My First CVE
CVE-2024-6239

2024/8/31

第八屆臺灣好厲駭成果發表會

By 潘甫翰 Sharkkcode

Self Introduction

- 姓名與暱稱：**潘甫翰** Sharkkcode
- 學歷：國立清華大學資訊安全所，碩士生
- 在臺灣好厲駭的導師 (第八屆) ：國立陽明交通大學黃世昆教授
- 2023 資安技能金盾獎鋒芒畢露獎
- GitHub
  - https://github.com/Sharkkcode
- Blog
  - https://sharkkcode.github.io/

Table Of Contents

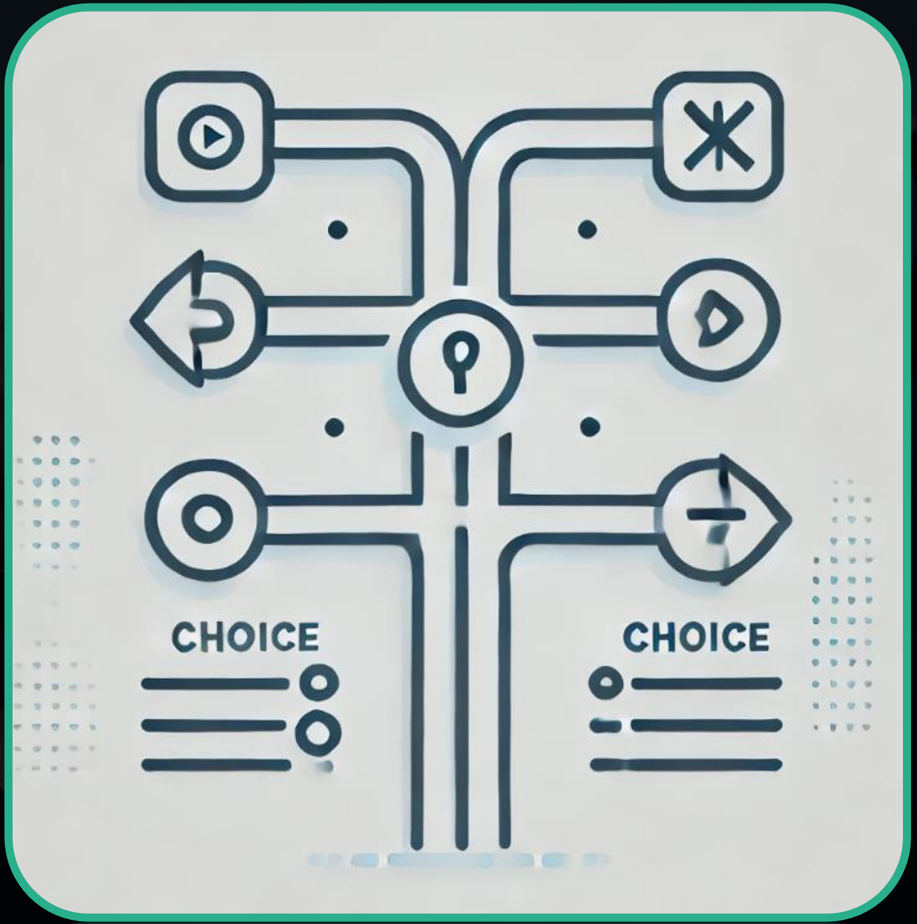
1. Selecting target
2. Narrowing the scope
3. Reading the source code
4. Fuzzing
5. Analyzing and categorizing crashes
6. Submitting issues to the author
7. Verifying patches
8. Attempting to submit CVE applications and finally obtaining a CVE

Table Of Contents

1. Selecting target
2. Narrowing the scope
3. Reading the source code
4. Fuzzing
5. Analyzing and categorizing crashes
6. Submitting issues to the author
7. Verifying patches
8. Attempting to submit CVE applications and finally obtaining a CVE

1. Selecting Target

- Not open sourced project
  - Lot of reverse engineering
  - Lots of Windows stuff
- Open sourced project
  - Easy to debug
  - Lots of Linux stuff



1. Selecting Target

- Not open sourced project
  - Lot of reverse engineering
  - Lots of Windows stuff
- **Open sourced project**
  - **Easy to debug**
  - **Lots of Linux stuff**

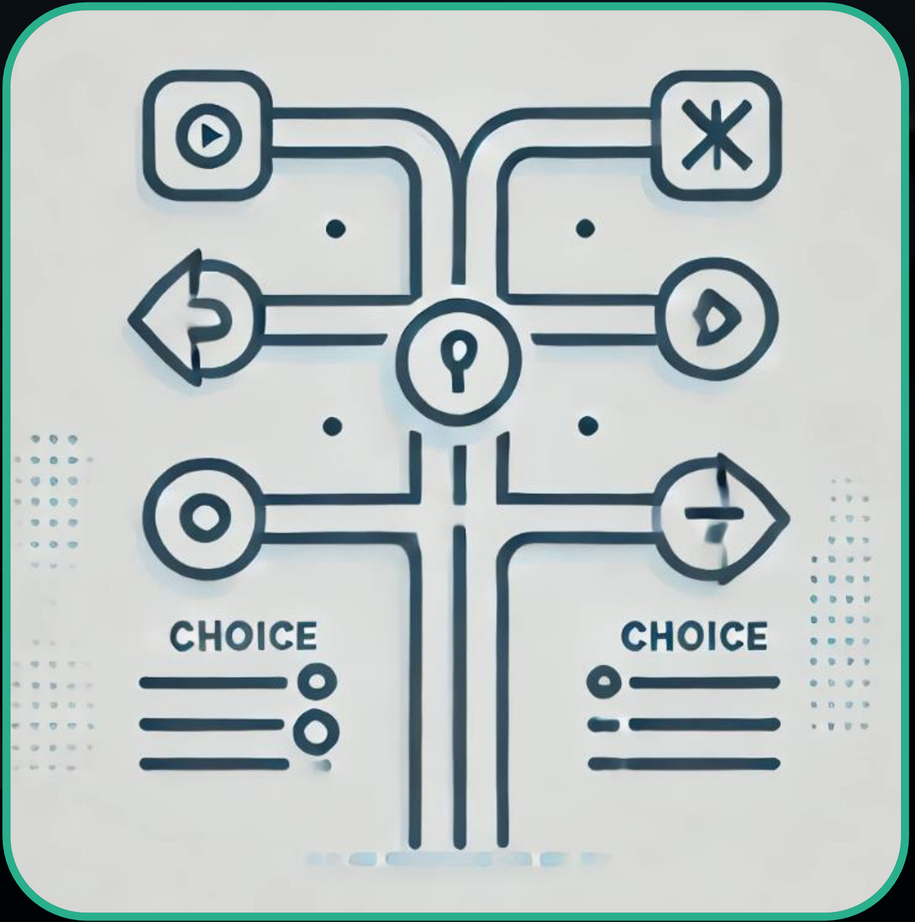
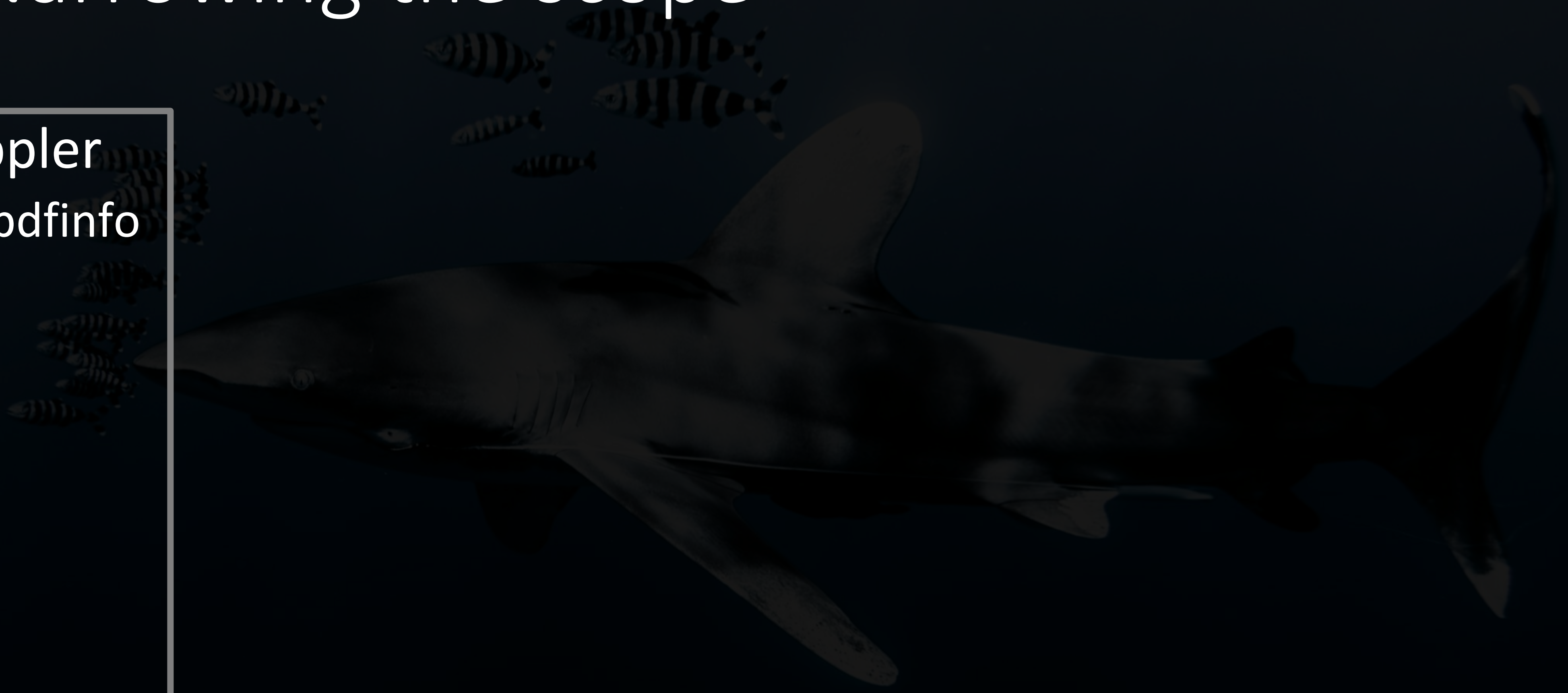


Table Of Contents

1. Selecting target
2. Narrowing the scope
3. Reading the source code
4. Fuzzing
5. Analyzing and categorizing crashes
6. Submitting issues to the author
7. Verifying patches
8. Attempting to submit CVE applications and finally obtaining a CVE

2. Narrowing the scope

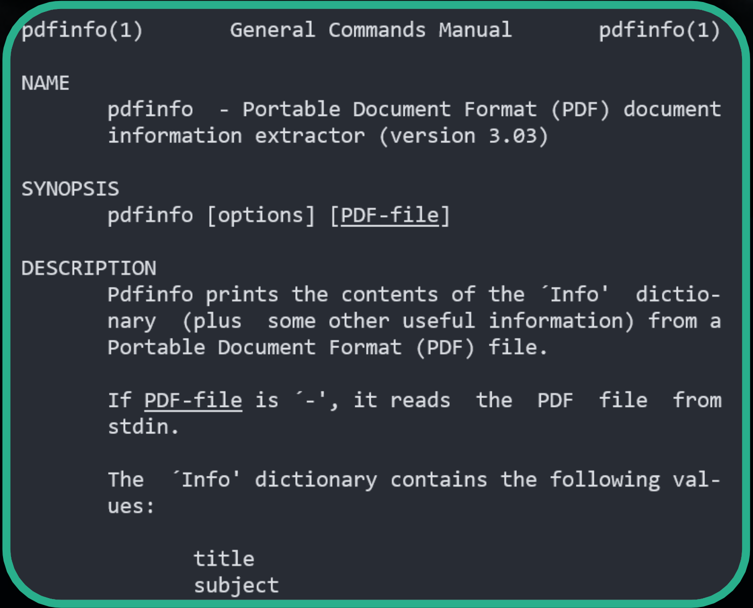
- Poppler
  - pdfinfo



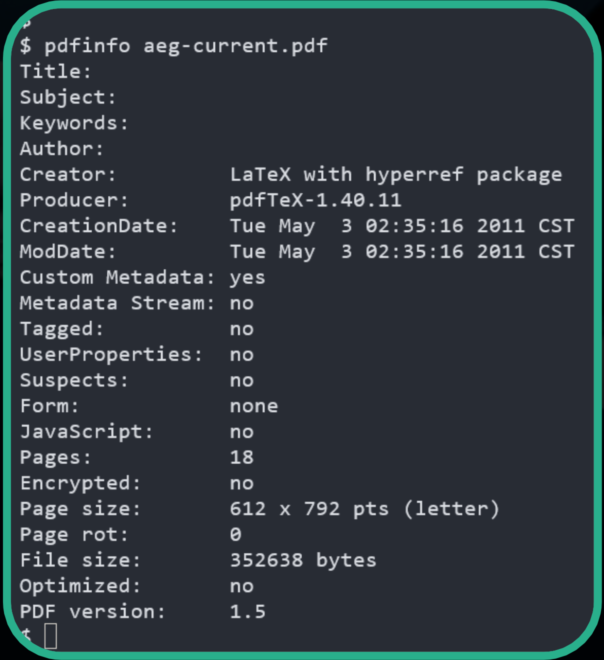
2. Narrowing the scope

- Poppler
  - pdfinfo

- pdfinfo
  - Portable Document Format (PDF) document information extractor.



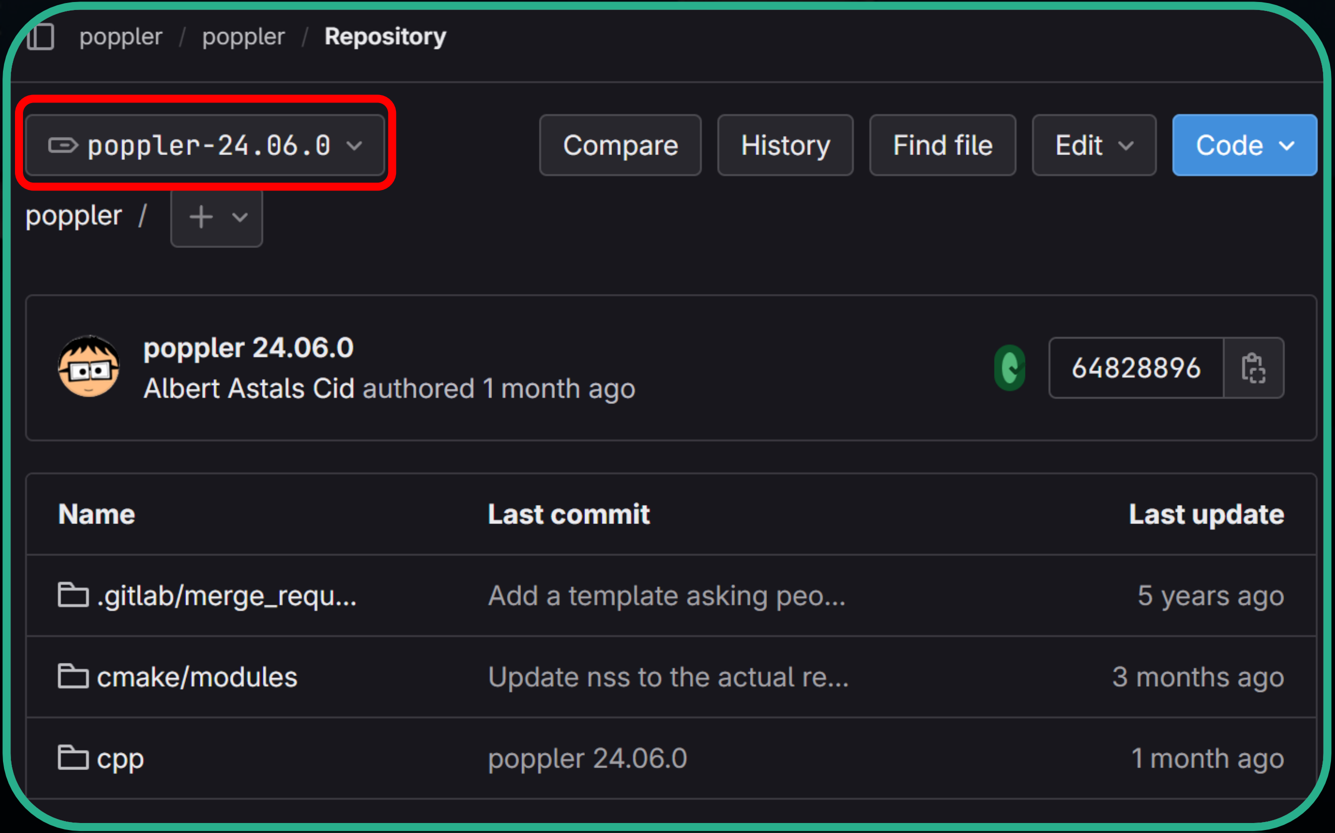
```
pdfinfo(1)         General Commands Manual         pdfinfo(1)

NAME
       pdfinfo  - Portable Document Format (PDF) document
       information extractor (version 3.03)

SYNOPSIS
       pdfinfo [options] [PDF-file]

DESCRIPTION
       Pdfinfo prints the contents of the ´Info'  dictio-
       nary  (plus  some other useful information) from a
       Portable Document Format (PDF) file.

       If PDF-file is ´-', it reads  the  PDF  file  from
       stdin.

       The  ´Info' dictionary contains the following val-
       ues:

              title
              subject
```



```
$
$ pdfinfo aeg-current.pdf
Title:
Subject:
Keywords:
Author:
Creator:         LaTeX with hyperref package
Producer:        pdfTeX-1.40.11
CreationDate:    Tue May  3 02:35:16 2011 CST
ModDate:         Tue May  3 02:35:16 2011 CST
Custom Metadata: yes
Metadata Stream: no
Tagged:          no
UserProperties:  no
Suspects:        no
Form:            none
JavaScript:      no
Pages:           18
Encrypted:       no
Page size:       612 x 792 pts (letter)
Page rot:        0
File size:       352638 bytes
Optimized:       no
PDF version:     1.5
$ 
```

2. Narrowing the scope

- poppler version : 24.06.0



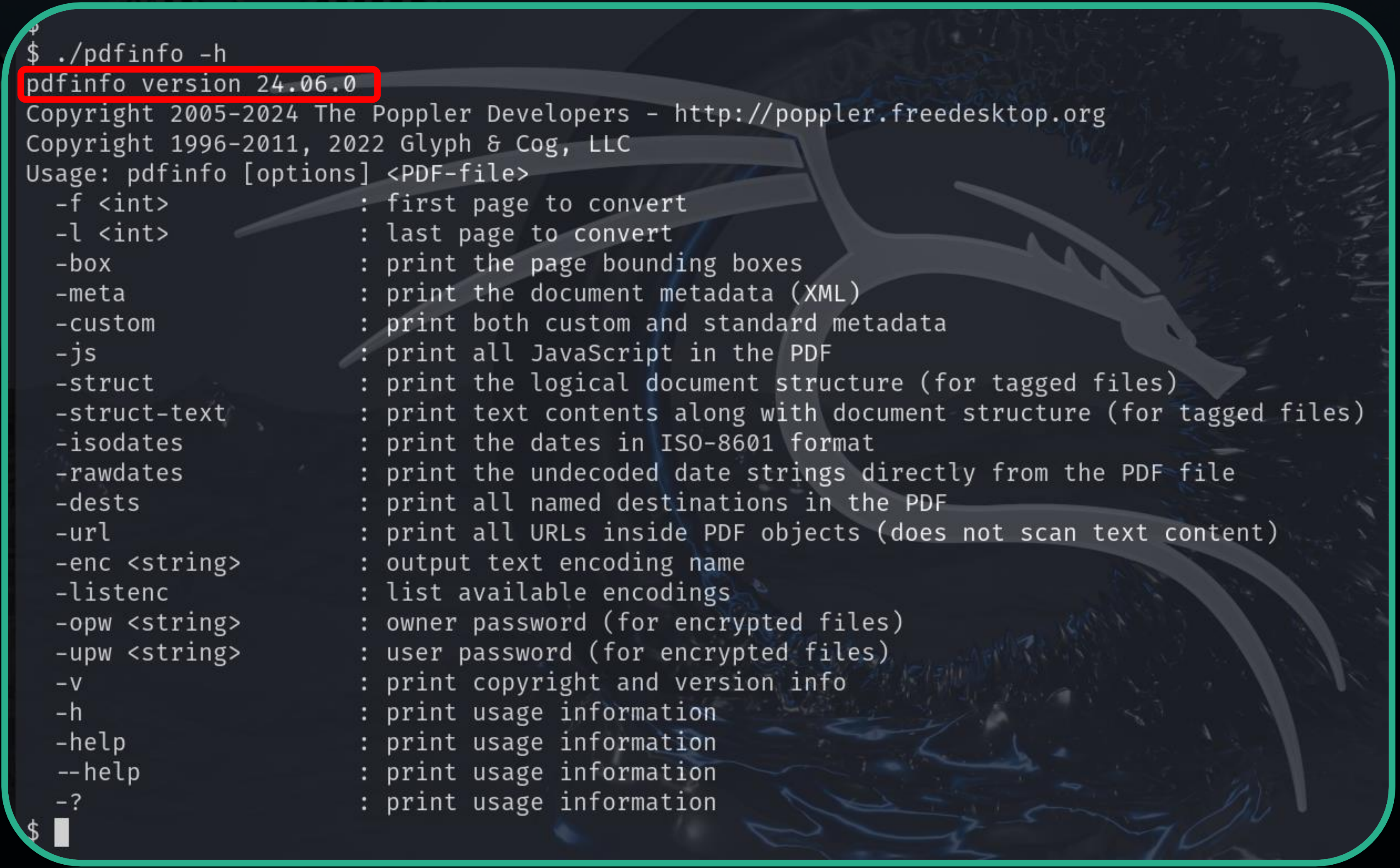
poppler 24.06.0
Albert Astals Cid authored 1 month ago

64828896

Name	Last commit	Last update
.gitlab/merge_requ...	Add a template asking peo...	5 years ago
cmake/modules	Update nss to the actual re...	3 months ago
cpp	poppler 24.06.0	1 month ago

2. Narrowing the scope

- Build and check the version



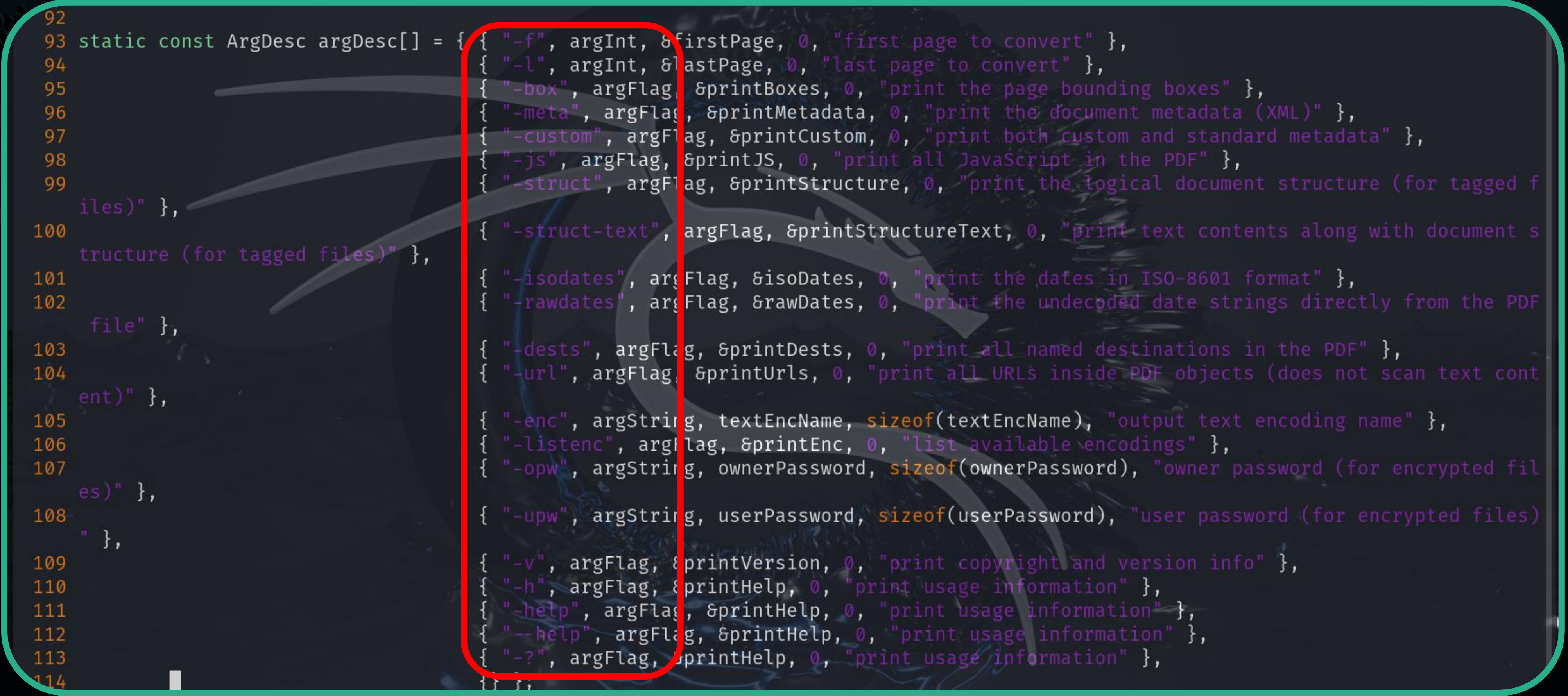
```
$ ./pdfinfo -h
pdfinfo version 24.06.0
Copyright 2005-2024 The Poppler Developers - http://poppler.freedesktop.org
Copyright 1996-2011, 2022 Glyph & Cog, LLC
Usage: pdfinfo [options] <PDF-file>
  -f <int>             : first page to convert
  -l <int>             : last page to convert
  -box                 : print the page bounding boxes
  -meta                : print the document metadata (XML)
  -custom              : print both custom and standard metadata
  -js                  : print all JavaScript in the PDF
  -struct              : print the logical document structure (for tagged files)
  -struct-text         : print text contents along with document structure (for tagged files)
  -isodates            : print the dates in ISO-8601 format
  -rawdates            : print the undecoded date strings directly from the PDF file
  -dests               : print all named destinations in the PDF
  -url                 : print all URLs inside PDF objects (does not scan text content)
  -enc <string>        : output text encoding name
  -listenc             : list available encodings
  -opw <string>        : owner password (for encrypted files)
  -upw <string>        : user password (for encrypted files)
  -v                   : print copyright and version info
  -h                   : print usage information
  -help                : print usage information
  --help               : print usage information
  -?                   : print usage information
$ 
```

Table Of Contents

1. Selecting target
2. Narrowing the scope
3. Reading the source code
4. Fuzzing
5. Analyzing and categorizing crashes
6. Submitting issues to the author
7. Verifying patches
8. Attempting to submit CVE applications and finally obtaining a CVE

3. Reading the source code

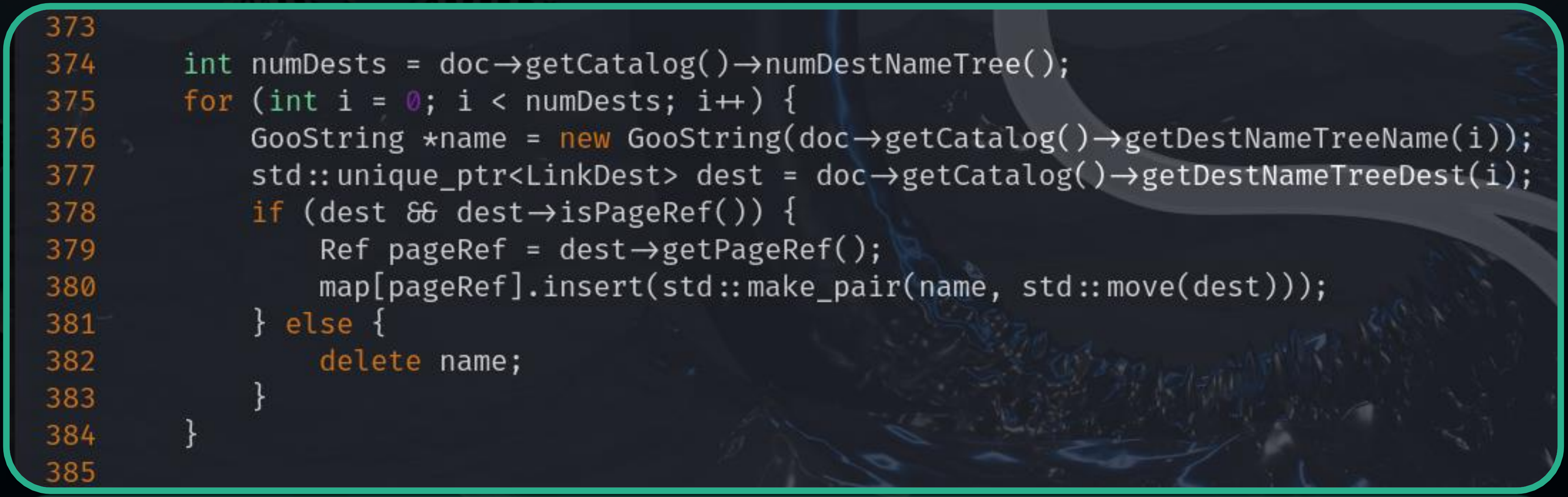
- Options overview



```
static const ArgDesc argDesc[] = { { "-f", argInt, &firstPage, 0, "first page to convert" },
                                   { "-l", argInt, &lastPage, 0, "last page to convert" },
                                   { "-box", argFlag, &printBoxes, 0, "print the page bounding boxes" },
                                   { "-meta", argFlag, &printMetadata, 0, "print the document metadata (XML)" },
                                   { "-custom", argFlag, &printCustom, 0, "print both custom and standard metadata" },
                                   { "-js", argFlag, &printJS, 0, "print all JavaScript in the PDF" },
                                   { "-struct", argFlag, &printStructure, 0, "print the logical document structure (for tagged files)" },
                                   { "-struct-text", argFlag, &printStructureText, 0, "print text contents along with document structure (for tagged files)" },
                                   { "-isodates", argFlag, &isoDates, 0, "print the dates in ISO-8601 format" },
                                   { "-rawdates", argFlag, &rawDates, 0, "print the undecoded date strings directly from the PDF file" },
                                   { "-dests", argFlag, &printDests, 0, "print all named destinations in the PDF" },
                                   { "-url", argFlag, &printUrls, 0, "print all URLs inside PDF objects (does not scan text content)" },
                                   { "-enc", argString, textEncName, sizeof(textEncName), "output text encoding name" },
                                   { "-listenc", argFlag, &printEnc, 0, "list available encodings" },
                                   { "-opw", argString, ownerPassword, sizeof(ownerPassword), "owner password (for encrypted files)" },
                                   { "-upw", argString, userPassword, sizeof(userPassword), "user password (for encrypted files)" },
                                   { "-v", argFlag, &printVersion, 0, "print copyright and version info" },
                                   { "-h", argFlag, &printHelp, 0, "print usage information" },
                                   { "-help", argFlag, &printHelp, 0, "print usage information" },
                                   { "--help", argFlag, &printHelp, 0, "print usage information" },
                                   { "-?", argFlag, &printHelp, 0, "print usage information" },
                                   {} };
```

3. Reading the source code

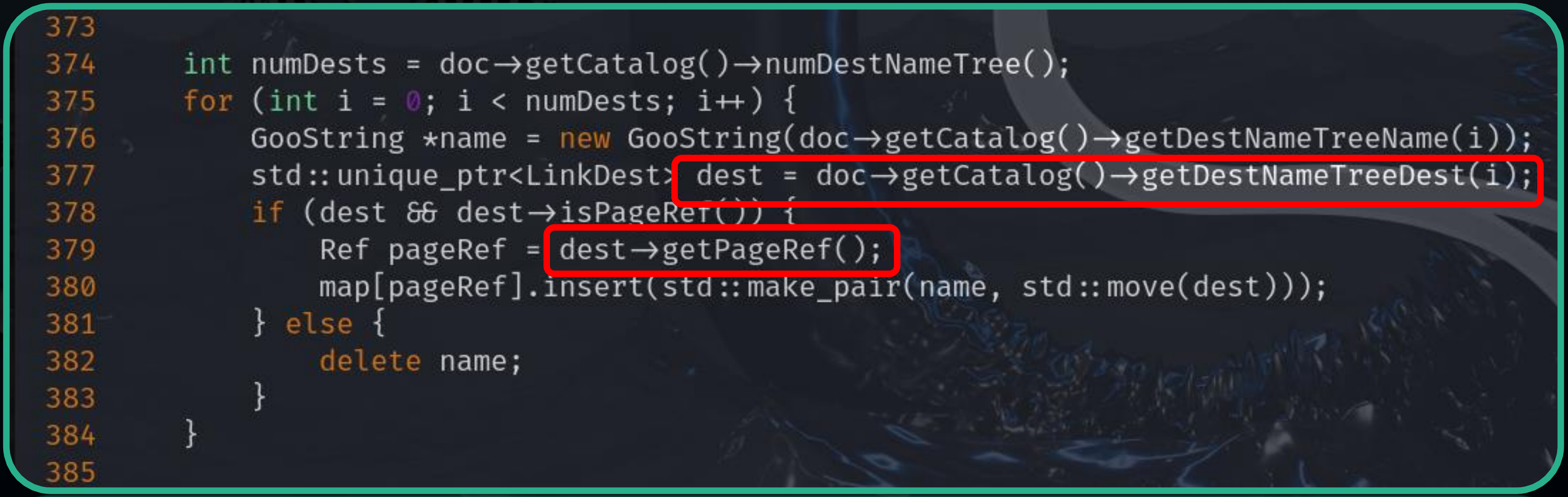
- Findings…
- Guessing…



```cpp

    int numDests = doc->getCatalog()->numDestNameTree();
    for (int i = 0; i < numDests; i++) {
        GooString *name = new GooString(doc->getCatalog()->getDestNameTreeName(i));
        std::unique_ptr<LinkDest> dest = doc->getCatalog()->getDestNameTreeDest(i);
        if (dest && dest->isPageRef()) {
            Ref pageRef = dest->getPageRef();
            map[pageRef].insert(std::make_pair(name, std::move(dest)));
        } else {
            delete name;
        }
    }

```

3. Reading the source code

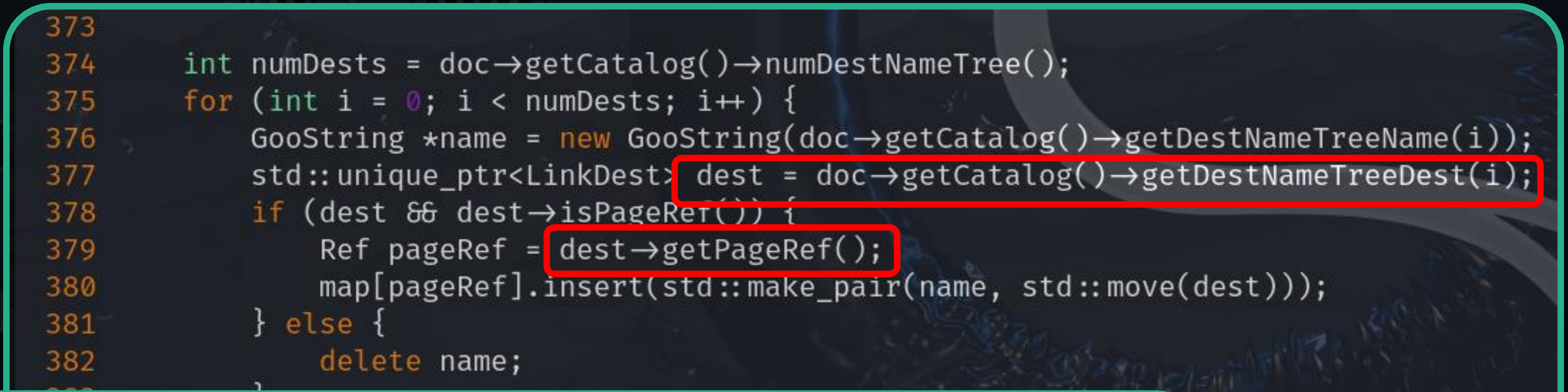
- Findings…
- Guessing…



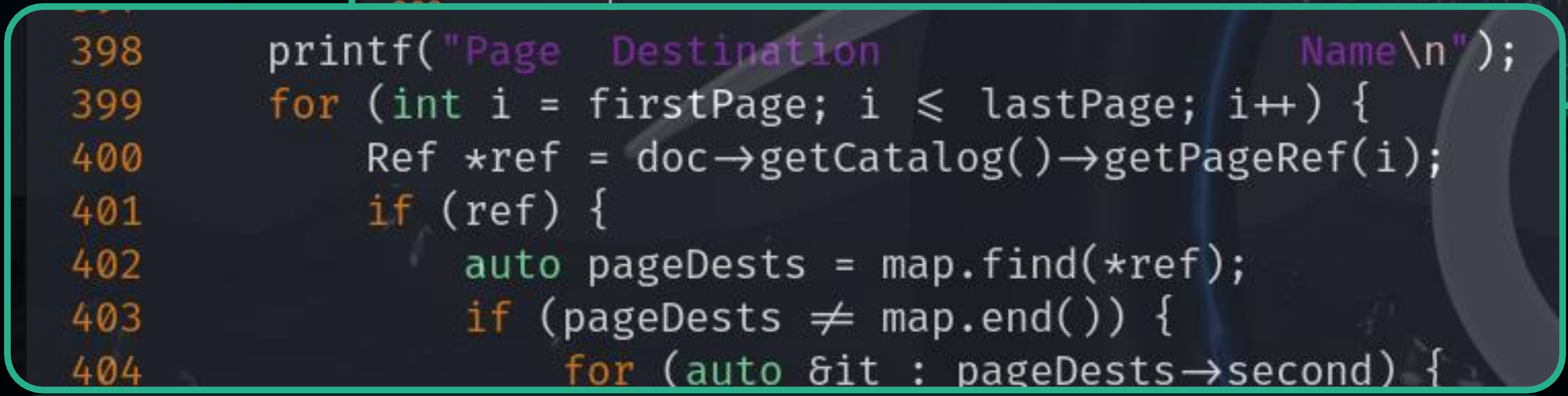
```
    int numDests = doc->getCatalog()->numDestNameTree();
    for (int i = 0; i < numDests; i++) {
        GooString *name = new GooString(doc->getCatalog()->getDestNameTreeName(i));
        std::unique_ptr<LinkDest> dest = doc->getCatalog()->getDestNameTreeDest(i);
        if (dest && dest->isPageRef()) {
            Ref pageRef = dest->getPageRef();
            map[pageRef].insert(std::make_pair(name, std::move(dest)));
        } else {
            delete name;
        }
    }
```

3. Reading the source code

- Findings…
- Guessing…



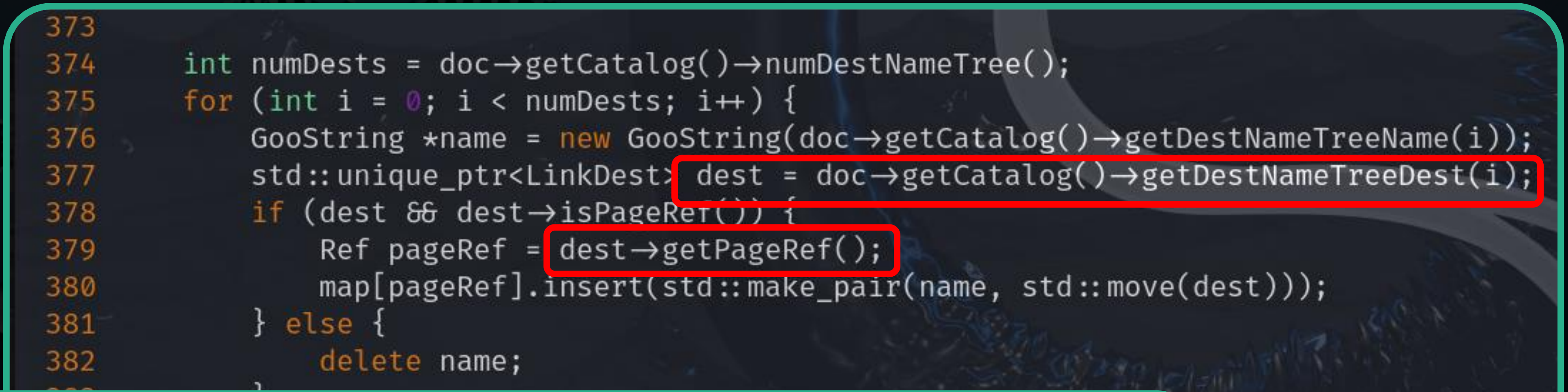
```
    int numDests = doc->getCatalog()->numDestNameTree();
    for (int i = 0; i < numDests; i++) {
        GooString *name = new GooString(doc->getCatalog()->getDestNameTreeName(i));
        std::unique_ptr<LinkDest> dest = doc->getCatalog()->getDestNameTreeDest(i);
        if (dest && dest->isPageRef()) {
            Ref pageRef = dest->getPageRef();
            map[pageRef].insert(std::make_pair(name, std::move(dest)));
        } else {
            delete name;
```



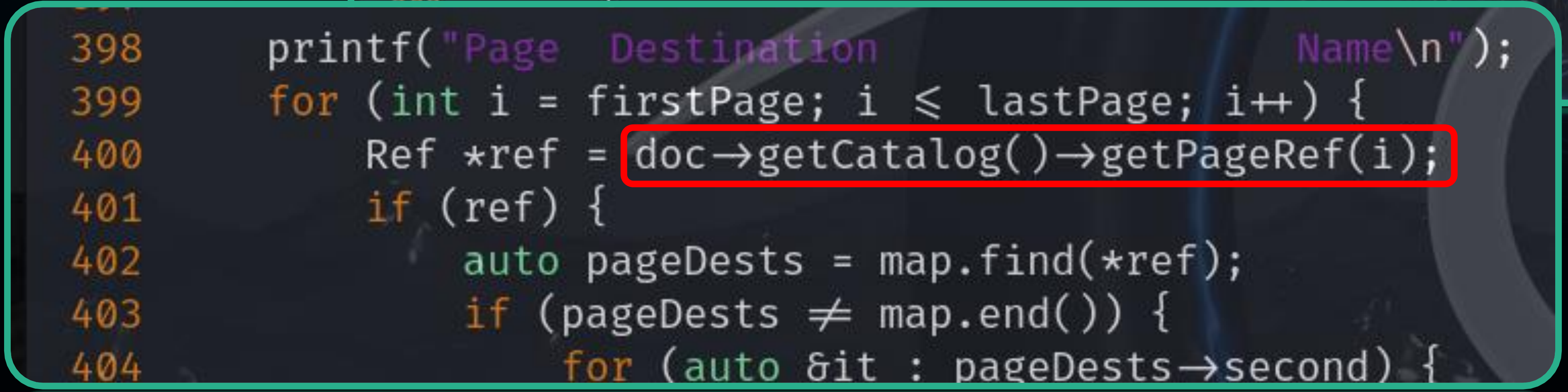
```
    printf("Page  Destination                 Name\n");
    for (int i = firstPage; i <= lastPage; i++) {
        Ref *ref = doc->getCatalog()->getPageRef(i);
        if (ref) {
            auto pageDests = map.find(*ref);
            if (pageDests != map.end()) {
                for (auto &it : pageDests->second) {
```

3. Reading the source code

- Findings…
- Guessing…



```
    int numDests = doc->getCatalog()->numDestNameTree();
    for (int i = 0; i < numDests; i++) {
        GooString *name = new GooString(doc->getCatalog()->getDestNameTreeName(i));
        std::unique_ptr<LinkDest> dest = doc->getCatalog()->getDestNameTreeDest(i);
        if (dest && dest->isPageRef()) {
            Ref pageRef = dest->getPageRef();
            map[pageRef].insert(std::make_pair(name, std::move(dest)));
        } else {
            delete name;
        }
```



```
    printf("Page  Destination                 Name\n");
    for (int i = firstPage; i <= lastPage; i++) {
        Ref *ref = doc->getCatalog()->getPageRef(i);
        if (ref) {
            auto pageDests = map.find(*ref);
            if (pageDests != map.end()) {
                for (auto &it : pageDests->second) {
```

Manually Crafting Payload

Fuzzing

~~Manually Crafting Payload~~

Fuzzing

Table Of Contents

1. Selecting target
2. Narrowing the scope
3. Reading the source code
4. Fuzzing
5. Analyzing and categorizing crashes
6. Submitting issues to the author
7. Verifying patches
8. Attempting to submit CVE applications and finally obtaining a CVE

4. Fuzzing

- The option I mainly want to fuzz
  - -dests

~~Harness 🥺~~

4. Fuzzing

- The option I mainly want to fuzz
  - -dests
- The other options to fuzz along the way
  - -box
  - -meta
  - -struct
  - -struct-text
  - -isodates
  - -rawdates

4. Fuzzing

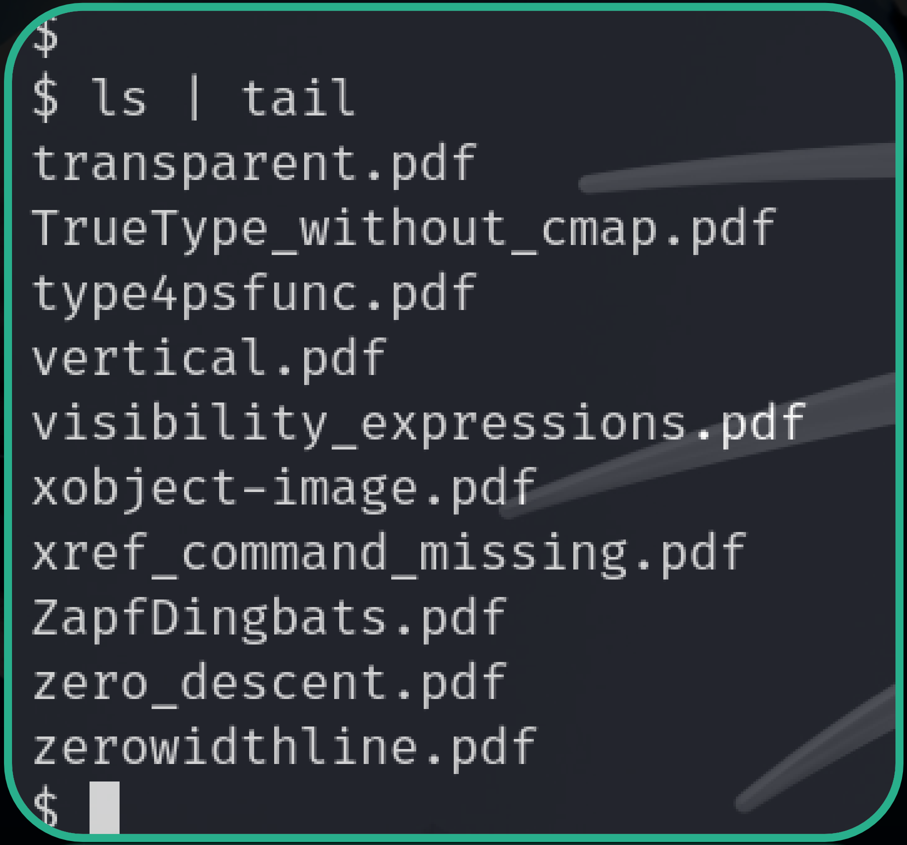
- Fuzzing approaches
  - White Box Fuzzing
  - Black Box Fuzzing

4. Fuzzing

- Fuzzing approaches
  - White Box Fuzzing
  - Black Box Fuzzing

4. Fuzzing

- Seed input
  - Finding some buggy pdf files...



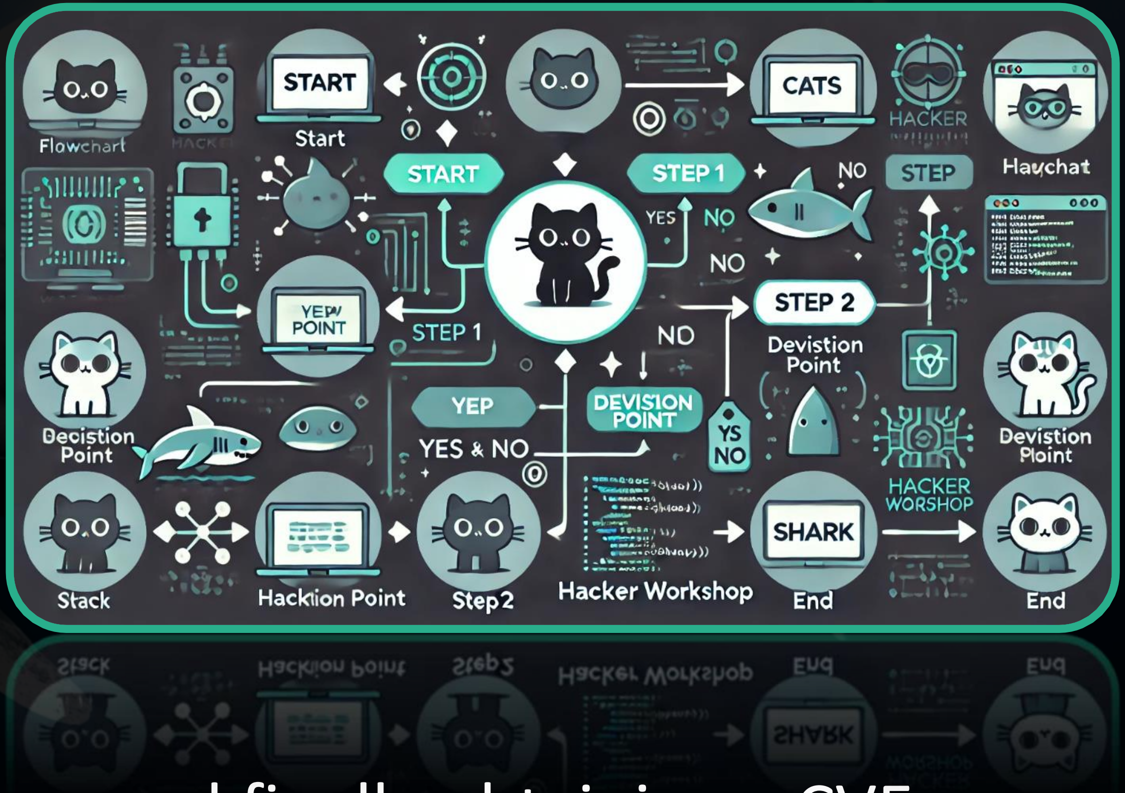
```
$
$ ls | tail
transparent.pdf
TrueType_without_cmap.pdf
type4psfunc.pdf
vertical.pdf
visibility_expressions.pdf
xobject-image.pdf
xref_command_missing.pdf
ZapfDingbats.pdf
zero_descent.pdf
zerowidthline.pdf
$ 
```

4. Fuzzing

- Command (use -dests as an example)
  - ./pdfinfo -dests seed_input.pdf

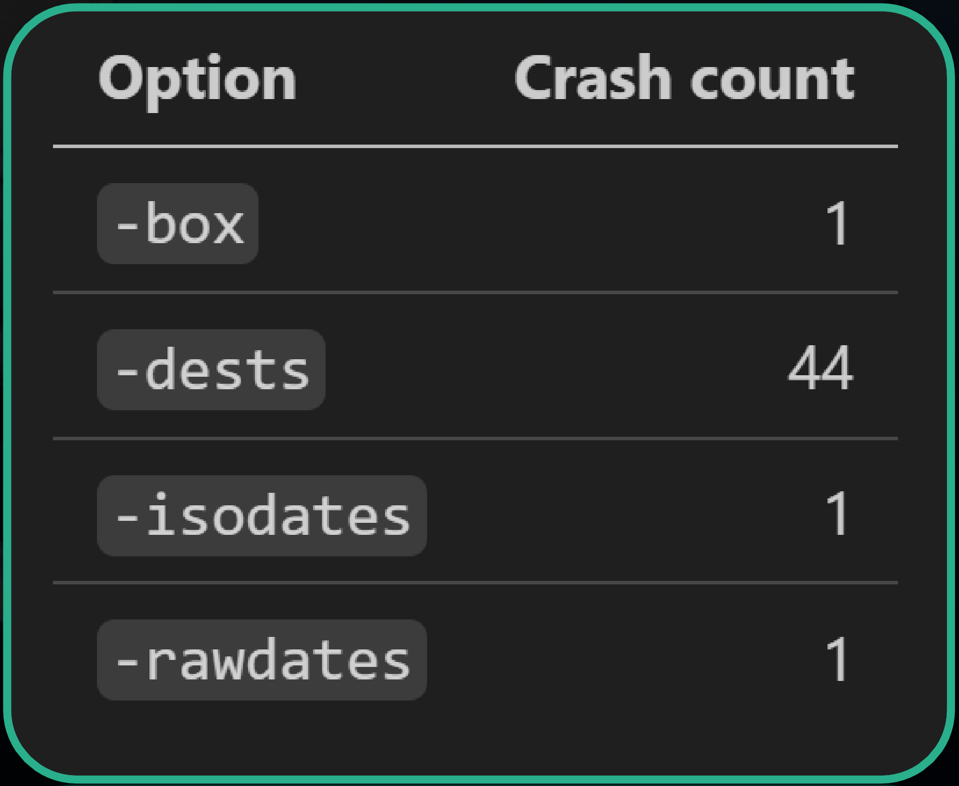
Table Of Contents

1. Selecting target
2. Narrowing the scope
3. Reading the source code
4. Fuzzing
5. Analyzing and categorizing crashes
6. Submitting issues to the author
7. Verifying patches
8. Attempting to submit CVE applications and finally obtaining a CVE



5. Analyzing and categorizing crashes

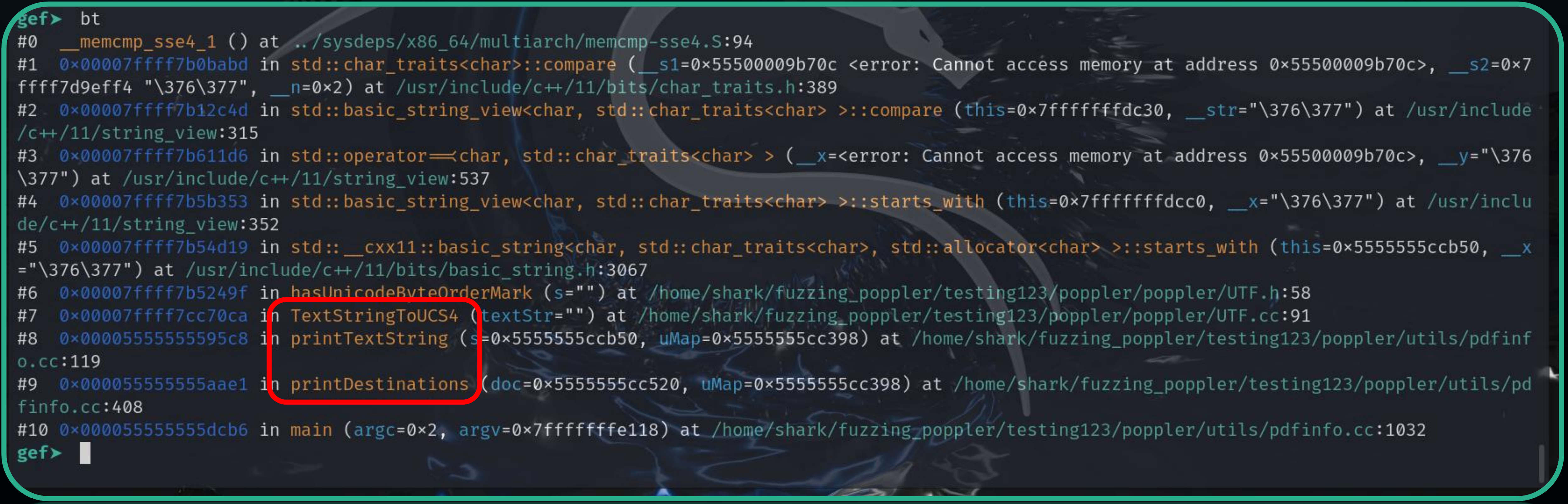
- Result
  - From the results of these fuzzing crashes, I can generally categorize the crashes into two groups, and this CVE belongs to one of those groups.



Option	Crash count
-box	1
-dests	44
-isodates	1
-rawdates	1

5. Analyzing and categorizing crashes

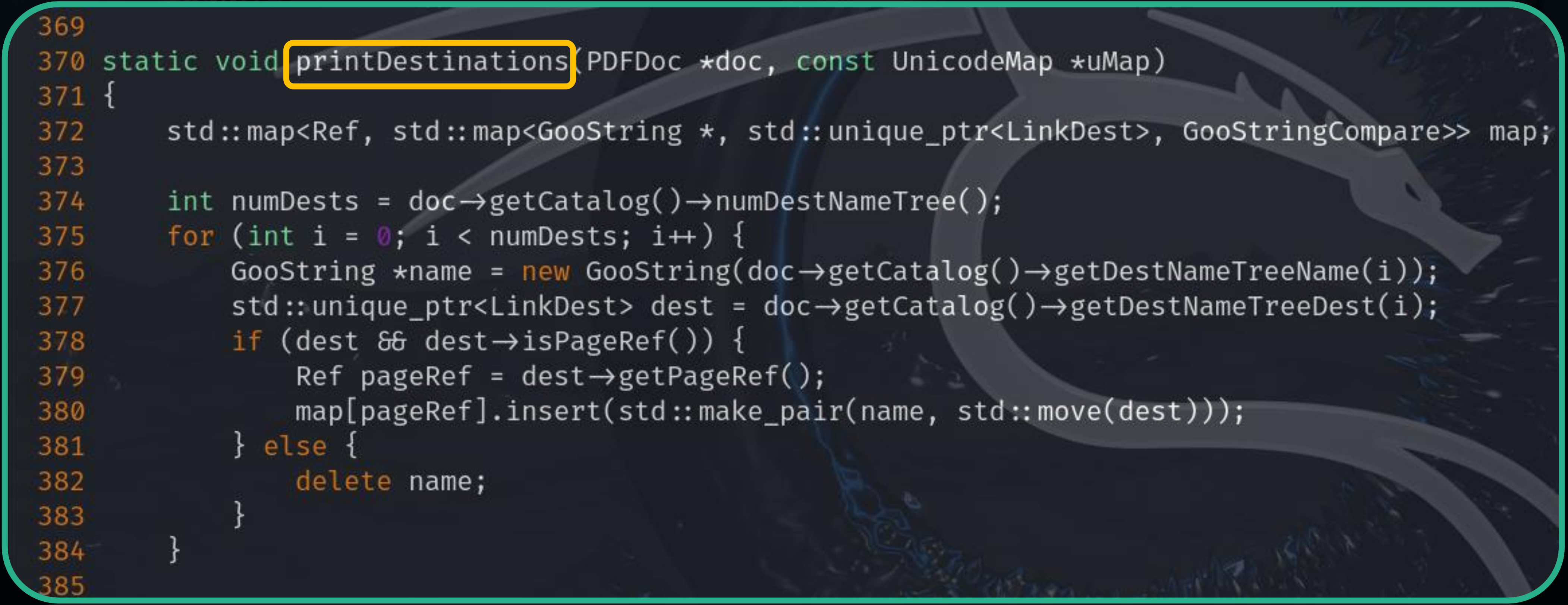
- Backtraceing



```
gef➤  bt
#0  __memcmp_sse4_1 () at ../sysdeps/x86_64/multiarch/memcmp-sse4.S:94
#1  0x00007ffff7b0babd in std::char_traits<char>::compare (__s1=0x55500009b70c <error: Cannot access memory at address 0x55500009b70c>, __s2=0x7
ffff7d9eff4 "\376\377", __n=0x2) at /usr/include/c++/11/bits/char_traits.h:389
#2  0x00007ffff7b12c4d in std::basic_string_view<char, std::char_traits<char> >::compare (this=0x7fffffffdc30, __str="\376\377") at /usr/include
/c++/11/string_view:315
#3  0x00007ffff7b611d6 in std::operator==<char, std::char_traits<char> > (__x=<error: Cannot access memory at address 0x55500009b70c>, __y="\376
\377") at /usr/include/c++/11/string_view:537
#4  0x00007ffff7b5b353 in std::basic_string_view<char, std::char_traits<char> >::starts_with (this=0x7fffffffdcc0, __x="\376\377") at /usr/inclu
de/c++/11/string_view:352
#5  0x00007ffff7b54d19 in std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::starts_with (this=0x5555555ccb50, __x
="\376\377") at /usr/include/c++/11/bits/basic_string.h:3067
#6  0x00007ffff7b5249f in hasUnicodeByteOrderMark (s="") at /home/shark/fuzzing_poppler/testing123/poppler/poppler/UTF.h:58
#7  0x00007ffff7cc70ca in TextStringToUCS4 (textStr="") at /home/shark/fuzzing_poppler/testing123/poppler/poppler/UTF.cc:91
#8  0x00005555555595c8 in printTextString (s=0x5555555ccb50, uMap=0x5555555cc398) at /home/shark/fuzzing_poppler/testing123/poppler/utils/pdfinf
o.cc:119
#9  0x000055555555aae1 in printDestinations (doc=0x5555555cc520, uMap=0x5555555cc398) at /home/shark/fuzzing_poppler/testing123/poppler/utils/pd
finfo.cc:408
#10 0x000055555555dcb6 in main (argc=0x2, argv=0x7fffffffe118) at /home/shark/fuzzing_poppler/testing123/poppler/utils/pdfinfo.cc:1032
gef➤ 
```

5. Analyzing and categorizing crashes

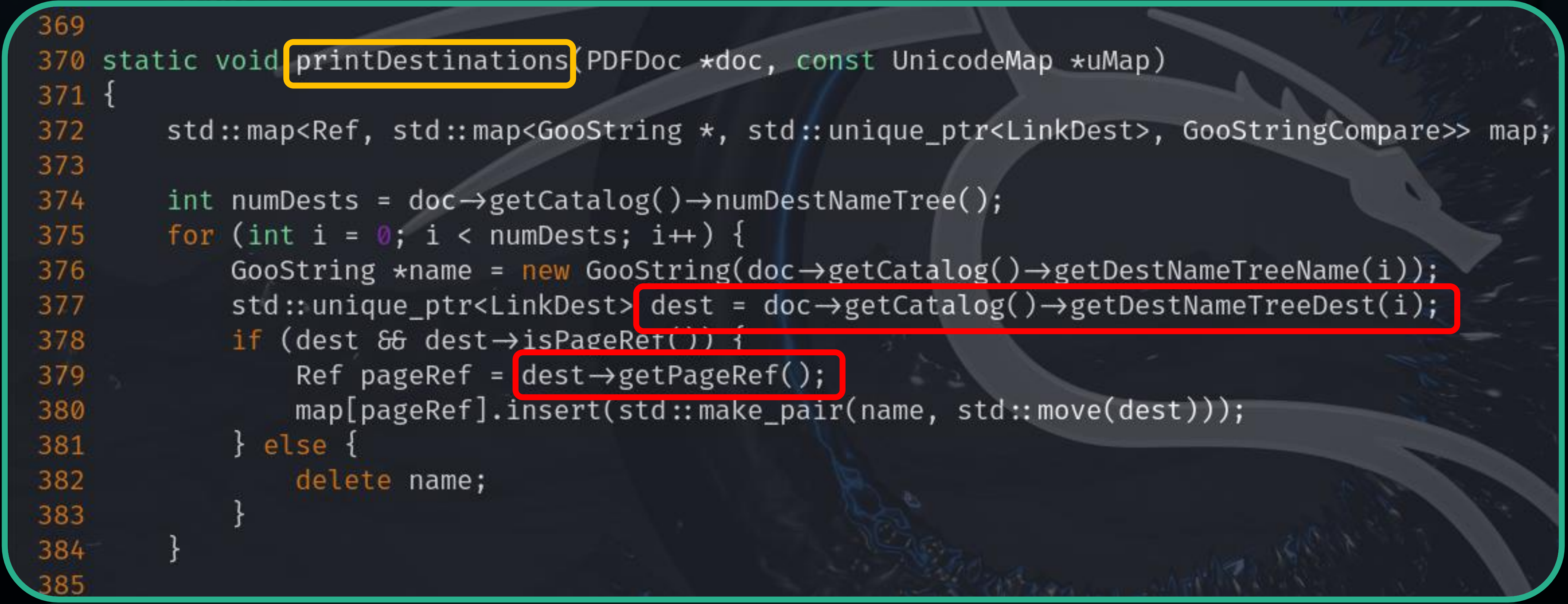
- Looking into printDestinations



```cpp

static void printDestinations(PDFDoc *doc, const UnicodeMap *uMap)
{
    std::map<Ref, std::map<GooString *, std::unique_ptr<LinkDest>, GooStringCompare>> map;

    int numDests = doc->getCatalog()->numDestNameTree();
    for (int i = 0; i < numDests; i++) {
        GooString *name = new GooString(doc->getCatalog()->getDestNameTreeName(i));
        std::unique_ptr<LinkDest> dest = doc->getCatalog()->getDestNameTreeDest(i);
        if (dest && dest->isPageRef()) {
            Ref pageRef = dest->getPageRef();
            map[pageRef].insert(std::make_pair(name, std::move(dest)));
        } else {
            delete name;
        }
    }

```

5. Analyzing and categorizing crashes

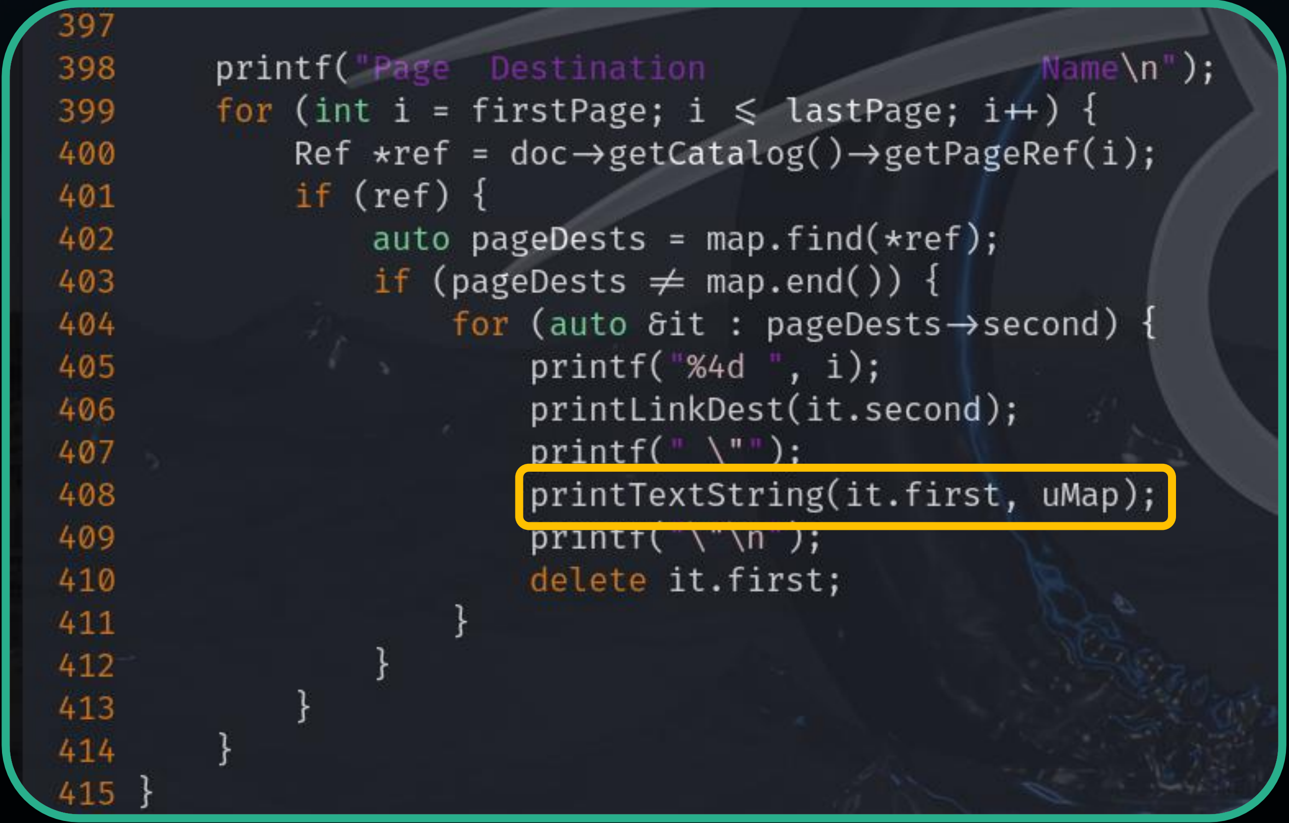
- Looking into printDestinations



```cpp

static void printDestinations(PDFDoc *doc, const UnicodeMap *uMap)
{
    std::map<Ref, std::map<GooString *, std::unique_ptr<LinkDest>, GooStringCompare>> map;

    int numDests = doc->getCatalog()->numDestNameTree();
    for (int i = 0; i < numDests; i++) {
        GooString *name = new GooString(doc->getCatalog()->getDestNameTreeName(i));
        std::unique_ptr<LinkDest> dest = doc->getCatalog()->getDestNameTreeDest(i);
        if (dest && dest->isPageRef()) {
            Ref pageRef = dest->getPageRef();
            map[pageRef].insert(std::make_pair(name, std::move(dest)));
        } else {
            delete name;
        }
    }

```

5. Analyzing and categorizing crashes

- Around printTextString

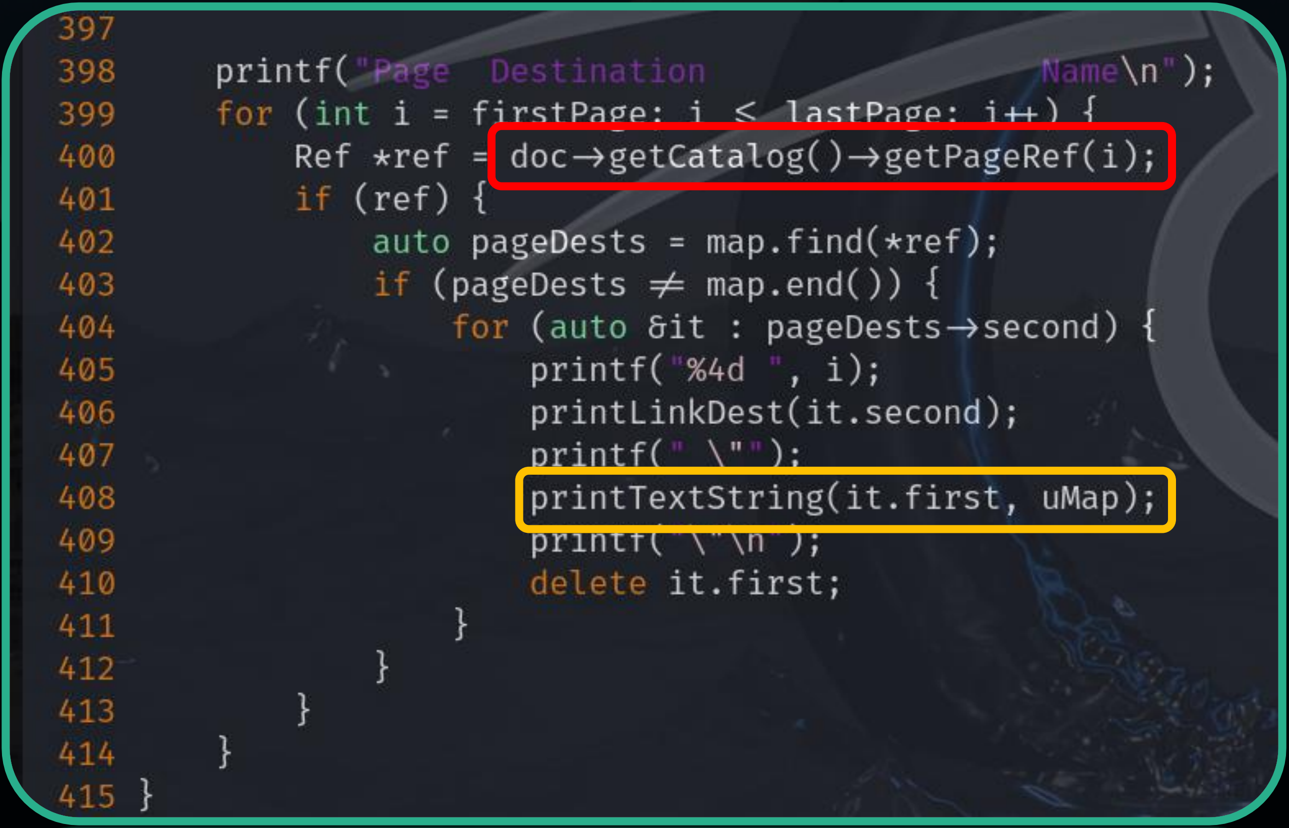


```
    printf("Page  Destination                 Name\n");
    for (int i = firstPage; i <= lastPage; i++) {
        Ref *ref = doc->getCatalog()->getPageRef(i);
        if (ref) {
            auto pageDests = map.find(*ref);
            if (pageDests != map.end()) {
                for (auto &it : pageDests->second) {
                    printf("%4d ", i);
                    printLinkDest(it.second);
                    printf(" \"");
                    printTextString(it.first, uMap);
                    printf("\"\n");
                    delete it.first;
                }
            }
        }
    }
}
```

By 潘甫翰 Sharkkcode

5. Analyzing and categorizing crashes

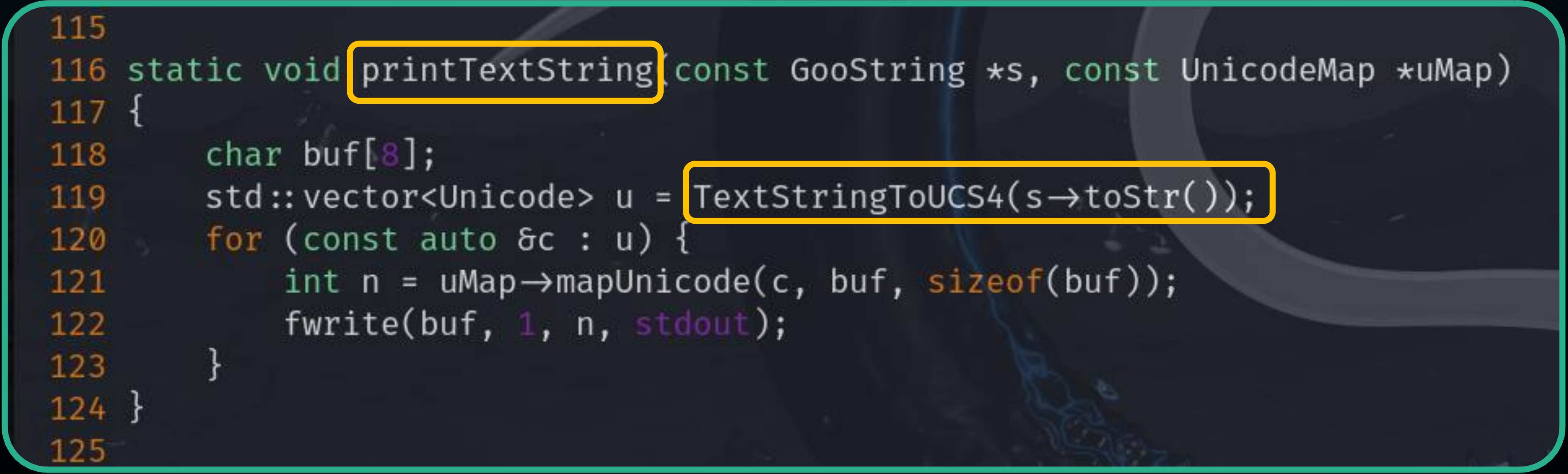
- Around printTextString



```
    printf("Page  Destination                 Name\n");
    for (int i = firstPage; i <= lastPage; i++) {
        Ref *ref = doc->getCatalog()->getPageRef(i);
        if (ref) {
            auto pageDests = map.find(*ref);
            if (pageDests != map.end()) {
                for (auto &it : pageDests->second) {
                    printf("%4d ", i);
                    printLinkDest(it.second);
                    printf(" \"");
                    printTextString(it.first, uMap);
                    printf("\"\n");
                    delete it.first;
                }
            }
        }
    }
}
```

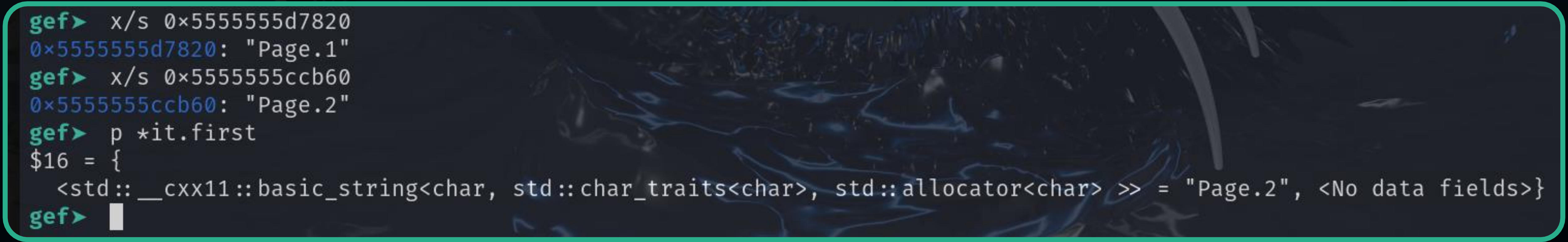
5. Analyzing and categorizing crashes

- Looking into printTextString



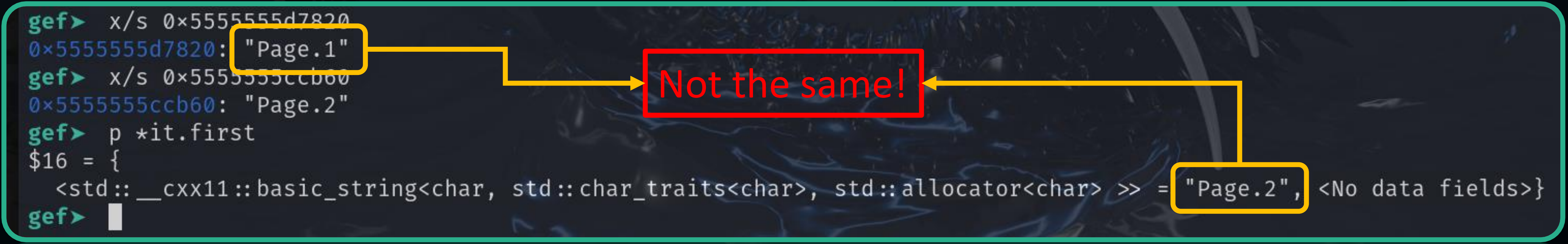
```
static void printTextString(const GooString *s, const UnicodeMap *uMap)
{
    char buf[8];
    std::vector<Unicode> u = TextStringToUCS4(s->toStr());
    for (const auto &c : u) {
        int n = uMap->mapUnicode(c, buf, sizeof(buf));
        fwrite(buf, 1, n, stdout);
    }
}
```

5. Analyzing and categorizing crashes



```
gef➤  x/s 0×5555555d7820
0×5555555d7820: "Page.1"
gef➤  x/s 0×5555555ccb60
0×5555555ccb60: "Page.2"
gef➤  p *it.first
$16 = {
  <std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >> = "Page.2", <No data fields>}
gef➤  
```

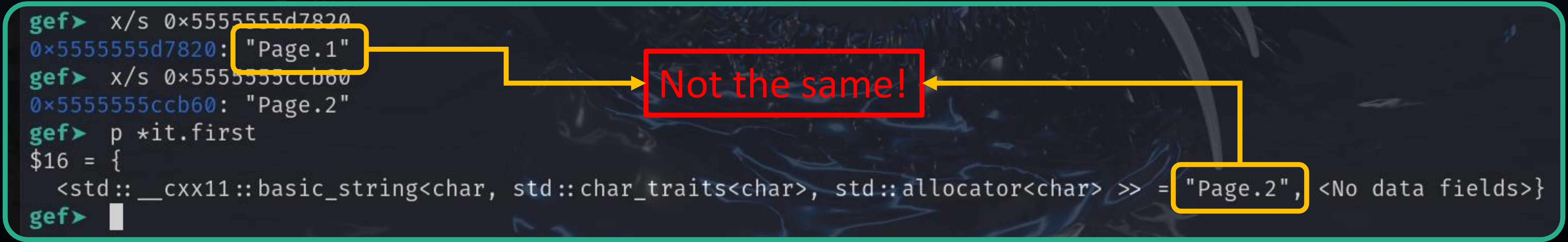
5. Analyzing and categorizing crashes



```
gef➤  x/s 0x5555555d7820
0x5555555d7820: "Page.1"
gef➤  x/s 0x5555555ccb60
0x5555555ccb60: "Page.2"
gef➤  p *it.first
$16 = {
  <std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >> = "Page.2", <No data fields>}
gef➤  
```

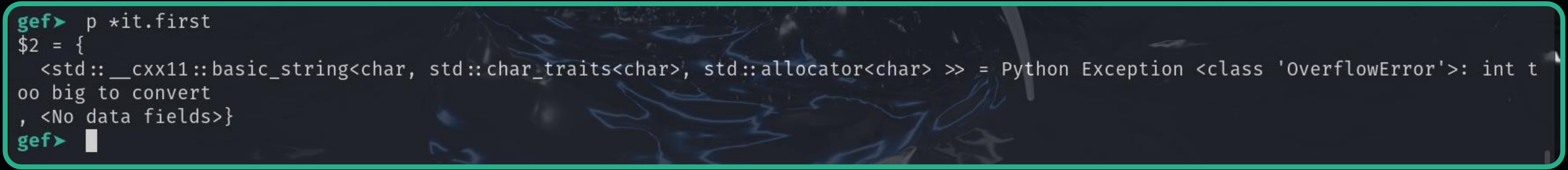
Not the same!

5. Analyzing and categorizing crashes



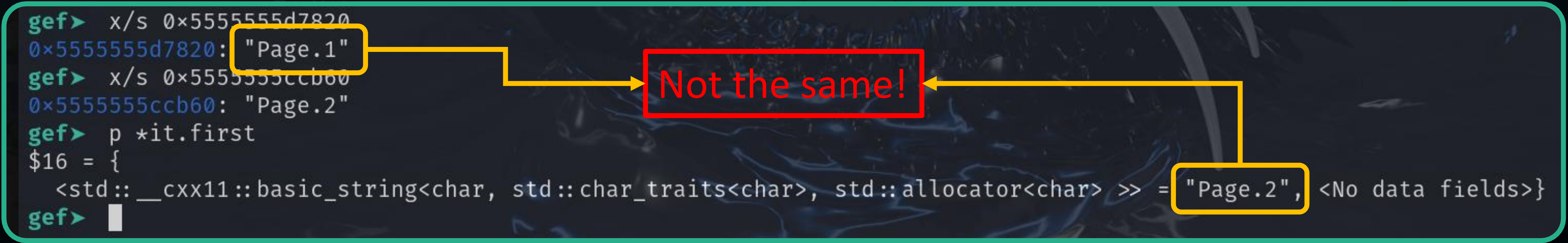
```
gef➤  x/s 0x5555555d7820
0x5555555d7820: "Page.1"
gef➤  x/s 0x5555555ccb60
0x5555555ccb60: "Page.2"
gef➤  p *it.first
$16 = {
  <std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >> = "Page.2", <No data fields>}
gef➤ 
```

Not the same!



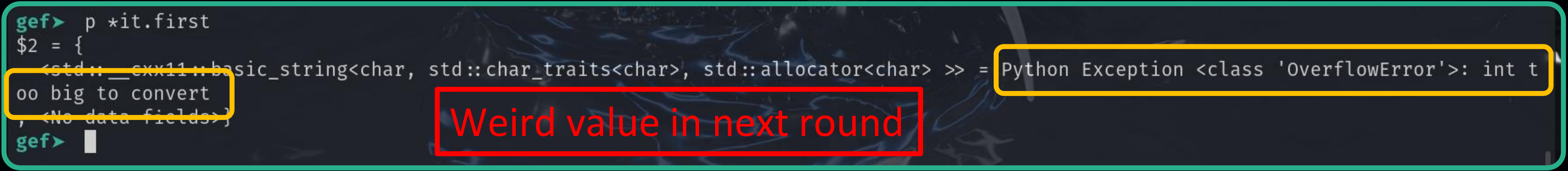
```
gef➤  p *it.first
$2 = {
  <std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >> = Python Exception <class 'OverflowError'>: int t
oo big to convert
, <No data fields>}
gef➤ 
```

5. Analyzing and categorizing crashes



```
gef➤  x/s 0x5555555d7820
0x5555555d7820: "Page.1"
gef➤  x/s 0x5555555ccb60
0x5555555ccb60: "Page.2"
gef➤  p *it.first
$16 = {
  <std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >> = "Page.2", <No data fields>}
gef➤
```

Not the same!

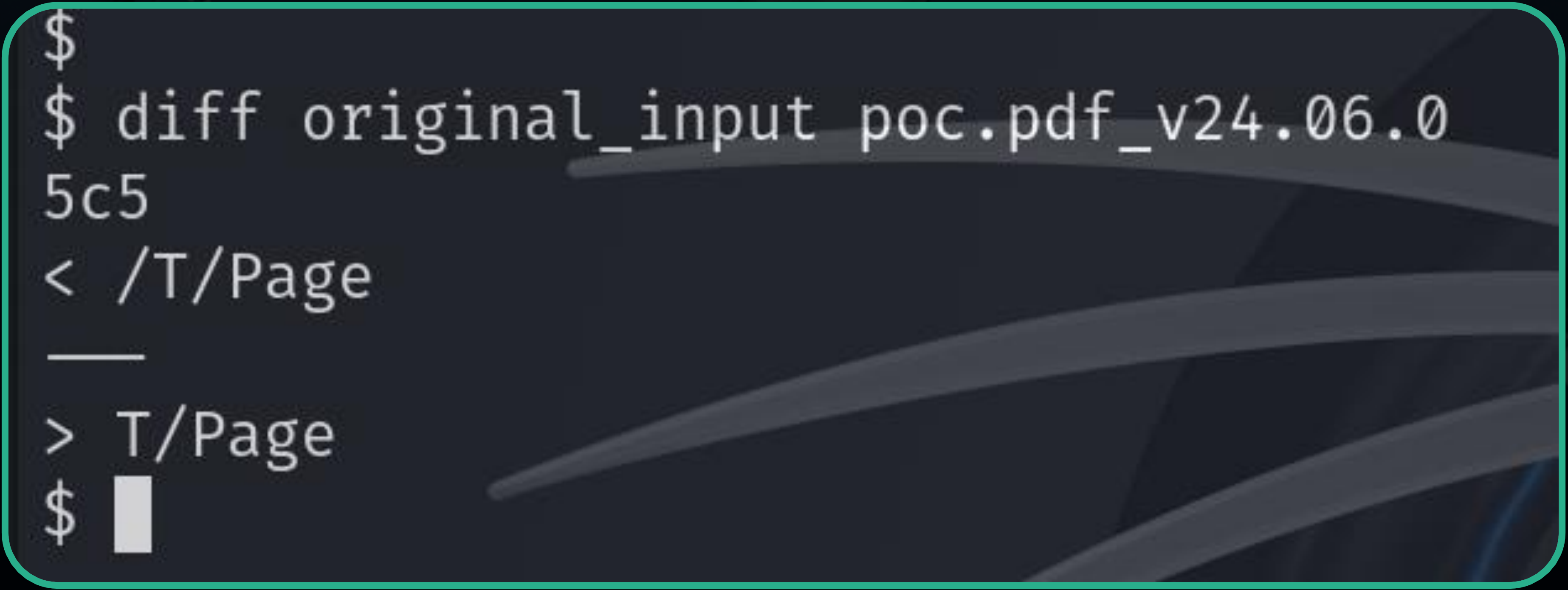


```
gef➤  p *it.first
$2 = {
  <std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >> = Python Exception <class 'OverflowError'>: int t
oo big to convert
, <No data fields>}
gef➤
```

Weird value in next round

5. Analyzing and categorizing crashes

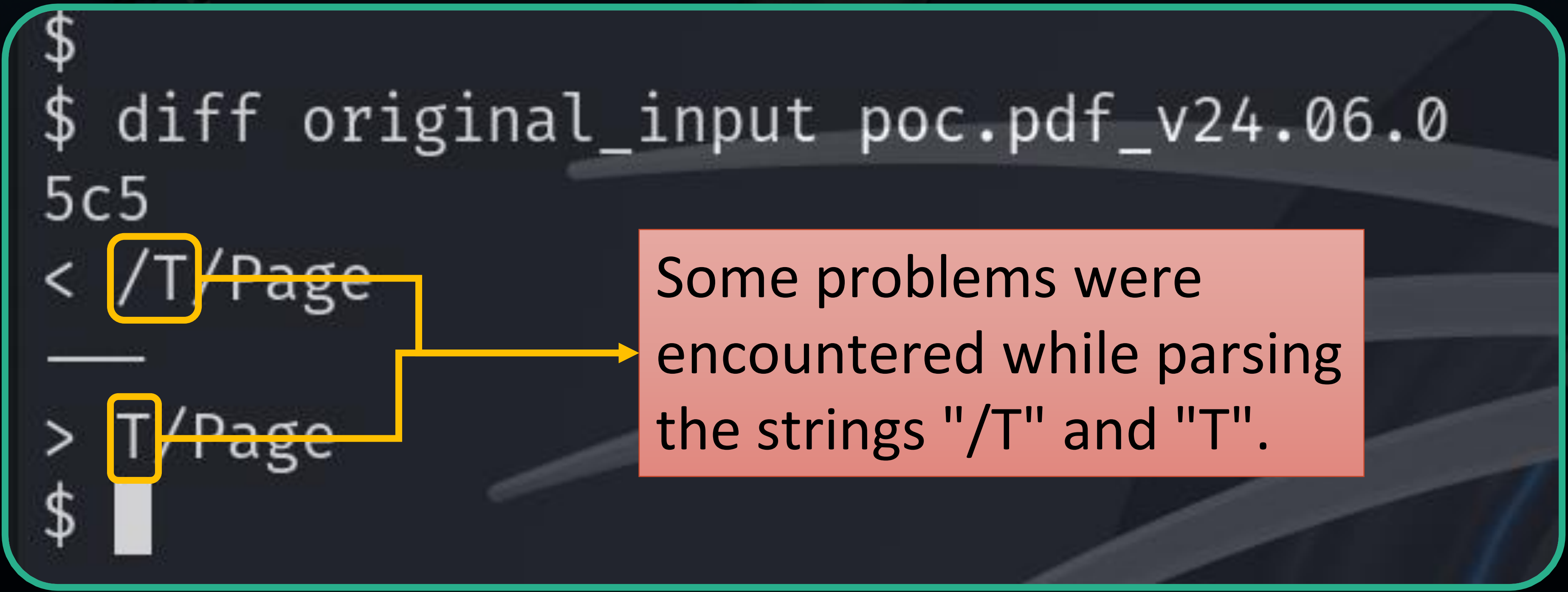
- Comparing mutation



```
$
$ diff original_input poc.pdf_v24.06.0
5c5
< /T/Page
---
> T/Page
$ █
```

5. Analyzing and categorizing crashes

- Comparing mutation



```
$
$ diff original_input poc.pdf_v24.06.0
5c5
< /T/Page
---
> T/Page
$ 
```

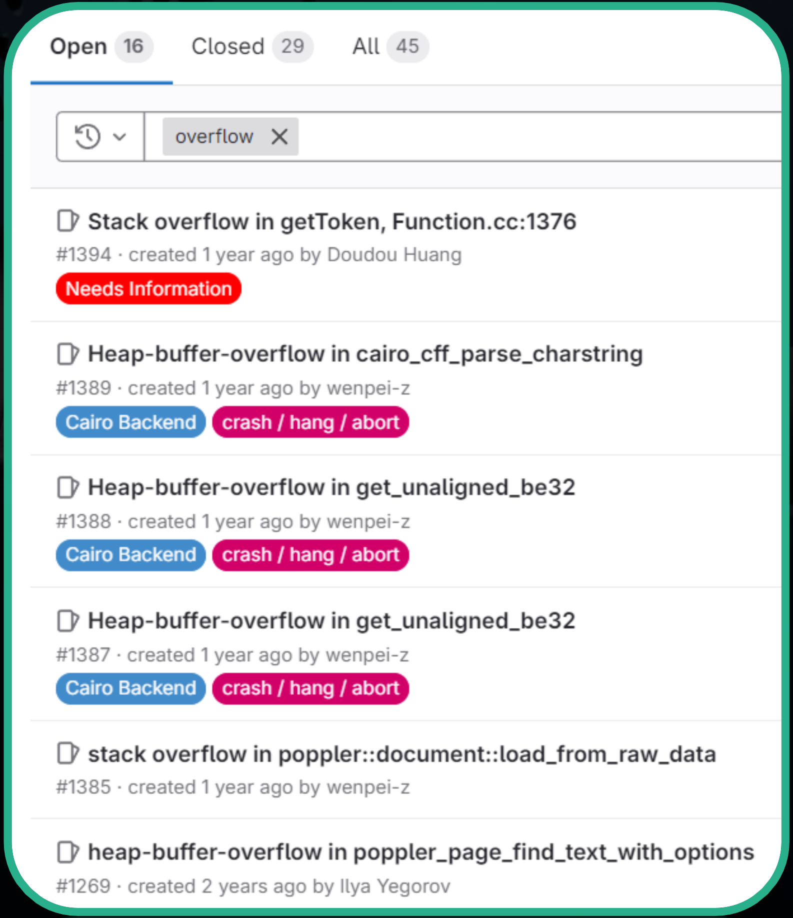
Some problems were encountered while parsing the strings "/T" and "T".

Table Of Contents

1. Selecting target
2. Narrowing the scope
3. Reading the source code
4. Fuzzing
5. Analyzing and categorizing crashes
6. Submitting issues to the author
7. Verifying patches
8. Attempting to submit CVE applications and finally obtaining a CVE

6. Submitting issues to the author

- Searching for samples...



Stack overflow in getToken, Function.cc:1376
#1394 · created 1 year ago by Doudou Huang
Needs Information

Heap-buffer-overflow in cairo_cff_parse_charstring
#1389 · created 1 year ago by wenpei-z
Cairo Backend crash / hang / abort

Heap-buffer-overflow in get_unaligned_be32
#1388 · created 1 year ago by wenpei-z
Cairo Backend crash / hang / abort

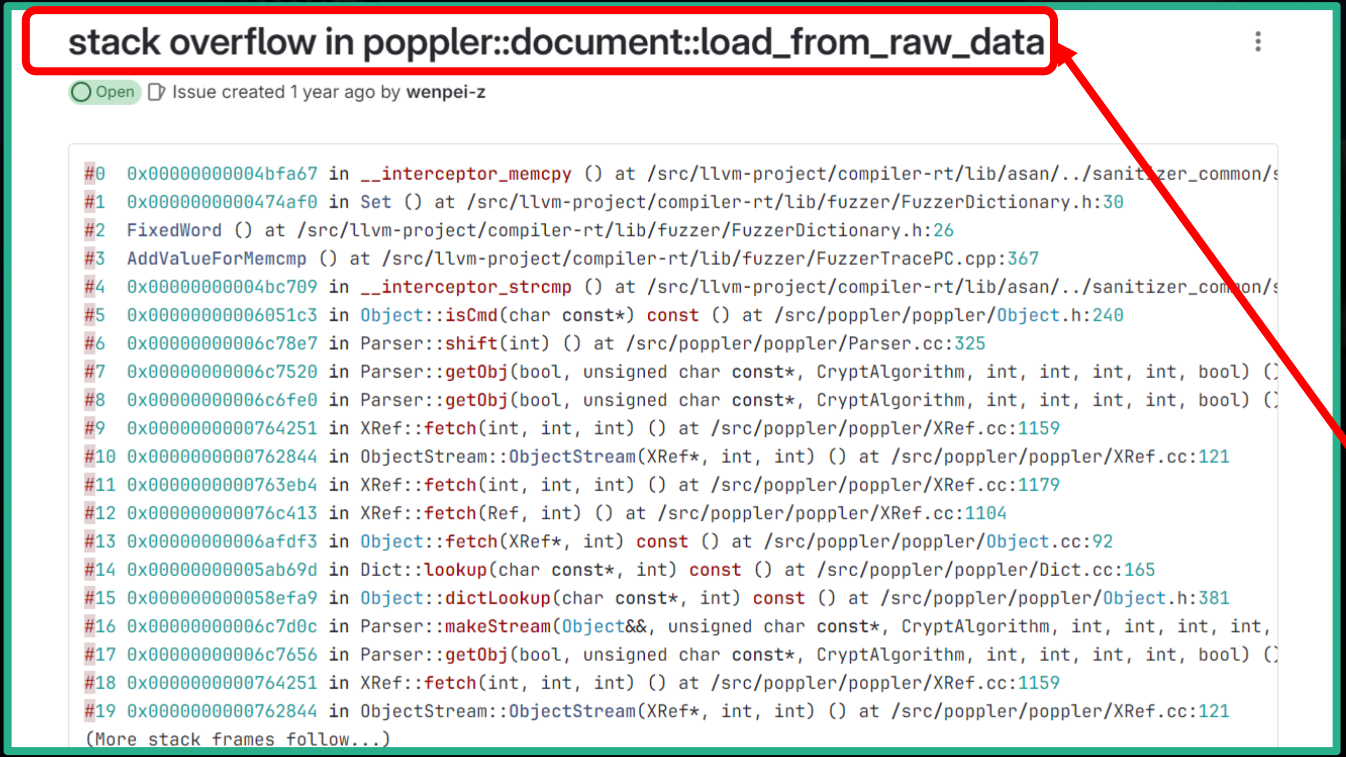
Heap-buffer-overflow in get_unaligned_be32
#1387 · created 1 year ago by wenpei-z
Cairo Backend crash / hang / abort

stack overflow in poppler::document::load_from_raw_data
#1385 · created 1 year ago by wenpei-z

heap-buffer-overflow in poppler_page_find_text_with_options
#1269 · created 2 years ago by Ilya Yegorov

6. Submitting issues to the author

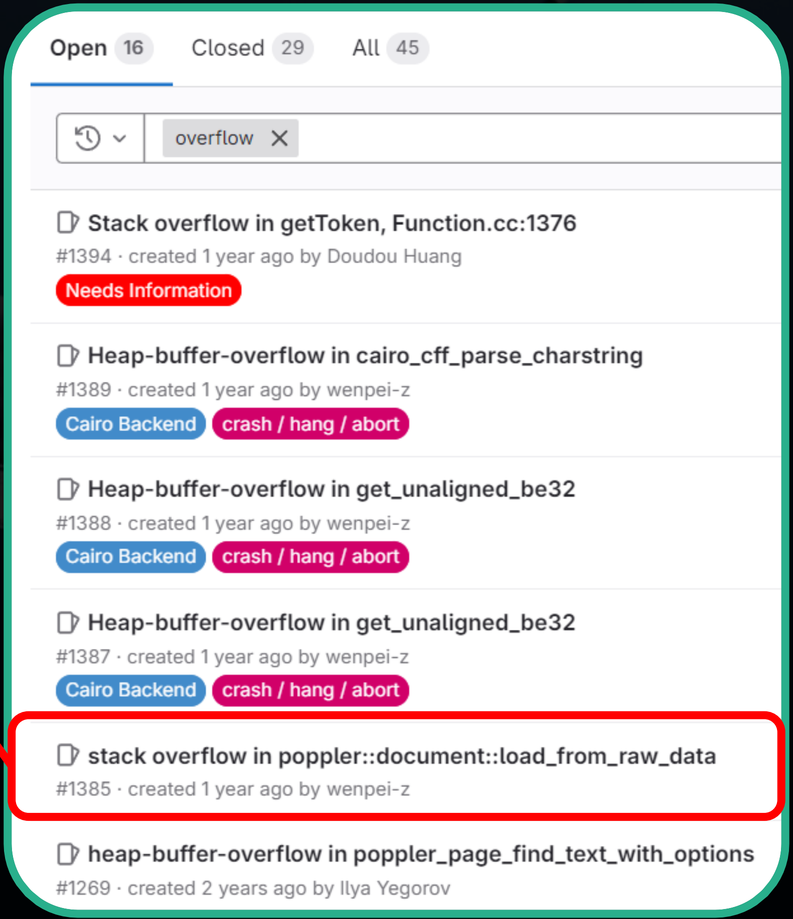
- Searching for samples...



stack overflow in poppler::document::load_from_raw_data

Open Issue created 1 year ago by **wenpei-z**

```
#0  0x00000000004bfa67 in __interceptor_memcpy () at /src/llvm-project/compiler-rt/lib/asan/../sanitizer_common/s
#1  0x0000000000474af0 in Set () at /src/llvm-project/compiler-rt/lib/fuzzer/FuzzerDictionary.h:30
#2  FixedWord () at /src/llvm-project/compiler-rt/lib/fuzzer/FuzzerDictionary.h:26
#3  AddValueForMemcmp () at /src/llvm-project/compiler-rt/lib/fuzzer/FuzzerTracePC.cpp:367
#4  0x00000000004bc709 in __interceptor_strcmp () at /src/llvm-project/compiler-rt/lib/asan/../sanitizer_common/s
#5  0x00000000006051c3 in Object::isCmd(char const*) const () at /src/poppler/poppler/Object.h:240
#6  0x00000000006c78e7 in Parser::shift(int) () at /src/poppler/poppler/Parser.cc:325
#7  0x00000000006c7520 in Parser::getObj(bool, unsigned char const*, CryptAlgorithm, int, int, int, int, bool) ()
#8  0x00000000006c6fe0 in Parser::getObj(bool, unsigned char const*, CryptAlgorithm, int, int, int, int, bool) ()
#9  0x0000000000764251 in XRef::fetch(int, int, int) () at /src/poppler/poppler/XRef.cc:1159
#10 0x0000000000762844 in ObjectStream::ObjectStream(XRef*, int, int) () at /src/poppler/poppler/XRef.cc:121
#11 0x0000000000763eb4 in XRef::fetch(int, int, int) () at /src/poppler/poppler/XRef.cc:1179
#12 0x000000000076c413 in XRef::fetch(Ref, int) () at /src/poppler/poppler/XRef.cc:1104
#13 0x00000000006afdf3 in Object::fetch(XRef*, int) const () at /src/poppler/poppler/Object.cc:92
#14 0x00000000005ab69d in Dict::lookup(char const*, int) const () at /src/poppler/poppler/Dict.cc:165
#15 0x000000000058efa9 in Object::dictLookup(char const*, int) const () at /src/poppler/poppler/Object.h:381
#16 0x00000000006c7d0c in Parser::makeStream(Object&&, unsigned char const*, CryptAlgorithm, int, int, int, int,
#17 0x00000000006c7656 in Parser::getObj(bool, unsigned char const*, CryptAlgorithm, int, int, int, int, bool) ()
#18 0x0000000000764251 in XRef::fetch(int, int, int) () at /src/poppler/poppler/XRef.cc:1159
#19 0x0000000000762844 in ObjectStream::ObjectStream(XRef*, int, int) () at /src/poppler/poppler/XRef.cc:121
(More stack frames follow...)
```



Stack overflow in getToken, Function.cc:1376
#1394 · created 1 year ago by Doudou Huang
Needs Information

Heap-buffer-overflow in cairo_cff_parse_charstring
#1389 · created 1 year ago by wenpei-z
Cairo Backend crash / hang / abort

Heap-buffer-overflow in get_unaligned_be32
#1388 · created 1 year ago by wenpei-z
Cairo Backend crash / hang / abort

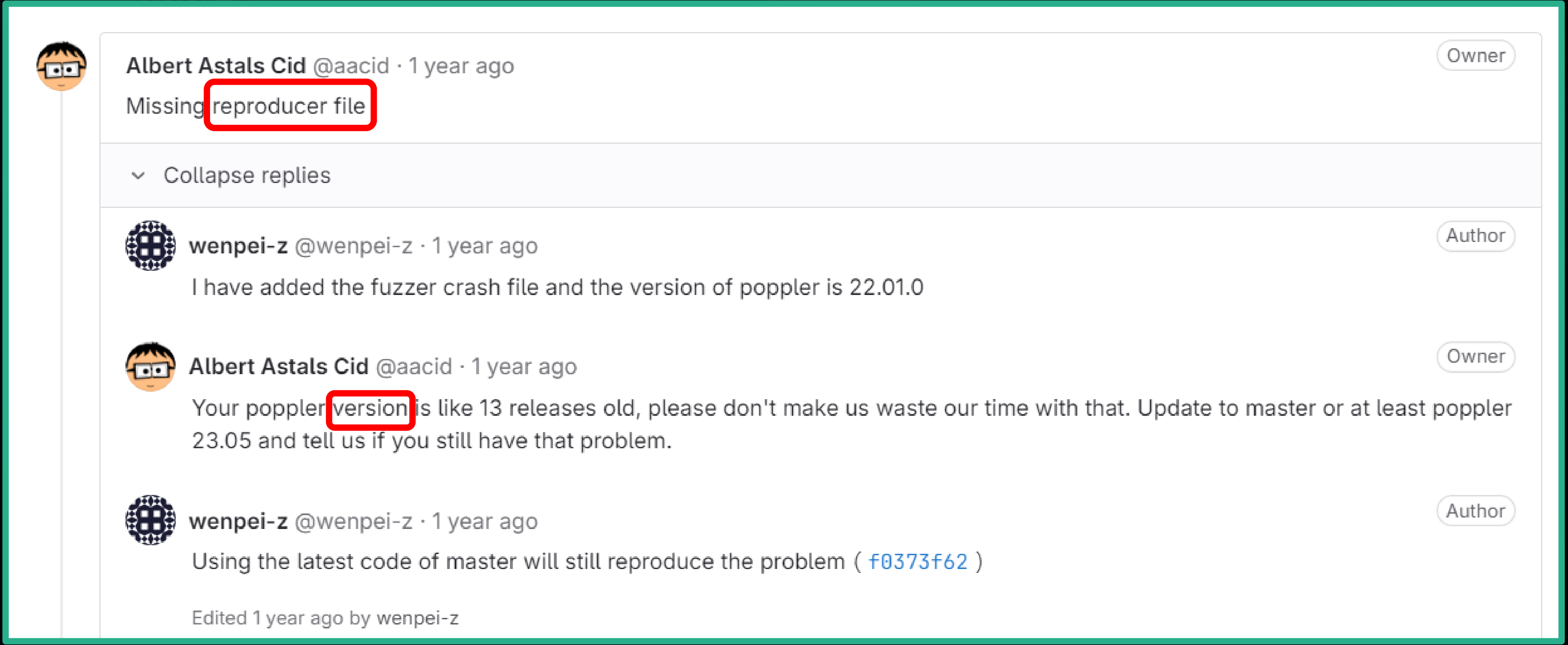
Heap-buffer-overflow in get_unaligned_be32
#1387 · created 1 year ago by wenpei-z
Cairo Backend crash / hang / abort

stack overflow in poppler::document::load_from_raw_data
#1385 · created 1 year ago by wenpei-z

heap-buffer-overflow in poppler_page_find_text_with_options
#1269 · created 2 years ago by Ilya Yegorov

6. Submitting issues to the author

- Searching for samples...



Albert Astals Cid @aacid · 1 year ago

Owner

Missing reproducer file

wenpei-z @wenpei-z · 1 year ago

Author

I have added the fuzzer crash file and the version of poppler is 22.01.0

Albert Astals Cid @aacid · 1 year ago

Owner

Your poppler version is like 13 releases old, please don't make us waste our time with that. Update to master or at least poppler 23.05 and tell us if you still have that problem.

wenpei-z @wenpei-z · 1 year ago

Author

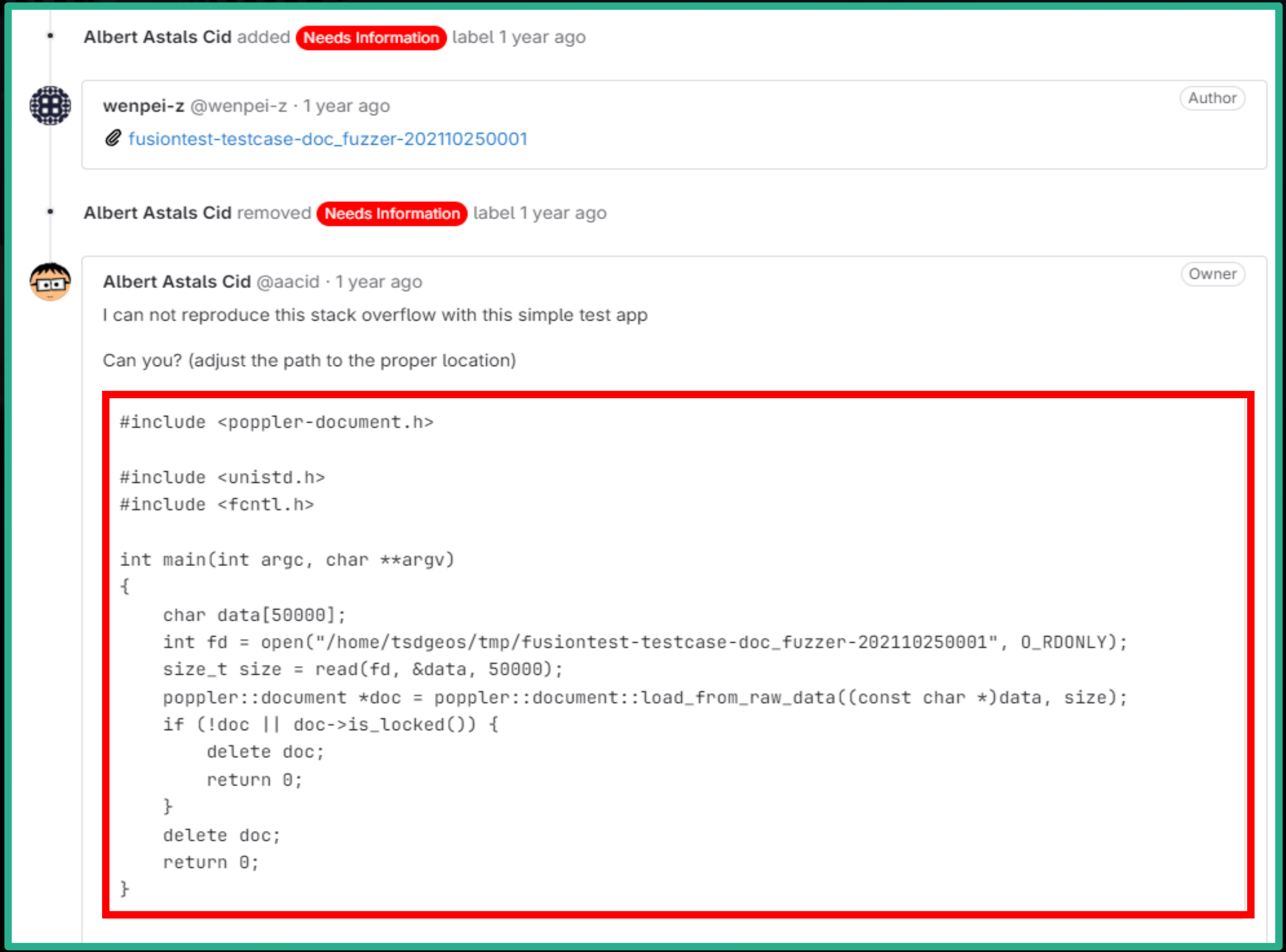
Using the latest code of master will still reproduce the problem (`f0373f62`)

Edited 1 year ago by wenpei-z

By 潘甫翰 Sharkkcode

6. Submitting issues to the author

- Searching for samples…



Albert Astals Cid added Needs Information label 1 year ago

wenpei-z @wenpei-z · 1 year ago

fusiontest-testcase-doc_fuzzer-202110250001

Albert Astals Cid removed Needs Information label 1 year ago

Albert Astals Cid @aacid · 1 year ago

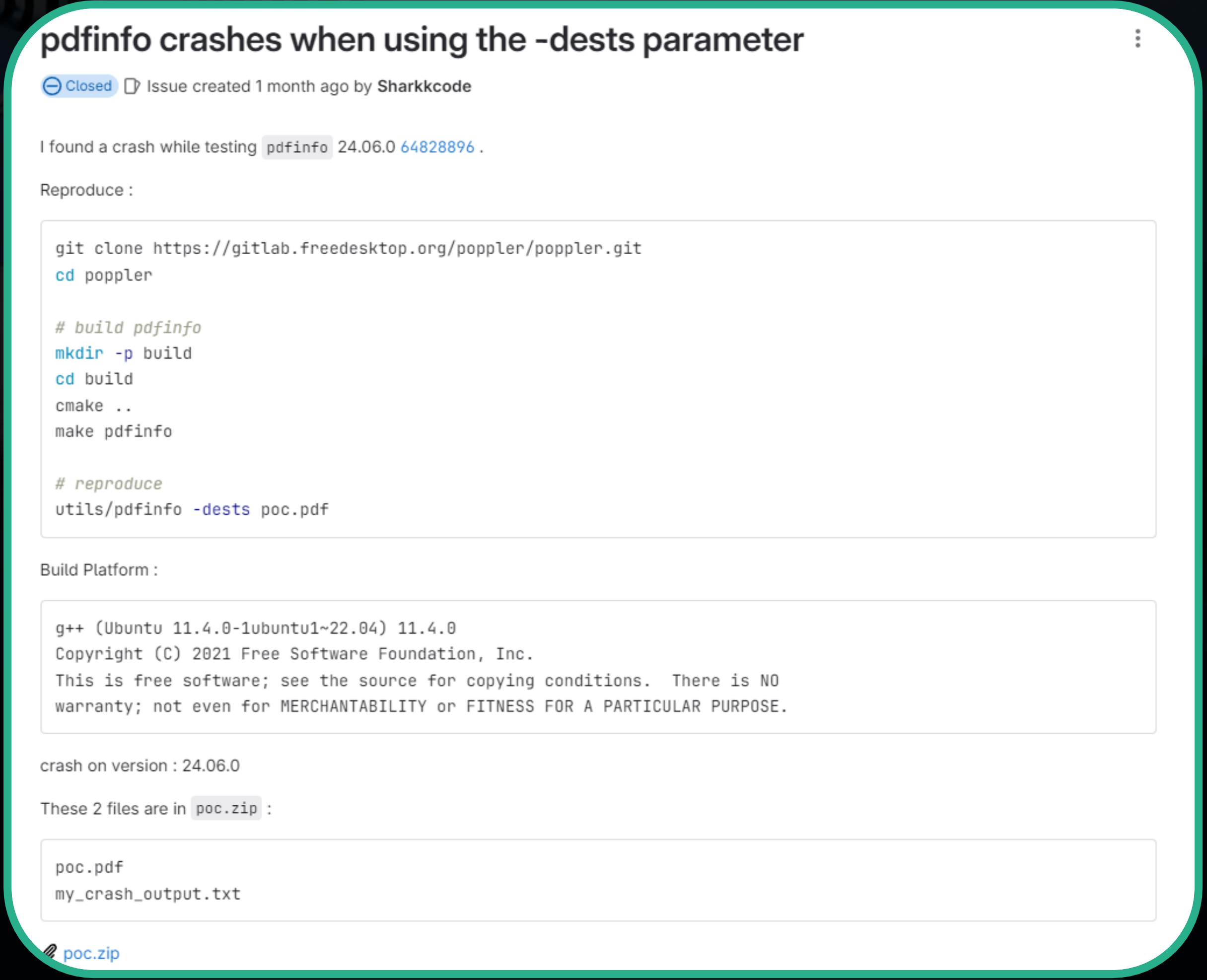
I can not reproduce this stack overflow with this simple test app

Can you? (adjust the path to the proper location)

```
#include <poppler-document.h>

#include <unistd.h>
#include <fcntl.h>

int main(int argc, char **argv)
{
    char data[50000];
    int fd = open("/home/tsdgeos/tmp/fusiontest-testcase-doc_fuzzer-202110250001", O_RDONLY);
    size_t size = read(fd, &data, 50000);
    poppler::document *doc = poppler::document::load_from_raw_data((const char *)data, size);
    if (!doc || doc->is_locked()) {
        delete doc;
        return 0;
    }
    delete doc;
    return 0;
}
```

6. Submitting issues to the author

- My issue report



pdfinfo crashes when using the -dests parameter

Closed Issue created 1 month ago by **Sharkkcode**

I found a crash while testing `pdfinfo` 24.06.0 64828896 .

Reproduce :

```
git clone https://gitlab.freedesktop.org/poppler/poppler.git
cd poppler

# build pdfinfo
mkdir -p build
cd build
cmake ..
make pdfinfo

# reproduce
utils/pdfinfo -dests poc.pdf
```

Build Platform :

```
g++ (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

crash on version : 24.06.0

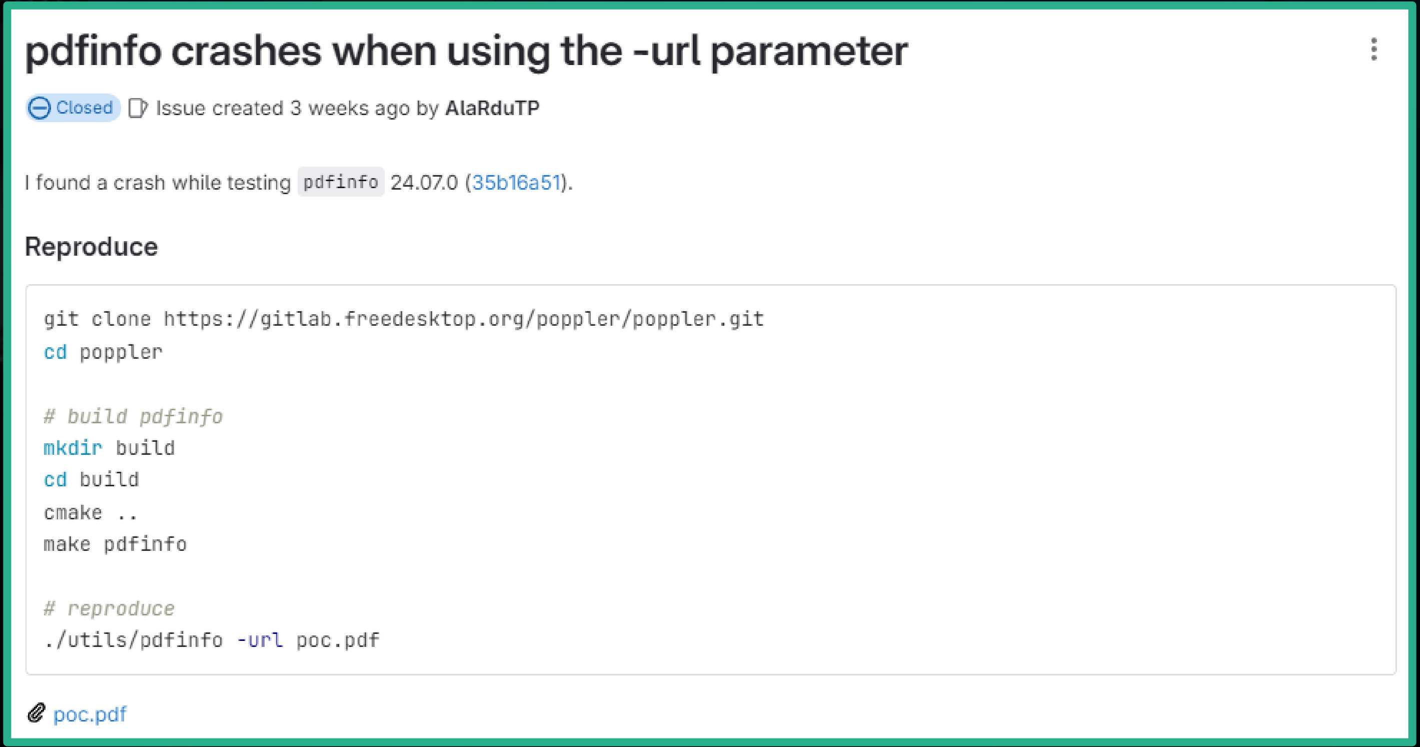
These 2 files are in `poc.zip` :

```
poc.pdf
my_crash_output.txt
```

poc.zip

6. Submitting issues to the author

- Somebody copied my template (maybe?) for other bugs



pdfinfo crashes when using the -url parameter

Closed Issue created 3 weeks ago by **AlaRduTP**

I found a crash while testing `pdfinfo` 24.07.0 (35b16a51).

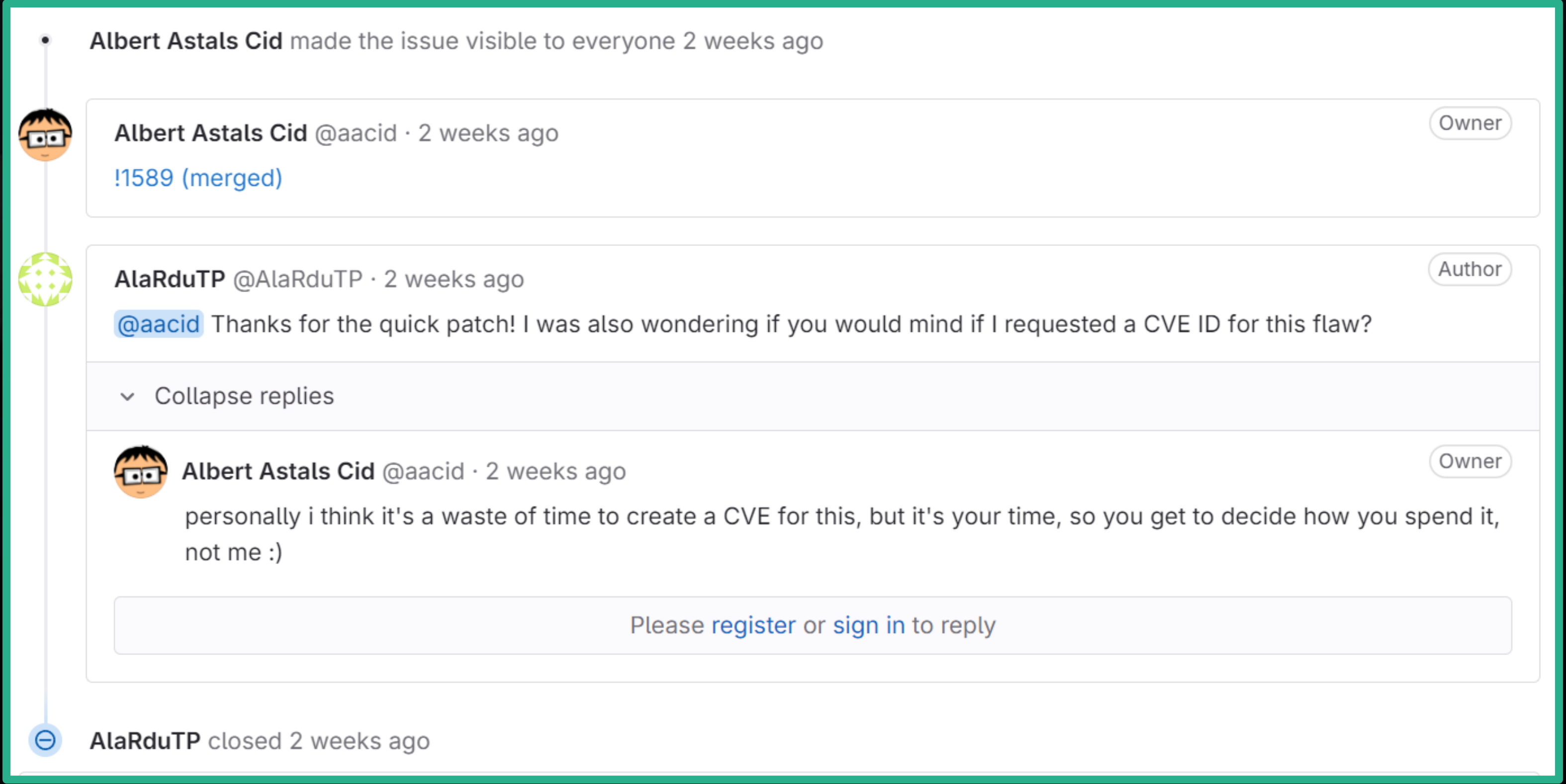
Reproduce

```
git clone https://gitlab.freedesktop.org/poppler/poppler.git
cd poppler

# build pdfinfo
mkdir build
cd build
cmake ..
make pdfinfo

# reproduce
./utils/pdfinfo -url poc.pdf
```

poc.pdf

6. Submitting issues to the author

- Somebody copied my template (maybe?) for other bugs
- The author responded in other issue reports on this account.



Albert Astals Cid made the issue visible to everyone 2 weeks ago

Albert Astals Cid @aacid · 2 weeks ago

!1589 (merged)

AlaRduTP @AlaRduTP · 2 weeks ago

@aacid Thanks for the quick patch! I was also wondering if you would mind if I requested a CVE ID for this flaw?

Albert Astals Cid @aacid · 2 weeks ago

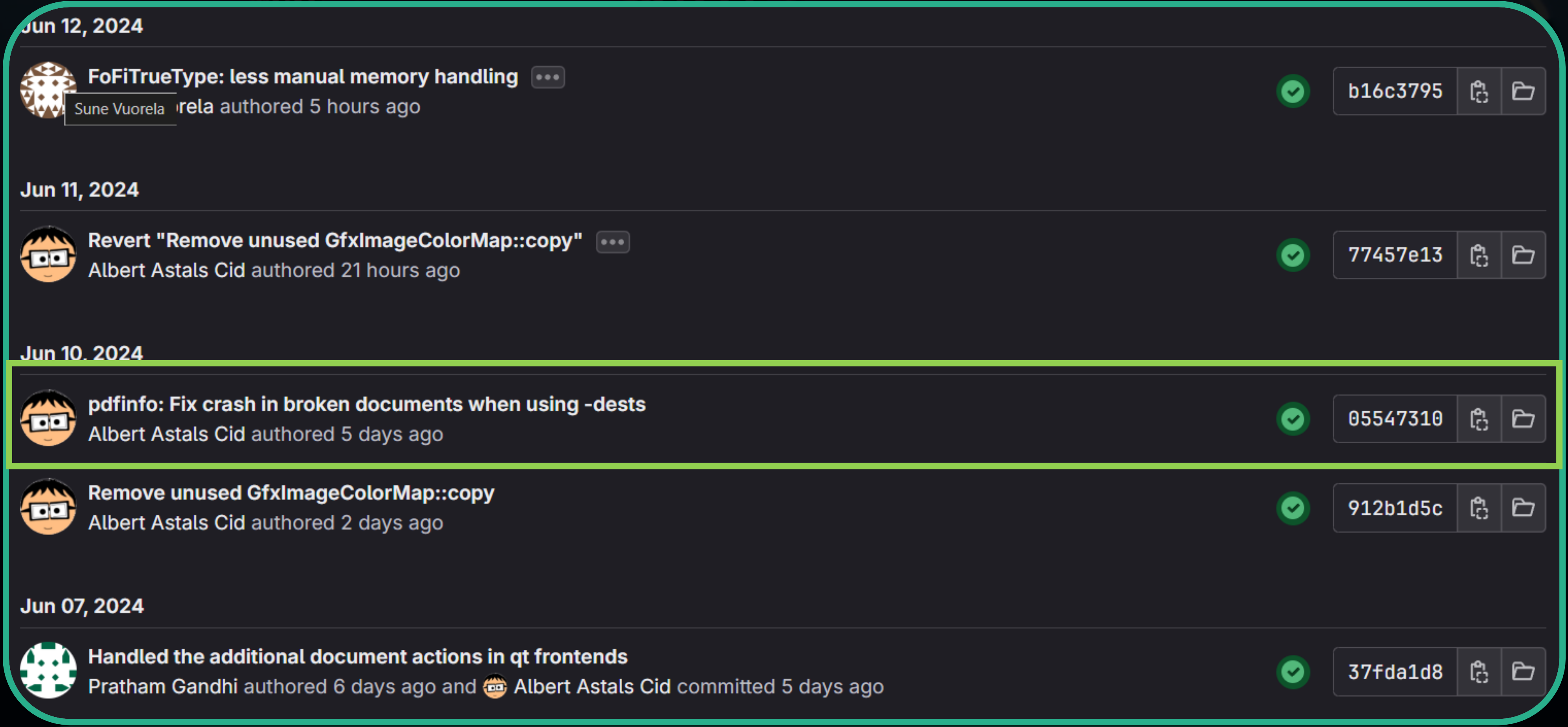
personally i think it's a waste of time to create a CVE for this, but it's your time, so you get to decide how you spend it, not me :)

AlaRduTP closed 2 weeks ago

Table Of Contents

1. Selecting target
2. Narrowing the scope
3. Reading the source code
4. Fuzzing
5. Analyzing and categorizing crashes
6. Submitting issues to the author
7. Verifying patches
8. Attempting to submit CVE applications and finally obtaining a CVE

7. Verifying patches



Jun 12, 2024

FoFiTrueType: less manual memory handling ⋯
Sune Vuorela rela authored 5 hours ago
✓ b16c3795

Jun 11, 2024

Revert "Remove unused GfxImageColorMap::copy" ⋯
Albert Astals Cid authored 21 hours ago
✓ 77457e13

Jun 10, 2024

pdfinfo: Fix crash in broken documents when using -dests
Albert Astals Cid authored 5 days ago
✓ 05547310

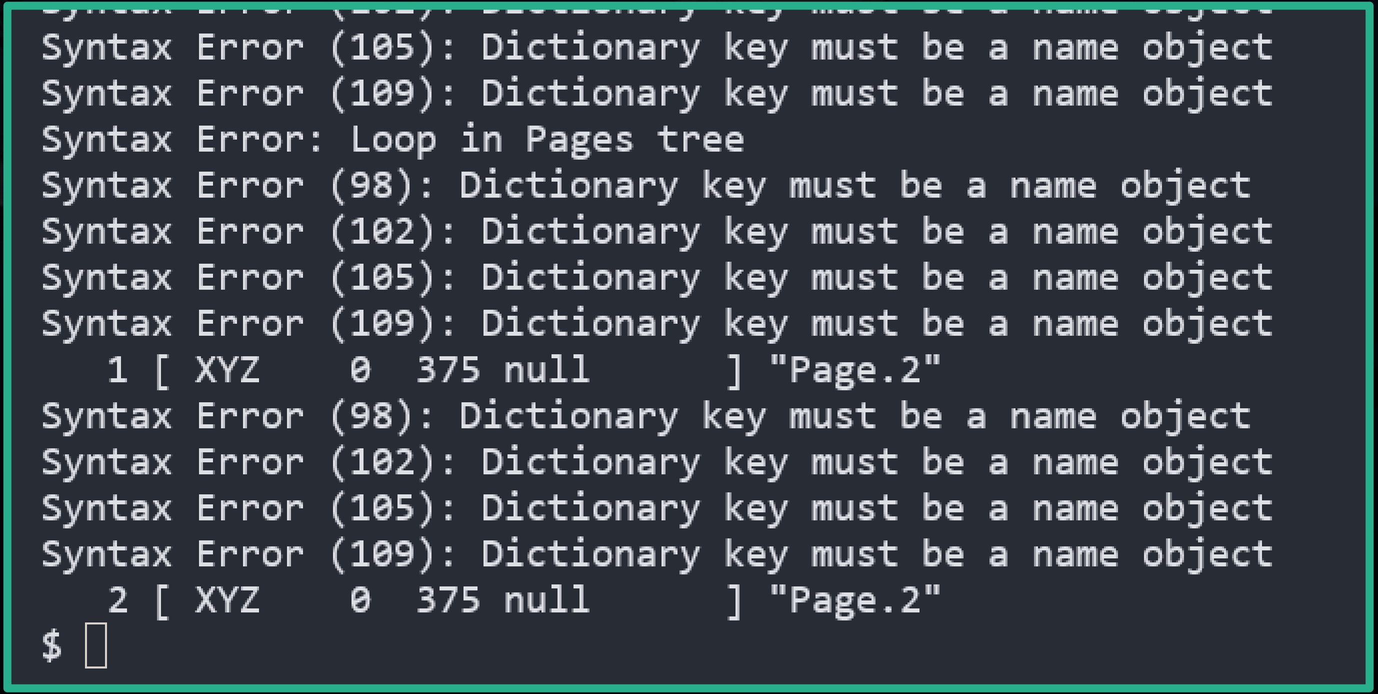
Remove unused GfxImageColorMap::copy
Albert Astals Cid authored 2 days ago
✓ 912b1d5c

Jun 07, 2024

Handled the additional document actions in qt frontends
Pratham Gandhi authored 6 days ago and Albert Astals Cid committed 5 days ago
✓ 37fda1d8

7. Verifying patches

NO CRASH, BUG FIXED.

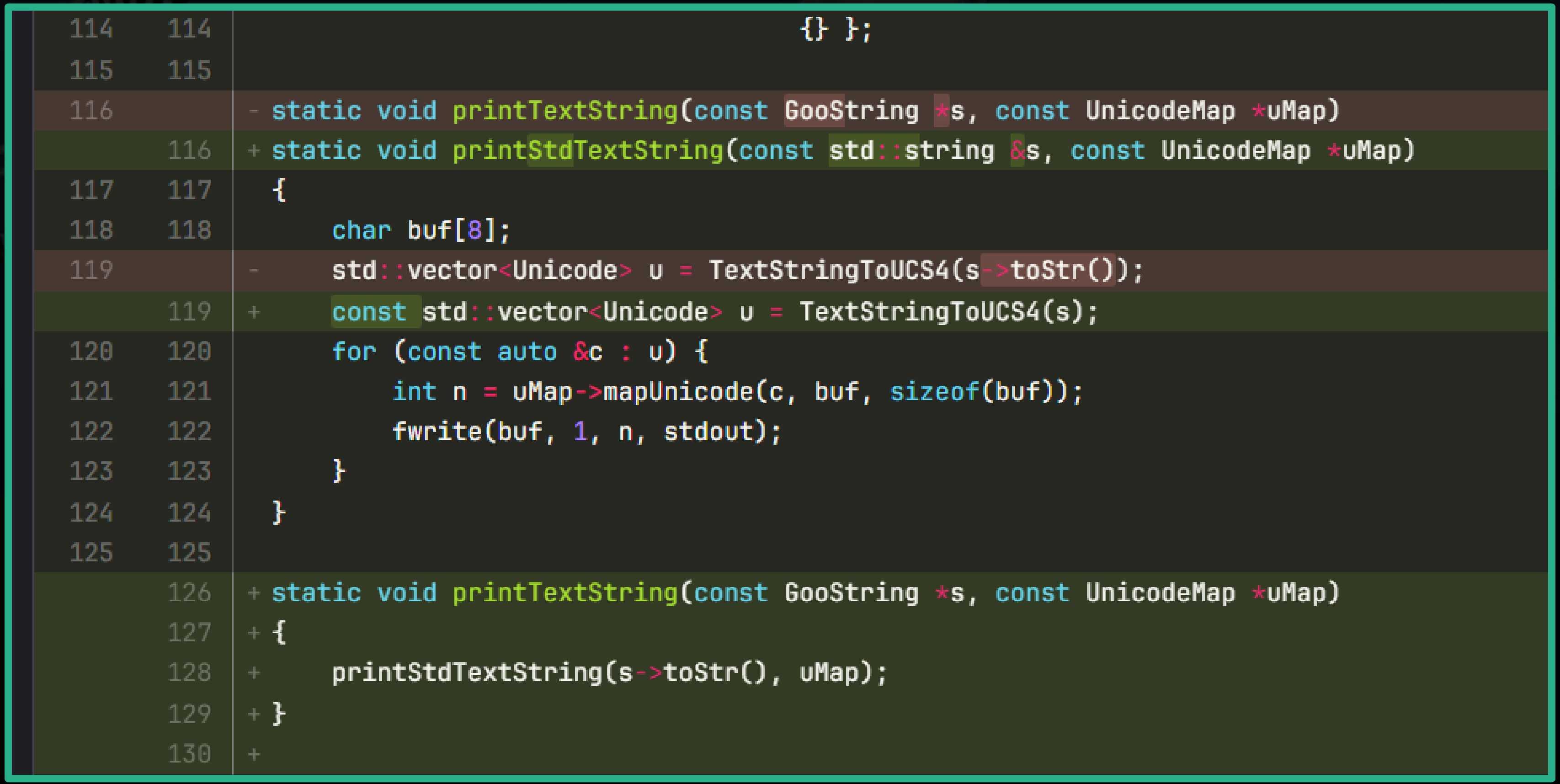


```
Syntax Error (102): Dictionary key must be a name object
Syntax Error (105): Dictionary key must be a name object
Syntax Error (109): Dictionary key must be a name object
Syntax Error: Loop in Pages tree
Syntax Error (98): Dictionary key must be a name object
Syntax Error (102): Dictionary key must be a name object
Syntax Error (105): Dictionary key must be a name object
Syntax Error (109): Dictionary key must be a name object
   1 [ XYZ    0  375 null      ] "Page.2"
Syntax Error (98): Dictionary key must be a name object
Syntax Error (102): Dictionary key must be a name object
Syntax Error (105): Dictionary key must be a name object
Syntax Error (109): Dictionary key must be a name object
   2 [ XYZ    0  375 null      ] "Page.2"
$ 
```

By 潘甫翰 Sharkkcode

7. Verifying patches

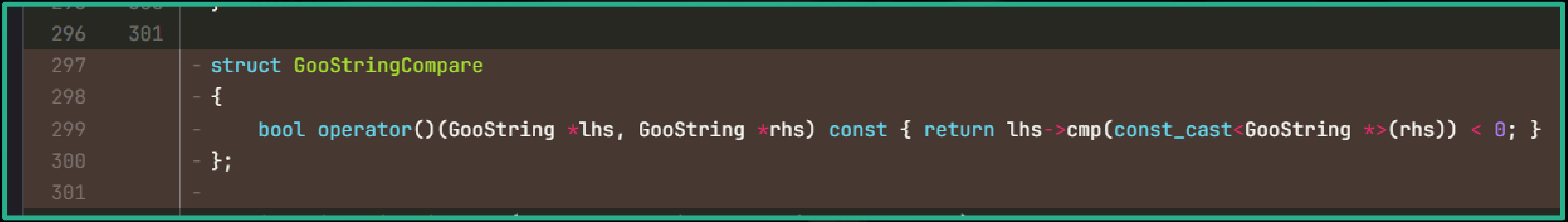
std::string



```
                                   {} };

- static void printTextString(const GooString *s, const UnicodeMap *uMap)
+ static void printStdTextString(const std::string &s, const UnicodeMap *uMap)
  {
      char buf[8];
-     std::vector<Unicode> u = TextStringToUCS4(s->toStr());
+     const std::vector<Unicode> u = TextStringToUCS4(s);
      for (const auto &c : u) {
          int n = uMap->mapUnicode(c, buf, sizeof(buf));
          fwrite(buf, 1, n, stdout);
      }
  }

+ static void printTextString(const GooString *s, const UnicodeMap *uMap)
+ {
+     printStdTextString(s->toStr(), uMap);
+ }
+
```

7. Verifying patches

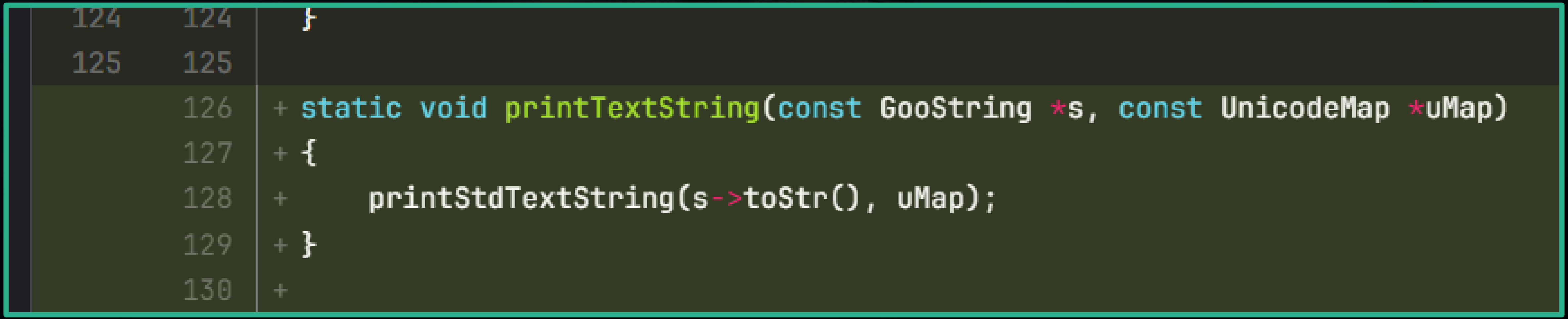
Remove unused function



```
  }

- struct GooStringCompare
- {
-     bool operator()(GooString *lhs, GooString *rhs) const { return lhs->cmp(const_cast<GooString *>(rhs)) < 0; }
- };
-
```

7. Verifying patches

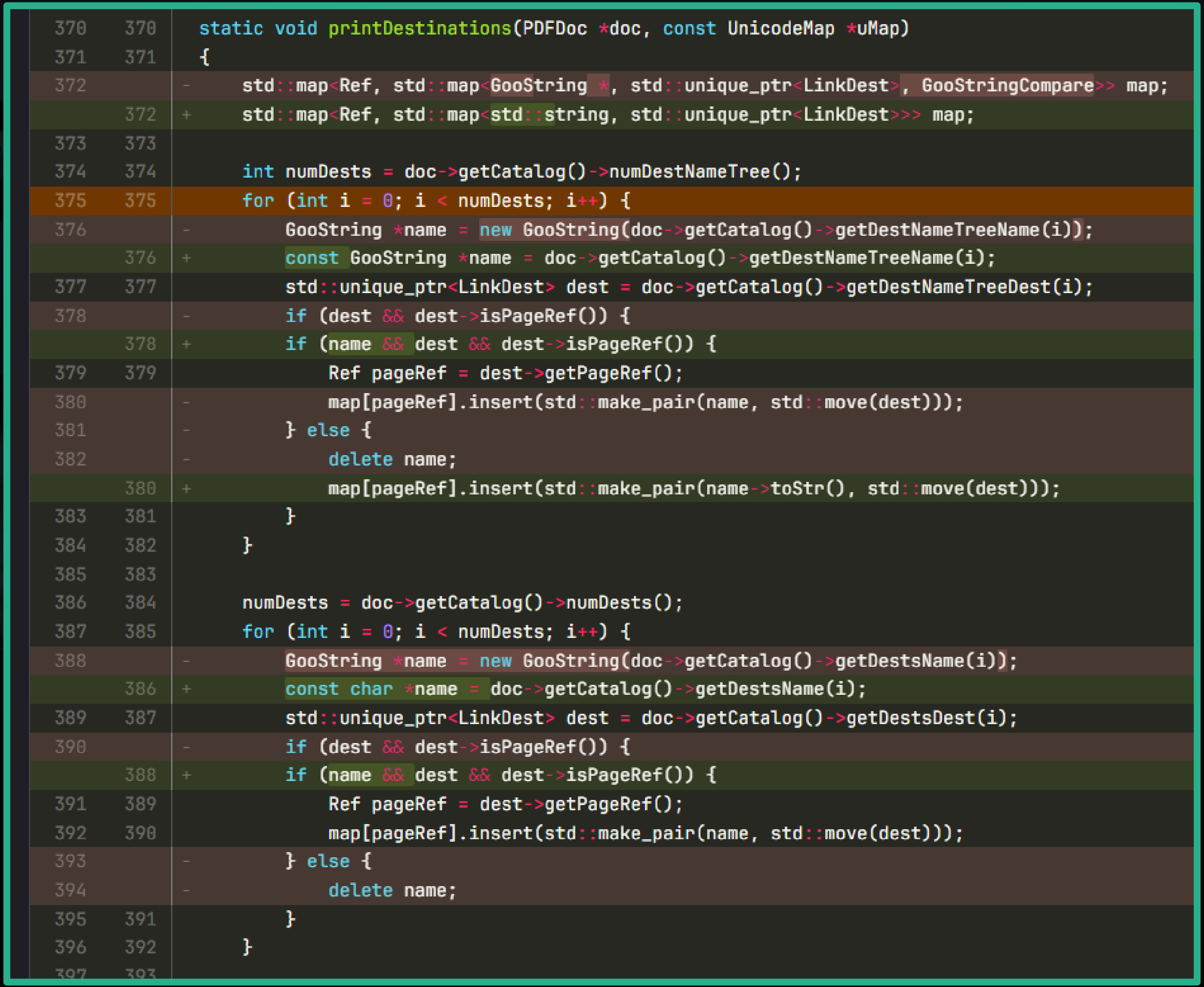
printTextString



```
  }

+ static void printTextString(const GooString *s, const UnicodeMap *uMap)
+ {
+     printStdTextString(s->toStr(), uMap);
+ }
+
```

7. Verifying patches

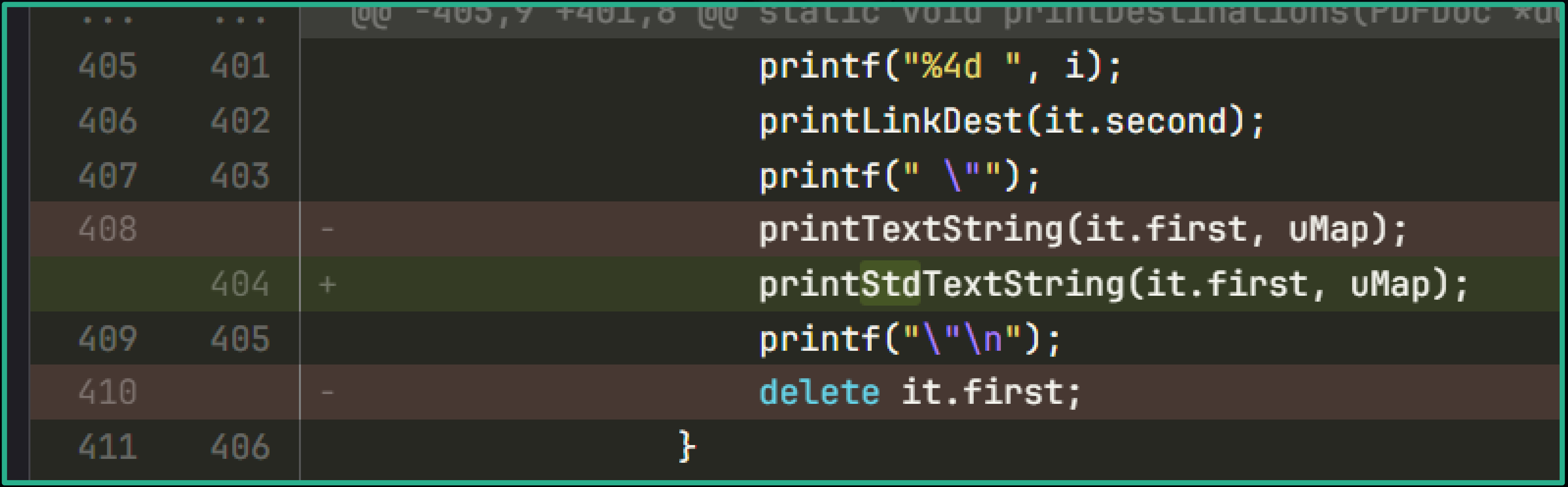
printDestinations



```
 static void printDestinations(PDFDoc *doc, const UnicodeMap *uMap)
 {
-    std::map<Ref, std::map<GooString *, std::unique_ptr<LinkDest>, GooStringCompare>> map;
+    std::map<Ref, std::map<std::string, std::unique_ptr<LinkDest>>> map;

     int numDests = doc->getCatalog()->numDestNameTree();
     for (int i = 0; i < numDests; i++) {
-        GooString *name = new GooString(doc->getCatalog()->getDestNameTreeName(i));
+        const GooString *name = doc->getCatalog()->getDestNameTreeName(i);
         std::unique_ptr<LinkDest> dest = doc->getCatalog()->getDestNameTreeDest(i);
-        if (dest && dest->isPageRef()) {
+        if (name && dest && dest->isPageRef()) {
             Ref pageRef = dest->getPageRef();
-            map[pageRef].insert(std::make_pair(name, std::move(dest)));
-        } else {
-            delete name;
+            map[pageRef].insert(std::make_pair(name->toStr(), std::move(dest)));
         }
     }

     numDests = doc->getCatalog()->numDests();
     for (int i = 0; i < numDests; i++) {
-        GooString *name = new GooString(doc->getCatalog()->getDestsName(i));
+        const char *name = doc->getCatalog()->getDestsName(i);
         std::unique_ptr<LinkDest> dest = doc->getCatalog()->getDestsDest(i);
-        if (dest && dest->isPageRef()) {
+        if (name && dest && dest->isPageRef()) {
             Ref pageRef = dest->getPageRef();
             map[pageRef].insert(std::make_pair(name, std::move(dest)));
-        } else {
-            delete name;
         }
     }
```

7. Verifying patches

Using new function

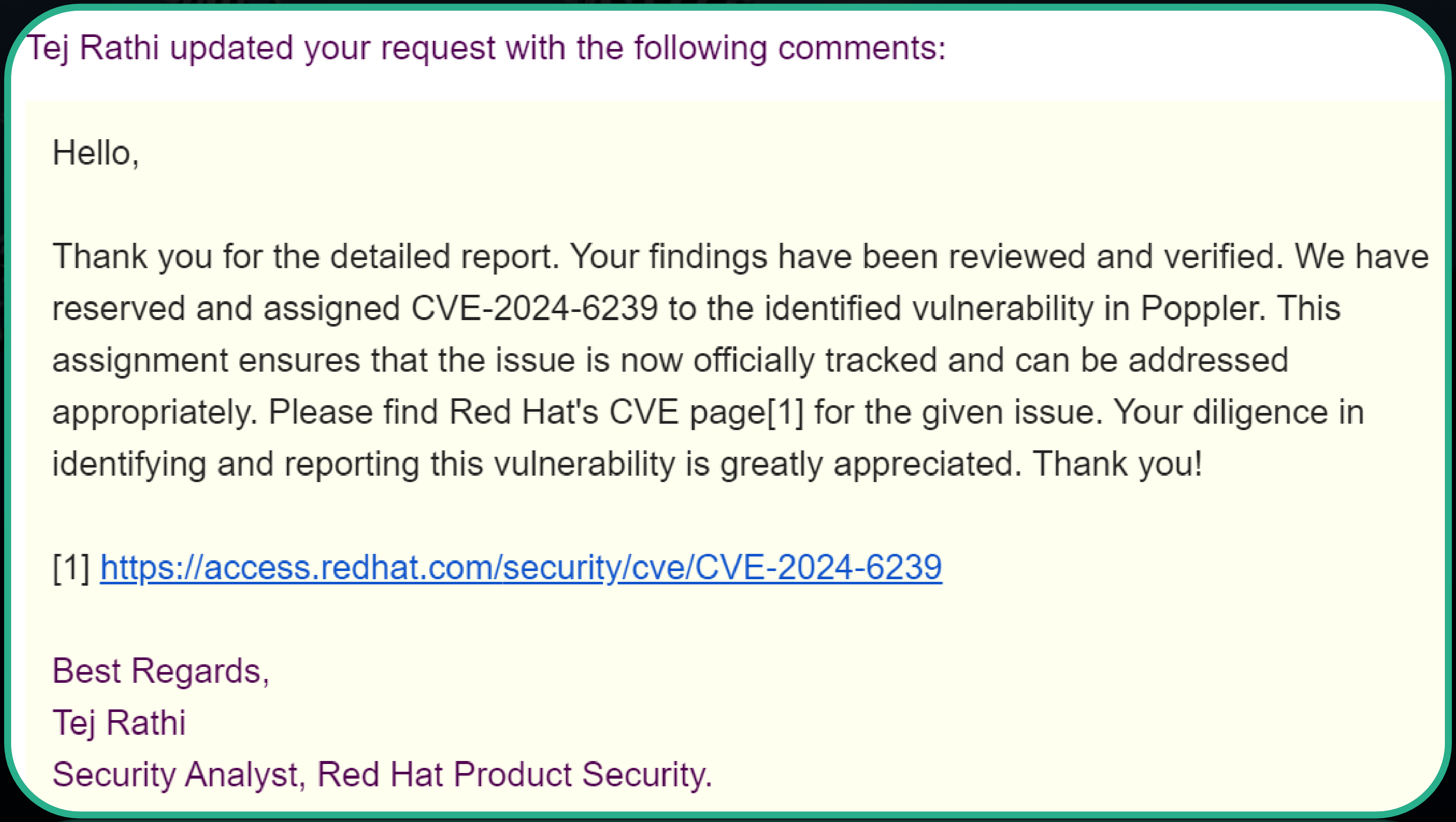


```
@@ -405,9 +401,8 @@ static void printDestinations(PDFDoc *d
                printf("%4d ", i);
                printLinkDest(it.second);
                printf(" \"");
-               printTextString(it.first, uMap);
+               printStdTextString(it.first, uMap);
                printf("\"\n");
-               delete it.first;
            }
```

Table Of Contents

1. Selecting target
2. Narrowing the scope
3. Reading the source code
4. Fuzzing
5. Analyzing and categorizing crashes
6. Submitting issues to the author
7. Verifying patches
8. Attempting to submit CVE applications and finally obtaining a CVE

8. Attempting to submit CVE applications and finally obtaining a CVE



Tej Rathi updated your request with the following comments:

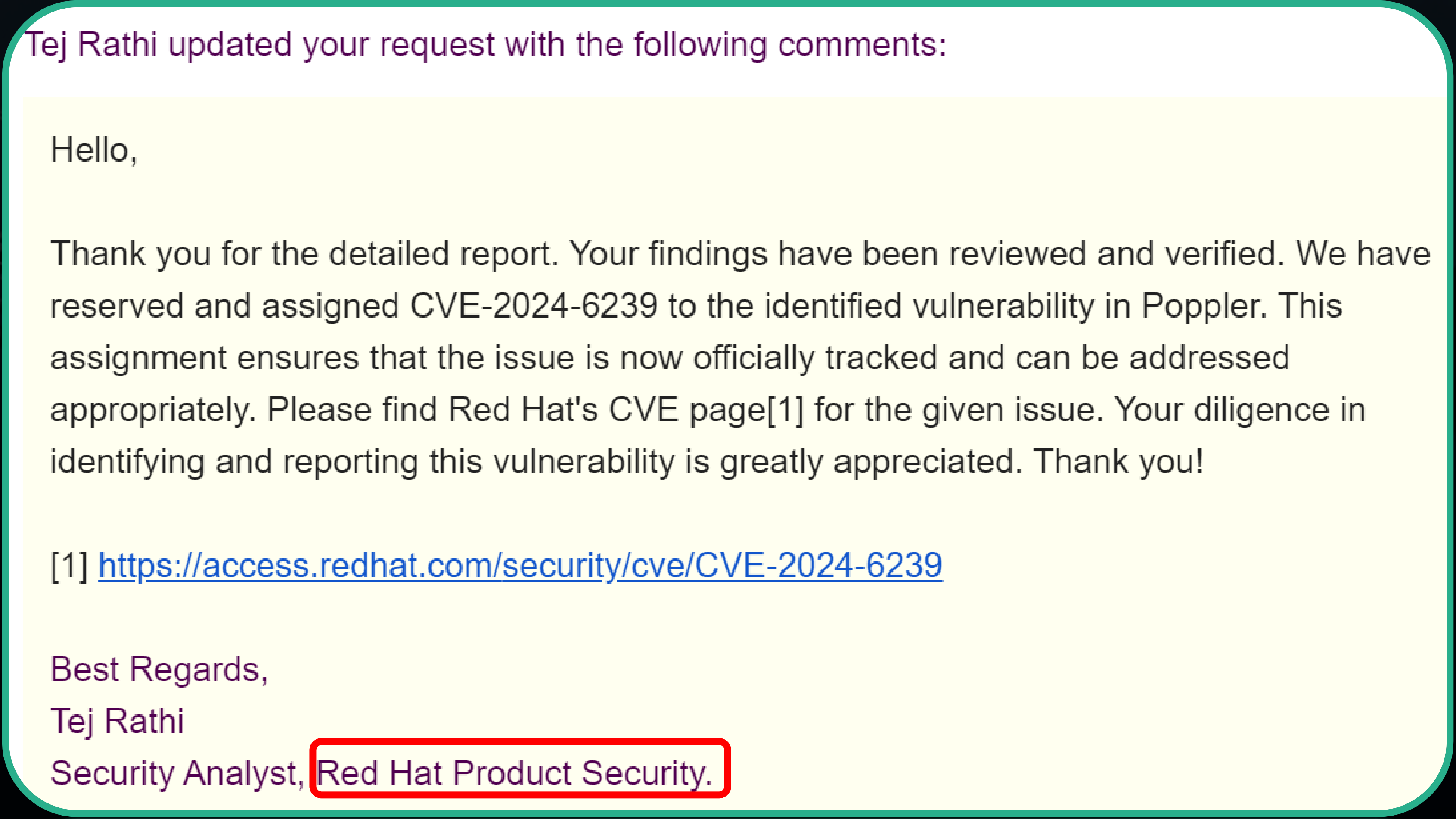
Hello,

Thank you for the detailed report. Your findings have been reviewed and verified. We have reserved and assigned CVE-2024-6239 to the identified vulnerability in Poppler. This assignment ensures that the issue is now officially tracked and can be addressed appropriately. Please find Red Hat's CVE page[1] for the given issue. Your diligence in identifying and reporting this vulnerability is greatly appreciated. Thank you!

[1] https://access.redhat.com/security/cve/CVE-2024-6239

Best Regards,
Tej Rathi
Security Analyst, Red Hat Product Security.

8. Attempting to submit CVE applications and finally obtaining a CVE



Tej Rathi updated your request with the following comments:

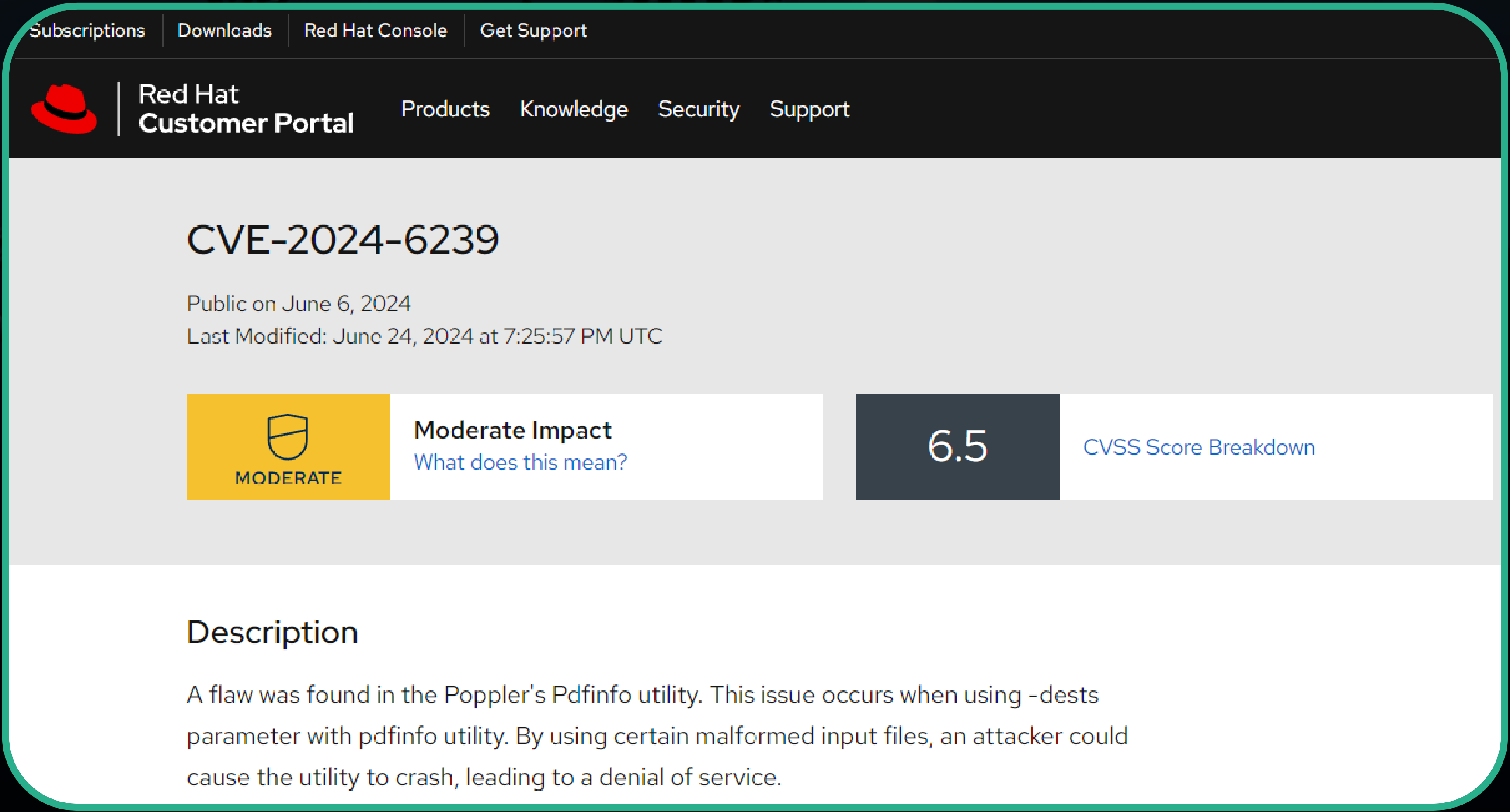
Hello,

Thank you for the detailed report. Your findings have been reviewed and verified. We have reserved and assigned CVE-2024-6239 to the identified vulnerability in Poppler. This assignment ensures that the issue is now officially tracked and can be addressed appropriately. Please find Red Hat's CVE page[1] for the given issue. Your diligence in identifying and reporting this vulnerability is greatly appreciated. Thank you!

[1] https://access.redhat.com/security/cve/CVE-2024-6239

Best Regards,
Tej Rathi
Security Analyst, Red Hat Product Security.

8. Attempting to submit CVE applications and finally obtaining a CVE



CVE-2024-6239

Public on June 6, 2024
Last Modified: June 24, 2024 at 7:25:57 PM UTC

Moderate Impact
What does this mean?

6.5

CVSS Score Breakdown

Description

A flaw was found in the Poppler's Pdfinfo utility. This issue occurs when using -dests parameter with pdfinfo utility. By using certain malformed input files, an attacker could cause the utility to crash, leading to a denial of service.

Thanks to my mentor!

Thanks for listening!

Q & A