

[New issue](#)[Jump to bottom](#)

# Manipulated inline images can force PyPDF2 into an infinite loop #329

 Closedsekrause opened this issue on Feb 17, 2017 · 0 comments · Fixed by [#740](#)

Labels

is-bug

nf-performance

nf-security

sekrause commented on Feb 17, 2017

Contributor

When you try to get the content stream of [this attached PDF](#), PyPDF2 will end up in an infinite loop. So this is probably a security issue because it might be possible to denial-of-service applications using PyPDF2.

The reason is that the last while-loop in [ContentStream.\\_readInlineImage](#) only terminates when it finds the `EI` token, but never actually checks if the stream has already ended. So it's as simple as adding a (broken) inline image that doesn't have an `EI` token at all, like the attached PDF.

You can see the infinite loop by running this test script with the attached PDF:

```
import sys

from PyPDF2 import PdfFileReader, PdfFileWriter
from PyPDF2.pdf import ContentStream

with open(sys.argv[1], 'rb') as f:
    pdf = PdfFileReader(f, strict=False)
    for page in pdf.pages:
        contentstream = ContentStream(page.getContents(), pdf)
        for operands, command in contentstream.operations:
            if command == b'INLINE IMAGE':
                data = operands['data']
                print(len(data))
```

I will soon prepare a pull request that fixes this issue.

 5

  sekrause mentioned this issue on Feb 17, 2017

Improved performance and security for ContentStream\_readInlineImage #331

 Closed

  sekrause mentioned this issue on Jul 28, 2017



Endless Loop When Processing Certain Large PDF with PdfFileWriter #358

 Closed

  sekrause mentioned this issue on Mar 3, 2018

Rebooting PyPDF2 Maintenance #385

 Closed

  MartinThoma added `is-bug` `nf-performance` `nf-security` labels on Apr 8

  sekrause mentioned this issue on Apr 12

Improved performance and security for ContentStream\_readInlineImage. #740

 Merged

 MartinThoma closed this as completed in [#740](#) on Apr 15

---

 **MartinThoma** pushed a commit that referenced this issue on Apr 15

 SEC/PERF: ContentStream\_readInlineImage ([#740](#)) ...

✓ d71fb3e

#### Assignees

No one assigned

#### Labels

`is-bug` `nf-performance` `nf-security`

#### Projects

None yet


## Milestone

No milestone

---

## Development

Successfully merging a pull request may close this issue.

 **Improved performance and security for ContentStream\_readInlinelImage.**  
sekrause/PyPDF2

 **Improved performance and security for ContentStream\_readInlinelImage**

---

2 participants

