Code Pull requests 21 Actions Projects Wiki Security ...

New issue

Jump to bottom

# A list of bugs found #78

Closed ZanderHuang wants to merge 1 commit into `itext:develop` from `ZanderHuang:bugs_report`

Conversation 8 Commits 1 Checks 0 Files changed 1

**ZanderHuang** commented on Oct 26, 2021 • edited

## Unqiue Bugs Found

Recently we ([Zhang Cen](https://github.com/occia) , [Huang Wenjie](https://github.com/ZanderHuang) and [Zhang Xiaohan](https://github.com/Han0nly)) discovered a series of bugs in latest itextpdf (version 7.1.17). Every bug we reported in the following is unique and reproducable. Furthermore, they have been manually analyzed and triaged in removing the duplicates.
Due to the lack of contextual knowledge in the itextpdf library, we cannot thoroughly fix some bugs hence we look forward to any proposed plan from the developers in fixing these bugs.

## Bug Report

The bug report folder can be downloaded from
https://drive.google.com/drive/folders/1b38Mi8fKp05vzMbth1oiopFYNH92GWrK?usp=sharing

Total 56 bugs are reported in this pull request.
A full list is provided below.

### Folder structure

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

### report.txt content:

1. Exception type
2. Error location

3. Steps to reproduce

## Bug full list

1. java.lang.ArrayIndexOutOfBoundsException
   -- com.itextpdf.kernel.crypto.ARCFOUREncryption.encryptARCFOUR--ARCFOUREncryption.java-93
   -- com.itextpdf.kernel.crypto.securityhandler.StandardHandlerUsingStandard128.computeOwnerKey--StandardHandlerUsingStandard128.java-81
   -- com.itextpdf.kernel.pdf.PdfXrefTable.clear--PdfXrefTable.java-448
   -- com.itextpdf.kernel.pdf.PdfXrefTable.get--PdfXrefTable.java-153
   -- com.itextpdf.kernel.pdf.PdfXrefTable.initFreeReferencesList--PdfXrefTable.java-185
2. java.lang.ClassCastException
   -- com.itextpdf.kernel.crypto.securityhandler.StandardHandlerUsingStandard40.initKeyAndReadDictionary--StandardHandlerUsingStandard40.java-193
   -- com.itextpdf.kernel.pdf.PdfDocument.open--PdfDocument.java-1958
   -- com.itextpdf.kernel.pdf.PdfEncryption.readAndSetCryptoModeForStdHandler--PdfEncryption.java-531
   -- com.itextpdf.kernel.pdf.PdfEncryption.readAndSetCryptoModeForStdHandler--PdfEncryption.java-534
   -- com.itextpdf.kernel.pdf.PdfReader.readObject--PdfReader.java-1344
3. java.lang.NegativeArraySizeException
   -- com.itextpdf.kernel.pdf.PdfXrefTable.extendXref--PdfXrefTable.java-598
4. java.lang.NullPointerException
   -- com.itextpdf.kernel.crypto.securityhandler.StandardHandlerUsingStandard40.initKeyAndReadDictionary--StandardHandlerUsingStandard40.java-194
   -- com.itextpdf.kernel.crypto.securityhandler.StandardSecurityHandler.getIsoBytes--StandardSecurityHandler.java-94
   -- com.itextpdf.kernel.pdf.PdfArray.get--PdfArray.java-374
   -- com.itextpdf.kernel.pdf.PdfObjectWrapper.markObjectAsIndirect--PdfObjectWrapper.java-141
   -- com.itextpdf.kernel.pdf.PdfReader.getOriginalFileId--PdfReader.java-669
   -- com.itextpdf.kernel.pdf.PdfReader.readDecryptObj--PdfReader.java-1287
   -- com.itextpdf.kernel.pdf.PdfReader.readObject--PdfReader.java-1344
   -- com.itextpdf.kernel.pdf.PdfReader.readObjectStream--PdfReader.java-738
   -- com.itextpdf.kernel.pdf.PdfReader.readObjectStream--PdfReader.java-739
   -- com.itextpdf.kernel.pdf.PdfReader.readObjectStream--PdfReader.java-740
   -- com.itextpdf.kernel.pdf.PdfReader.readObjectStream--PdfReader.java-773
   -- com.itextpdf.kernel.pdf.PdfReader.readObjectStream--PdfReader.java-792
5. java.lang.NumberFormatException
   -- com.itextpdf.io.source.PdfTokenizer.getIntValue--PdfTokenizer.java-512
   -- com.itextpdf.io.source.PdfTokenizer.nextValidToken--PdfTokenizer.java-314
   -- com.itextpdf.io.source.PdfTokenizer.nextValidToken--PdfTokenizer.java-315
6. java.lang.OutOfMemoryError
   -- com.itextpdf.kernel.pdf.PdfReader.readStreamBytesRaw--PdfReader.java-391

com.itextpdf.io.source.ByteBuffer.append--ByteBuffer.java-110
-- com.itextpdf.io.source.PdfTokenizer.getStringValue--PdfTokenizer.java-187
-- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-341
-- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-343
-- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-361
-- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-377
-- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-413
-- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-452
-- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-469
-- com.itextpdf.io.source.PdfTokenizer.nextValidToken--PdfTokenizer.java-271
-- com.itextpdf.io.source.PdfTokenizer.nextValidToken--PdfTokenizer.java-300
-- com.itextpdf.io.source.PdfTokenizer.nextValidToken--PdfTokenizer.java-306
-- com.itextpdf.io.source.PdfTokenizer.nextValidToken--PdfTokenizer.java-314
-- com.itextpdf.io.source.RandomAccessFileOrArray.read--RandomAccessFileOrArray.java-138
-- com.itextpdf.io.util.MessageFormatUtil.format--MessageFormatUtil.java-55
-- com.itextpdf.kernel.pdf.PdfDictionary.putAll--PdfDictionary.java-333
-- com.itextpdf.kernel.pdf.PdfName.compareTo--PdfName.java-1003
-- com.itextpdf.kernel.pdf.PdfNumber.generateValue--PdfNumber.java-180
-- com.itextpdf.kernel.pdf.PdfReader.readArray--PdfReader.java-944
-- com.itextpdf.kernel.pdf.PdfReader.readDictionary--PdfReader.java-923
-- com.itextpdf.kernel.pdf.PdfReader.readObject--PdfReader.java-1336
-- com.itextpdf.kernel.pdf.PdfReader.readObject--PdfReader.java-1344
-- com.itextpdf.kernel.pdf.PdfReader.readObject--PdfReader.java-801
-- com.itextpdf.kernel.pdf.PdfReader.readObject--PdfReader.java-845
-- com.itextpdf.kernel.pdf.PdfReader.readPdfName--PdfReader.java-912
-- com.itextpdf.kernel.pdf.PdfReader.readReference--PdfReader.java-817
-- com.itextpdf.kernel.pdf.PdfReader.readReference--PdfReader.java-834

8. java.lang.StringIndexOutOfBoundsException
   -- com.itextpdf.io.source.PdfTokenizer.checkPdfHeader--PdfTokenizer.java-239

Any further discussion for these vulnerabilities including fix is welcomed and look forward to hearing from you.

`Create bugs_report.md` … ✓ `fe6bb22`

**ZanderHuang** changed the title ~~Create bugs_report.md~~ **A list of bugs found** on Nov 3, 2021

**ZanderHuang** commented on Nov 13, 2021 Author

Hi itext7 Team, any updates on the issues mentioned? **@Snipx**

handling invalid input could have been done better on iText library side.
While we don't have any updates to share at this point, we would certainly welcome any Pull Requests that attempt to fix the problems and we could collaborate on converging towards the proper fix in the scope of individual PRs.

We don't expose GitHub's *Issues* functionality at this point, hence I am closing this PR as it's more of an *Issue* rather than a *PR* but rest assured we got the message. Thanks again

👍 1

**Snipx** closed this on Nov 30, 2021

**Snipx** commented on Dec 28, 2021 Contributor

Hi **@ZanderHuang**, would you mind signing our CLA (details available at https://itextpdf.com/en/how-buy/legal/itext-contributor-license-agreement) so that we are able to use your files in our tests?

**Han0nly** commented on Jan 27 • edited ▾

Hi **@Snipx** , I'm the collaborator of **@ZanderHuang**, we are willing to donate our test cases. I'll sign the CLA document and send it to dev.intern@itextpdf.com using my Github public email address.

👍 1

**Snipx** commented on Apr 5 • edited ▾ Contributor

> java.lang.OutOfMemoryError
> -- com.itextpdf.kernel.pdf.PdfReader.readStreamBytesRaw--PdfReader.java-391
> -- com.itextpdf.kernel.pdf.PdfXrefTable.extendXref--PdfXrefTable.java-598

Has been reported as https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-24196

This has been fixed in commit `3213363`

The fix is available in 7.2.2

**Snipx** commented on Apr 5 Contributor

> -- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-341
> -- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-343
> -- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-361
> -- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-377
> -- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-413
> -- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-452
> -- com.itextpdf.io.source.PdfTokenizer.nextToken--PdfTokenizer.java-469
> -- com.itextpdf.io.source.PdfTokenizer.nextValidToken--PdfTokenizer.java-271
> -- com.itextpdf.io.source.PdfTokenizer.nextValidToken--PdfTokenizer.java-300
> -- com.itextpdf.io.source.PdfTokenizer.nextValidToken--PdfTokenizer.java-306
> -- com.itextpdf.io.source.PdfTokenizer.nextValidToken--PdfTokenizer.java-314
> -- com.itextpdf.io.source.RandomAccessFileOrArray.read--RandomAccessFileOrArray.java-138
> -- com.itextpdf.io.util.MessageFormatUtil.format--MessageFormatUtil.java-55
> -- com.itextpdf.kernel.pdf.PdfDictionary.putAll--PdfDictionary.java-333
> -- com.itextpdf.kernel.pdf.PdfName.compareTo--PdfName.java-1003
> -- com.itextpdf.kernel.pdf.PdfNumber.generateValue--PdfNumber.java-180
> -- com.itextpdf.kernel.pdf.PdfReader.readArray--PdfReader.java-944
> -- com.itextpdf.kernel.pdf.PdfReader.readDictionary--PdfReader.java-923
> -- com.itextpdf.kernel.pdf.PdfReader.readObject--PdfReader.java-1336
> -- com.itextpdf.kernel.pdf.PdfReader.readObject--PdfReader.java-1344
> -- com.itextpdf.kernel.pdf.PdfReader.readObject--PdfReader.java-801
> -- com.itextpdf.kernel.pdf.PdfReader.readObject--PdfReader.java-845
> -- com.itextpdf.kernel.pdf.PdfReader.readPdfName--PdfReader.java-912
> -- com.itextpdf.kernel.pdf.PdfReader.readReference--PdfReader.java-817
> -- com.itextpdf.kernel.pdf.PdfReader.readReference--PdfReader.java-834

Has been reported as https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-24197

This has been fixed in commit `88c9cb7`

The fix is available in 7.2.2

**Snipx** commented on Apr 5 Contributor

> java.lang.ArrayIndexOutOfBoundsException
> -- com.itextpdf.kernel.crypto.ARCFOUREncryption.encryptARCFOUR--ARCFOUREncryption.java-93
> -- com.itextpdf.kernel.crypto.securityhandler.StandardHandlerUsingStandard128.computeOwnerKey--StandardHandlerUsingStandard128.java-81
> -- com.itextpdf.kernel.pdf.PdfXrefTable.clear--PdfXrefTable.java-448
> -- com.itextpdf.kernel.pdf.PdfXrefTable.get--PdfXrefTable.java-153
> -- com.itextpdf.kernel.pdf.PdfXrefTable.initFreeReferencesList--PdfXrefTable.java-185

Has been reported as https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-24198

We don't see any description of the potential exploitation of this. The fact that the user of the library may not know that this exception can be thrown does not by itself mean that we have a vulnerability in the library and that cannot be associated with Denial of Service vulnerability in our opinion. Java is the programming language that safeguards against buffer overflow attacks and therefore nothing related to information leakage is applicable here.
We do agree that the library could have thrown a better exception but the fact that it throws ArrayIndexOutOfBoundsException is not by itself a vulnerability, and no case supporting that this behavior may be exploited to cause DoS has been presented to us.

👍 2

**Vsevolod-Bro** mentioned this pull request on Jun 19

**[FP]: CVE-2022-24198 in kernel-7.2.2.jar** jeremylong/DependencyCheck#4613

⊙ Open

**qwertysxz** commented 25 days ago • edited ▾

Hi, can check if this CVE affects iText 5 libraries as well? Because iText 5 is listed under the known affected software configurations for this CVE in NVD website.

Reviewers

No reviews

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone