target please let us know or add some prefix to the name, so this will not happen again

sounds good -- I'll also send over an email after the assessment so you can see details about the findings we got using Jazzer

👍 1

---

henryrneh commented on Sep 14

Great, thank you!

📤 🔵 **ws-on-ws** (bot) mentioned this issue on Oct 2

jackson-databind-2.13.3.jar: 2 vulnerabilities (highest severity is: 7.5) whitesource/agents#85

⊙ Open

📤 ⚫ **github-actions** (bot) mentioned this issue on Oct 2

Denial of Service (DoS) SNYK-JAVA-COMFASTERXMLJACKSONCORE-3038424 RADAR-base/radar-jersey#65

⊘ Closed

📤 🔴 **ws-ghe** (bot) mentioned this issue on Oct 4

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

⊙ Open

📤 **eperret** added a commit to eperret/omakase that referenced this issue on Oct 4

🧑 `fix: updated version of com.fasterxml.jackson` ⋯   ✕ 24b6caf

📤 🧑 **eperret** mentioned this issue on Oct 4

fix: updated version of com.fasterxml.jackson salesforce/omakase#36

⎇ Merged

📤 **eperret** added a commit to salesforce/omakase that referenced this issue on Oct 4

🧑

fix: updated version of com.fasterxml.jackson (#36) ⋯ ✓ 5222837

eperret added a commit to eperret/omakase that referenced this issue on Oct 4

fix: updated version of com.fasterxml.jackson ⋯ 6334dcf

ws-ghe (bot) mentioned this issue on Oct 4

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

⊙ Open

eperret added a commit to eperret/omakase that referenced this issue on Oct 5

fix: updated version of com.fasterxml.jackson ⋯ e17c652

cowtowncoder changed the title ~~Add check in `BeanDeserializer._deserializeFromArray()` to prevent use of deeply nested arrays~~ Add check in `BeanDeserializer._deserializeFromArray()` to prevent use of deeply nested arrays [CVE-2022-42004] on Oct 5

cowtowncoder added the CVE label on Oct 5

4 hidden items

Load more...

cowtowncoder added a commit that referenced this issue on Oct 5

Add CVE markers for #3582, #3590 450a2d9

eperret added a commit to eperret/omakase that referenced this issue on Oct 5

fix: updated version of com.fasterxml.jackson ⋯ 4605a42

eperret added a commit to eperret/omakase that referenced this issue on Oct 5

fix: updated version of com.fasterxml.jackson ⋯ 7192eda

dersvenhesse mentioned this issue on Oct 7

Vulnerabilities found in jackson-databind v2.13.4 & v2.13.3 #3616

⊘ Closed

---

abstractj mentioned this issue on Oct 10

[CVE-2022-42004]- Denial of Service (DoS) vulnerability in com.fasterxml.jackson.core:jackson-databind keycloak/keycloak#14833

⊘ Closed

---

wakingrufus commented on Oct 11

Is there any plan to backport this fix to 2.12 for those of us using jersey1 still?

👍 1

---

cowtowncoder commented on Oct 11   Member   Author

@wakingrufus Very unlikely unless it turns out impossible to upgrade to Jackson 2.13.x and there is a lot of usage. I'd rather help resolve issues, if any, for Jackson upgrade (meaning 2.13.x fixes to let Jersey 1.x use it if there is something blocking that -- not aware of anything as of now).

---

wakingrufus commented on Oct 11

> @wakingrufus Very unlikely unless it turns out impossible to upgrade to Jackson 2.13.x and there is a lot of usage. I'd rather help resolve issues, if any, for Jackson upgrade (meaning 2.13.x fixes to let Jersey 1.x use it if there is something blocking that -- not aware of anything as of now).

The issue is this change: FasterXML/jackson-jaxrs-providers#134

---

cowtowncoder commented on Oct 11   Member   Author

Ah. Ok, that is good to know and yes, would prevent clean upgrade.
I think it is actually quite possible that 2.13.x of core components (jackson-annotation, jackson-core, jackson-databind) would work ok with `jackson-jaxrs-json-provider` 2.12.x but that is not a recommended configuration as a general rule.

Question then is... how widely is Jersey 1.x still used? Do actively maintained framework have strong dependency on it?
I guess I'd be open to backporting via separate issue filed by someone who needs that; and that can then be processed as separate from this discussion.

cesarhernandezgt pushed a commit to cesarhernandezgt/jackson-databind that referenced this issue on Oct 11

Fix FasterXML#3582 ...   a74038b

---

cesarhernandezgt commented on Oct 11   Contributor

I just pushed a backport for #3590 and Fix #3582 via PR: #3622
It's my first contribution to this project so CLA was just sent :).

🎉 1

---

wakingrufus commented on Oct 11

> Ah. Ok, that is good to know and yes, would prevent clean upgrade.
> I think it is actually quite possible that 2.13.x of core components (jackson-annotation, jackson-core, jackson-databind) would work ok with `jackson-jaxrs-json-provider` 2.12.x but that is not a recommended configuration as a general rule.
>
> Question then is... how widely is Jersey 1.x still used? Do actively maintained framework have strong dependency on it?
> I guess I'd be open to backporting via separate issue filed by someone who needs that; and that can then be processed as separate from this discussion.

The only reason I'm not using `jackson-jaxrs-json-provider` 2.12.x is that Jackson libraries seem to have a transitive reference to the BOM now which, in gradle, prevents a mixed version scenario (probably for the best).
As for what is using Jersey 1, it is an internal framework in my org. We are working on upgrading it, but I was just wondering if the backport is planned because we would use it in the meantime
Thanks!

👍 1

---

cesarhernandezgt mentioned this issue on Oct 12

Backport Fix #3590 and Fix #3582 to 2.12 branch #3623
✓ Closed

---

cesarhernandezgt commented on Oct 12   Contributor

> I just pushed a backport for #3590 and Fix #3582 via PR: #3622

Yesterday I forgot to create the GitHub issue, here it's is: #3623

cowtowncoder pushed a commit that referenced this issue on Oct 12

backport Fix #3590 and Fix #3582 (#3622)   cd09097

cowtowncoder commented on Oct 12   Member   Author

Thanks to @cesarhernandezgt 's help, there is now `jackson-databind` 2.12.7.1; and matching `jackson-bom`
version `2.12.7.20221012`. In addition to original fix going in 2.13.4.

debora-ito mentioned this issue on Oct 13

jackson-databind before 2.14.0-rc1 has security vulnerability CVE-2022-42003 aws/aws-sdk-
java#2861

⊘ Closed

cxronen mentioned this issue on Oct 12

CVE-2022-42004 @ Maven-com.fasterxml.jackson.core:jackson-databind-2.0.4
cxronen/BookStore_VSCode#230

⊙ Open

github-actions (bot) mentioned this issue on Oct 17

Denial of Service (DoS) SNYK-JAVA-COMFASTERXMLJACKSONCORE-3038424 RADAR-
base/RADAR-Rest-Source-Auth#203

⊘ Closed

chadlwilson commented on Oct 18

FYI, I have contacted NVD to get the "known affected software configurations" on
https://nvd.nist.gov/vuln/detail/CVE-2022-42004 updated to reflect the backport/micro-patch in `2.12.7.1`.
The CVE linked to #3590 has also now been updated to reflect the backport of that particular fix.

uap-universe mentioned this issue on Oct 19

Update dependency to jackson library due to CVE-2022-42003 and CVE-2022-42004
auth0/java-jwt#624

chadlwilson commented on Oct 19

And https://nvd.nist.gov/vuln/detail/CVE-2022-42004 is now updated - this one was much quicker!

👍 1

---

cowtowncoder commented on Oct 22   Member   Author

Big thank you **@chadlwilson**! I appreciate this and I am sure everyone with a Jackson dependency & sec scanning system does so too. :)

👍 3

🔗 ⊙ **github-actions** (bot) mentioned this issue on Oct 24

Denial of Service (DoS) SNYK-JAVA-COMFASTERXMLJACKSONCORE-3038424 RADAR-base/radar-app-config#31

✓ Closed

---

jensborrmann commented 23 days ago

Is it also possible to have the information in mvnrepository (e.g. https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind/2.12.7.1) fixed?

---

chadlwilson commented 23 days ago

@jensborrmann I've no idea how mvnrepository gets CVE data, links CPEs to artifacts or keeps it in sync. (Or don't as the case may be?) As convenient as it is, mvnrepository is to my knowledge a closed source and commercial operation which publishes no information about how it works/data sources so as a non-community initiative I've not much personal interest spending time helping fix potentially proprietary data (could go away any day, so not in the public good).

There is a contact us link if you want to try your lucking sending an email though...

---

jensborrmann commented 23 days ago

@chadlwilson : Thanks for the quick reply!

I reached out to mvnrepository and will let you know if I will find out something of relevance.

👍 1

---

cxronen mentioned this issue 23 days ago

CVE-2022-42004 @ Maven-com.fasterxml.jackson.core:jackson-databind-2.9.10.7
cxronen/AST_BookStore#245

⊙ Open

---

adakeles mentioned this issue 17 days ago

Match compile and runtime versions of jackson library dependency. auth0/java-jwt#639

⊙ Open

---

sonnyhcl mentioned this issue 11 days ago

[GHSA-rgv9-q543-rqg4] Uncontrolled Resource Consumption in FasterXML jackson-databind github/advisory-database#825

⑁ Merged

Assignees

No one assigned

Labels

2.14   **CVE**

Projects

None yet

Milestone

2.13.4

Development

No branches or pull requests

---

7 participants

/ participants