New issue

Jump to bottom

A list of bugs found (33 bugs in total) #561

Closed **ZanderHuang** opened this issue on Dec 10, 2021 · 16 comments

Labels good-first-issue image-queue

ZanderHuang commented on Dec 10, 2021 • edited

1. Unique Bugs Found

Recently we (Zhang Cen, Huang Wenjie and Zhang Xiaohan) discovered a series of bugs in latest metadta-extractor (2.16.0).
Every bug we reported in the following is unique and reproducable. We sorted and refined them from thousands of crashes.
Furthermore, they have been manually analyzed and triaged in removing the duplicates.

Due to the lack of contextual knowledge in the metadta-extractor library, we cannot thoroughly fix some bugs hence we look forward to any proposed plan from the developers in fixing these bugs.

2. Bug Report and Crash Seeds

The bug report folder can be downloaded from
https://drive.google.com/drive/folders/17UpSofkqh1KV1L5yGWzRFOT37LAJgliM?usp=sharing
It contains both reports and crash seeds.

3. Test Program to Reproduce Crashes

The test program can be downloaded from https://drive.google.com/file/d/1TfMaxAUyjuQIwXfQzHT-xUN7f25piqWt/view?usp=sharing

Total 33 bugs are reported in this pull request.
A full list is provided below.

4. Folder Structure

- Level 1 (folder): exception type
- Level 2 (folder): error location
- Level 3 (files): POC file and **report.txt** including reproducing steps

5. report.txt content:

1. Exception type
2. Error location
3. Bug cause and impact
4. Crash thread's stacks
5. Steps to reproduce

6. Bug Full List

metadata-extractor_reported_crashes

├── java.lang.ArithmeticException
│ └── com.drew.metadata.exif.PanasonicRawDistortionDescriptor.getDistortionScaleDescription--PanasonicRawDistortionDescriptor.java-97

├── java.lang.ArrayIndexOutOfBoundsException
│ ├── com.drew.metadata.exif.ExifDescriptorBase.formatCFAPattern--ExifDescriptorBase.java-586
│ └── com.drew.metadata.mp3.Mp3Reader.extract--Mp3Reader.java-96

├── java.lang.IllegalArgumentException
│ ├── com.drew.lang.StreamReader.skip--StreamReader.java-95
│ ├── com.drew.metadata.mov.atoms.FileTypeCompatibilityAtom.--FileTypeCompatibilityAtom.java-46
│ ├── com.drew.metadata.mov.atoms.SampleDescriptionAtom.--SampleDescriptionAtom.java-44
│ └── com.drew.metadata.mov.atoms.TimeToSampleAtom.--TimeToSampleAtom.java-44

├── java.lang.IndexOutOfBoundsException
│ ├── com.drew.metadata.mov.atoms.SoundSampleDescriptionAtom.addMetadata--SoundSampleDescriptionAtom.java-49
│ ├── com.drew.metadata.mov.atoms.SubtitleSampleDescriptionAtom.addMetadata--SubtitleSampleDescriptionAtom.java-75
│ ├── com.drew.metadata.mov.atoms.TextSampleDescriptionAtom.addMetadata--TextSampleDescriptionAtom.java-48
│ ├── com.drew.metadata.mov.atoms.TimecodeSampleDescriptionAtom.addMetadata--TimecodeSampleDescriptionAtom.java-48
│ ├── com.drew.metadata.mov.atoms.TimeToSampleAtom.addMetadata--TimeToSampleAtom.java-64
│ ├── com.drew.metadata.mov.atoms.VideoSampleDescriptionAtom.addMetadata--VideoSampleDescriptionAtom.java-49
│ └── com.drew.metadata.mov.metadata.QuickTimeDataHandler.processData--QuickTimeDataHandler.java-105

```
├── java.lang.NegativeArraySizeException
│   ├── com.drew.lang.SequentialByteArrayReader.getBytes--SequentialByteArrayReader.java-77
│   ├── com.drew.lang.SequentialReader.getNullTerminatedBytes--SequentialReader.java-374
│   └── com.drew.lang.StreamReader.getBytes--StreamReader.java-71

├── java.lang.NullPointerException
│   ├── com.drew.imaging.quicktime.QuickTimeHandler.addError--QuickTimeHandler.java-63
│   ├── com.drew.metadata.Directory.setString--Directory.java-287
│   ├── com.drew.metadata.Metadata.getFirstDirectoryOfType--Metadata.java-101
│   ├── com.drew.metadata.mov.atoms.SubtitleSampleDescriptionAtom.addMetadata--
SubtitleSampleDescriptionAtom.java-77
│   ├── com.drew.metadata.mov.atoms.TimeToSampleAtom.addMetadata--TimeToSampleAtom.java-64
│   ├── com.drew.metadata.mov.media.QuickTimeSoundHandler.processTimeToSample--
QuickTimeSoundHandler.java-73
│   ├── com.drew.metadata.mov.metadata.QuickTimeDirectoryHandler.processData--
QuickTimeDirectoryHandler.java-88
│   ├── com.drew.metadata.mov.QuickTimeMediaHandler.--QuickTimeMediaHandler.java-48
│   ├── com.drew.metadata.mp4.boxes.TimeToSampleBox.addMetadata--TimeToSampleBox.java-58
│   └── com.drew.metadata.mp4.boxes.TimeToSampleBox.addMetadata--TimeToSampleBox.java-65

├── java.lang.OutOfMemoryError
│   ├── com.drew.lang.SequentialByteArrayReader.getBytes--SequentialByteArrayReader.java-77
│   ├── com.drew.lang.SequentialReader.getNullTerminatedBytes--SequentialReader.java-374
│   └── com.drew.lang.StreamReader.getBytes--StreamReader.java-71

└── java.lang.StringIndexOutOfBoundsException
    ├── com.drew.metadata.exif.ExifTiffHandler.processPrintIM--ExifTiffHandler.java-733
    ├── com.drew.metadata.icc.IccDescriptor.getTagDataString--IccDescriptor.java-196
    └── com.drew.metadata.icc.IccDescriptor.getTagDataString--IccDescriptor.java-94
```

Any further discussion for these vulnerabilities including fix is welcomed and look forward to hearing from you.
Feel free to contact me at wenjiezander@gmail.com

drewnoakes commented on Dec 13, 2021 — Owner

Thank you for sharing these.

Are you willing and able to donate the repro images to the library's public regression test suite?

Most of these look like very simple fixes, just needing some additional tests on value ranges. We would welcome pull requests.

ZanderHuang commented on Dec 16, 2021 • edited — Author

Hi **@drewnoakes**, glad it helps.
You can find the details of each bug in the list documented in respective report.txt in
https://drive.google.com/drive/folders/17UpSofkqh1KV1L5yGWzRFOT37LAJgliM?usp=sharing

Each sub-folder contains a report.txt and it describes reproducing steps, crash stacks and other information.

This is one of the report.txt files for your reference:
https://drive.google.com/file/d/1bh4lptZ6aH-fgQIStfFw4p8cdaD2HmVd/view?usp=sharing

drewnoakes commented on Dec 19, 2021 Owner

@ZanderHuang I am particularly interested in knowing the answer to:

> Are you willing and able to donate the repro images to the library's public regression test suite?

Where did these images come from, and are you willing and able to make the public? We are careful to only add images to the test data set when consent is given. We use these images for regression testing, and they are very valuable in preventing regressions.

ZanderHuang commented on Dec 30, 2021 Author

@drewnoakes These bugs are classified into different exception and error types. In each folder, there is poc.tar.gz which contains all the poc files (images) for triggering that kind of bugs.
https://drive.google.com/drive/folders/17UpSofkqh1KV1L5yGWzRFOT37LAJgliM?usp=sharing

For an example, this is from java.lang.ArithmeticException > com.drew.metadata.exif.PanasonicRawDistortionDescriptor.getDistortionScaleDescription--PanasonicRawDistortionDescriptor.java-97 (https://drive.google.com/drive/folders/1jd7rbENrI6EG1FQZXExkh4f2RxQo6Lxb?usp=sharing) - download and unzip the poc.tar.gz in this link to get the poc image files.

drewnoakes commented on Dec 30, 2021 Owner

@ZanderHuang thanks, but you still haven't explained where these images came from. I am concerned that we do not have permission to use them.

Han0nly commented on Jan 26 • edited

Hi **@drewnoakes** , I'm the collaborator of **@ZanderHuang**. To some extent, we can say we created these images. The whole process for creating these inputs is as follows:

1. These images were originally gotten from GitHub fuzz corpus repositories, mostly from this.

2. We use a technique called Fuzzing to randomly mutate these samples and observe how the software performs.
3. For those samples that performed abnormally, we performed manual analysis and triaged these problems.

We are willing to donate the repro images to the library's public regression test suite.

1

Han0nly commented on Jan 27

Any updates on these bugs?

drewnoakes added good-first-issue image-queue labels on Jan 29

drewnoakes commented on Jan 29 Owner

> were originally gotten from GitHub fuzz corpus ... randomly mutate these samples

Thanks for explaining. I am not 100% clear on whether these derived works are free from the original creator's constraints. I'll ask on their repo whether they are able to share them.

> Any updates on these bugs?

I'm not sure what update you'd like. I think all users of and contributors to this library appreciate the effort here, but none have stepped up to submit fixes. I suspect most of these fixes would be very small and easy to make, so I've labelled this as a "good first issue".

1

drewnoakes mentioned this issue on Jan 29

Use of derived works in the Metadata Extractor project dvyukov/go-fuzz-corpus#11

Closed

drewnoakes commented on Jan 30 Owner

The upstream project is Apache-2.0, so the fuzzed files seem appropriate for adding to https://github.com/drewnoakes/metadata-extractor-images, which would be the first step. We could then run the files, see the errors, and set about fixing them in both the Java and .NET implementations, as needed.

1

snoopysecurity commented on Feb 25

Why does this github issue have the CVE-2022-24614 (https://nvd.nist.gov/vuln/detail/CVE-2022-24614) assigned to it? these are bugs right, not security issues?

👍 4

This was referenced on Feb 25

CVE-2022-24614 (Medium) detected in metadata-extractor-2.10.1.jar samq-ghdemo/SEARCH-NCJIS-nibrs#259

Open

CVE-2022-24614 (Medium) detected in metadata-extractor-2.10.1.jar snowdensb/nibrs#335

Open

CVE-2022-24614 (Medium) detected in metadata-extractor-2.13.0.jar snowdensb/nifi#200

Open

Han0nly commented on Mar 18

We had a discussion several days before for the CVEs of DoS bugs of java libraries. We think applications built upon these java libraries can be vulnerable to DoS attacks. Hope this can be a reasonable explanation.

Thank you very much for pointing out your concern (**@snoopysecurity**), which indeed helps the community!

 **luocooong** mentioned this issue on Mar 18

DRILL-8164: Upgrade metadata-extractor because of CVE-2022-24613 apache/drill#2493

Merged

Nadahar commented on Mar 28 Contributor

The current "trend" seems to be that almost any bug can be considered a "security issue" since they can somehow be used to degrade the application, via DoS for example. To me this is a very bad situation, where CVE's are approaching something useless that I will just ignore. It's a problem though, because among them there will be "real" issues.

I think a start would be to classify things somewhat, for example a lot of problems that are relevant for web services aren't relevant under other circumstances (DoS, SQL injection etc). These should then only apply to applications that does this.

The way it is now, everything is being "stained" with a lot of CVE's and there's almost a "panic-like attitude" among those that don't understand this game. I think the current direction is a dead-end, it won't solve security issues - on the contrary, it will make a lot of people to fed up to care at all.

👍 3

cniles commented on Mar 30 • edited ▾ Contributor

I took a look into the OutOfMemory exceptions. Looks to have been the result of integer overflows. I've resolved some problems in SequentialByteArrayReader and StreamReader and opened a PR:

#570

edit:
also #571

CC **@drewnoakes** hopefully resolves CVE.

❤️ 1

drewnoakes added a commit to drewnoakes/metadata-extractor-images that referenced this issue on Apr 3

`Basedline fuzzed files from drewnoakes/metadata-extractor#561` `779dc52`

drewnoakes added a commit that referenced this issue on Apr 3

`Avoid incorrect negative value` ··· `ddc17c0`

drewnoakes added a commit that referenced this issue on Apr 3

`Include .fuzzed files when scanning files` ··· `d723003`

drewnoakes added a commit that referenced this issue on Apr 3

`Prevent NPE` ··· `b2e2dca`

drewnoakes added a commit that referenced this issue on Apr 3

`Validate atom sizes` ··· `e61968a`

drewnoakes added a commit that referenced this issue on Apr 3

 `Method must create an instance` ··· `6215a8d`

drewnoakes added a commit that referenced this issue on Apr 3

 `Handle IOException in RIFF files` ··· `998e052`

drewnoakes added a commit that referenced this issue on Apr 3

 `Handle invalid RIFF chunk sizes` ··· `7ae0389`

drewnoakes added a commit that referenced this issue on Apr 3

 `Handle MOV sample descriptions entry count bounds` ··· `537132e`

drewnoakes added a commit that referenced this issue on Apr 3

 `Handle MOV time to sample entry count bounds` ··· `71b6b35`

drewnoakes added a commit that referenced this issue on Apr 3

 `Prevent MOV time to sample null and bounds errors` ··· `085d36c`

drewnoakes added a commit that referenced this issue on Apr 3

 `Prevent MOV sound sample bounds error` ··· `09f0cd3`

drewnoakes added a commit that referenced this issue on Apr 3

 `Prevent NPE in subtitle sample description` ··· `4bce37b`

drewnoakes added a commit that referenced this issue on Apr 3

 `Handle invalid MP3 frequency index` ··· `ca0cb05`

drewnoakes added a commit that referenced this issue on Apr 3

 `Prevent exception due to null time scale` ··· `511ab14`

drewnoakes added a commit that referenced this issue on Apr 3

 `Prevent bounds exception in timecode sample handling` ··· `35710ef`

drewnoakes added a commit that referenced this issue on Apr 3

 `Prevent bounds exception in subtitle sample handling` ··· `3b83ae6`

drewnoakes added a commit that referenced this issue on Apr 3

 `Prevent bounds exception in text sample handling` ··· `65934b6`

drewnoakes added a commit that referenced this issue on Apr 3

 `Prevent bounds exception in video sample handling` ··· `b30691e`

drewnoakes added a commit that referenced this issue on Apr 3

 `Prevent bounds exception in QuickTimeDataHandler` ··· `d54596b`

drewnoakes added a commit that referenced this issue on Apr 3

 `Prevent bounds exception in BmpReader` ··· `cde596d`

drewnoakes added a commit that referenced this issue on Apr 3

 `Prevent bounds exception in ExifTiffHandler` ··· `3402b42`

drewnoakes added a commit that referenced this issue on Apr 3

 `Prevent bounds exception in QuickTimeDataHandler` ··· `053d23d`

drewnoakes added a commit that referenced this issue on Apr 3

 `Prevent bounds exception in CanonThumbnailAtom` ··· `47095a4`

drewnoakes added a commit that referenced this issue on Apr 3

 `Refactor MP4 handling` ··· `70ae244`

drewnoakes commented on Apr 3 Owner

I worked through these issues over the weekend. I don't see how these could be considered vulnerabilities in any meaningful way though. If someone has an example of how any of these issues could be used maliciously, please let me know privately.

Still, fixing these allows partial metadata to be returned when an error is detected, which is a general goal for the language — one that escaping exceptions prevent.

Thanks for the work here. I'll close this out now.

 **drewnoakes** closed this as completed on Apr 3

---

cniles commented on Apr 26 — Contributor

@drewnoakes would you be opposed to creating a release with these fixes?

drewnoakes commented on May 5 • edited — Owner

@cniles released in 2.18.0. Thanks!

This was referenced on May 13

CVE-2022-24613 (Medium) detected in metadata-extractor-2.10.1.jar samq-ghdemo/SEARCH-NCJIS-nibrs#260

Open

CVE-2022-24613 (Medium) detected in metadata-extractor-2.10.1.jar snowdensb/nibrs#336

Open

CVE-2022-24613 (Medium) detected in metadata-extractor-2.13.0.jar snowdensb/nifi#201

Open

Assignees

No one assigned

Labels

good-first-issue image-queue

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

6 participants