TexLive: Use-After-Free in SyncTeX Parser - CVE-2026-63729

7 days ago by Fatih Çelik

Updated 5 days ago • 6 min read

The Story

In this post, I will be discussing a Use-After-Free (UaF) vulnerability I discovered in the SyncTeX parser; however, unfortunately, this write-up won't be extremely detailed. The discovery dates back to early 2026, and it was fixed in February 2026. Since I don't currently have the time to recreate the PoC from scratch, I will be directly embedding the original report I submitted to the vendor into this post.

The story actually began when I was fuzzing the GNOME Evince project. While fuzzing, I realized that the crash I triggered actually originated from the SyncTeX parser embedded within Evince. Shortly after, I contacted the TeX Live team and reported the vulnerability.

- **Affected versions:** Up to and including TeX Live 2025
- **Fixed version:** TeX Live 2026
- **Fix commit**: https://github.com/TeX-Live/texlive-source/commit/002dcd3eac30db5c352f53d4181737961cc7ee9a

Below is the original vulnerability report I submitted.

Heap Use-After-Free in Evince SyncTeX Parser

Summary

A Heap Use-After-Free vulnerability was discovered in the SyncTeX parser during fuzzing. The vulnerability manifests when parsing malformed SyncTeX files, leading to a read/write access to a freed memory region. This can potentially be exploited for arbitrary code execution or Denial of Service (DoS).

Vulnerability Details

- **Type**: Heap Use-After-Free
- **Component**: `synctex_parser.c` (SyncTeX Parser)
- **URL**: https://github.com/TeX-Live/texlive-source/

Vulnerable Code Analysis

The vulnerability is caused by the recursive freeing of node structures without properly updating the links (parent, sibling, friend) that point to them.

1. **Recursive Freeing:** The `_synctex_free_node` function (and similar destructors like `_synctex_free_input`) recursively frees siblings and children.

```
    if (node) {
        SYNCTEX_SCANNER_REMOVE_HANDLE_TO(node);
        SYNCTEX_WILL_FREE(node);
        synctex_node_free(__synctex_tree_sibling(node)); // <--- Recursive free of sibling
        synctex_node_free(_synctex_tree_child(node));    // <--- Recursive free of child
        _synctex_free(node);                             // <--- Free current node
    }
    return;
}
```

2. **The Vulnerability:** When `synctex_node_free(__synctex_tree_sibling(node))` is called, it frees the sibling node. However, the current `node`'s sibling pointer is **not set to NULL** before this call, nor is the sibling detached from the tree.

   If the sibling node (or any of its descendants) is referenced elsewhere (e.g., as a "friend" node, or via a `next` pointer in an iterator or alias), that reference becomes a dangling pointer.

3. **Trigger:** The malformed SyncTeX file creates a scenario where a `ref` node has no parent (i.e., `_synctex_tree_parent(ref)` returns `NULL`). Because of this missing parent, the parser's replacement routine (`__synctex_replace_ref`) fails to detach the node from its sibling chain.

   When the parser later unconditionally frees this `ref` node, its sibling pointer is still intact and pointing to the rest of the valid, live tree. The recursive destruction cascades into the live tree, prematurely freeing active sibling nodes. When the parser subsequently accesses these prematurely freed nodes, the Use-After-Free is triggered.

Reproduction

The vulnerability was verified using `evince-thumbnailer` compiled with AddressSanitizer.

Verification Steps

1. Compile the evince-thumbnailer with ASAN.
2. Run the evince-thumbnailer with the POC. I'll provide "uaf.pdf" and "uaf.synctex" files.

BASH

```
# Set environment variables to point to the built backends
export EV_BACKENDS_DIR="./build_asan/backend"
export LD_LIBRARY_PATH="./build_asan/libdocument:./build_asan/libview:$LD_LIBRARY_PATH"
export ASAN_OPTIONS=detect_leaks=0:halt_on_error=1

# Trigger the crash
./build_asan/thumbnailer/evince-thumbnailer ./uaf.pdf ./output.png
```

Crash Output

The execution confirms a Heap Use-After-Free in `synctex_node_free`:

```
0x7ffe0a07cb10 sp 0x7ffe0a07cb00
READ of size 8 at 0x506000005d80 thread T0
    #0 0x76e03618ca29 in synctex_node_free ../cut-n-paste/synctex/synctex_parser.c:288
    #1 0x76e03618ca29 in _synctex_free_leaf ../cut-n-paste/synctex/synctex_parser.c:920
    #2 0x76e03618ca29 in _synctex_free_leaf ../cut-n-paste/synctex/synctex_parser.c:916
    #3 0x76e0361b1bdb in synctex_node_free ../cut-n-paste/synctex/synctex_parser.c:288
    #4 0x76e0361b1bdb in _synctex_post_process_ref ../cut-n-paste/synctex/synctex_parser.c:5752
    #5 0x76e0361b6fe9 in _synctex_post_process ../cut-n-paste/synctex/synctex_parser.c:5877
    #6 0x76e0361b6fe9 in _synctex_scan_content ../cut-n-paste/synctex/synctex_parser.c:5958
    #7 0x76e0361b6fe9 in synctex_scanner_parse ../cut-n-paste/synctex/synctex_parser.c:6086
    #8 0x76e0361bb373 in synctex_scanner_new_with_output_file ../cut-n-paste/synctex/synctex_parser.c:6015
    #9 0x76e036170425 in ev_document_initialize_synctex ../libdocument/ev-document.c:377
    #10 0x76e036171b81 in ev_document_load_full ../libdocument/ev-document.c:433
    #11 0x76e0361760a4 in ev_document_factory_get_document_full ../libdocument/ev-document-factory.c:335
    #12 0x5582d6e84950 in evince_thumbnailer_get_document ../thumbnailer/evince-thumbnailer.c:165
    #13 0x5582d6e84950 in main ../thumbnailer/evince-thumbnailer.c:278
    #14 0x76e035a2a1c9 in __libc_start_call_main ../sysdeps/nptl/libc_start_call_main.h:58
    #15 0x76e035a2a28a in __libc_start_main_impl ../csu/libc-start.c:360
    #16 0x5582d6e850f4 in _start (/root/targets/evince/build_asan/thumbnailer/evince-thumbnailer+0x30f4) (BuildId:
59918a2795a5711a59a8f95a150e203786973f2f)

0x506000005d80 is located 0 bytes inside of 64-byte region [0x506000005d80,0x506000005dc0)
freed by thread T0 here:
    #0 0x76e0362fc4d8 in free ../../../../src/libsanitizer/asan/asan_malloc_linux.cpp:52
    #1 0x76e0361b1bdb in synctex_node_free ../cut-n-paste/synctex/synctex_parser.c:288
    #2 0x76e0361b1bdb in _synctex_post_process_ref ../cut-n-paste/synctex/synctex_parser.c:5752
    #3 0x76e0361b6fe9 in _synctex_post_process ../cut-n-paste/synctex/synctex_parser.c:5877
    #4 0x76e0361b6fe9 in _synctex_scan_content ../cut-n-paste/synctex/synctex_parser.c:5958
    #5 0x76e0361b6fe9 in synctex_scanner_parse ../cut-n-paste/synctex/synctex_parser.c:6086
    #6 0x76e0361bb373 in synctex_scanner_new_with_output_file ../cut-n-paste/synctex/synctex_parser.c:6015
    #7 0x76e036170425 in ev_document_initialize_synctex ../libdocument/ev-document.c:377
    #8 0x76e036171b81 in ev_document_load_full ../libdocument/ev-document.c:433
    #9 0x76e0361760a4 in ev_document_factory_get_document_full ../libdocument/ev-document-factory.c:335
    #10 0x5582d6e84950 in evince_thumbnailer_get_document ../thumbnailer/evince-thumbnailer.c:165
    #11 0x5582d6e84950 in main ../thumbnailer/evince-thumbnailer.c:278
    #12 0x76e035a2a1c9 in __libc_start_call_main ../sysdeps/nptl/libc_start_call_main.h:58
    #13 0x76e035a2a28a in __libc_start_main_impl ../csu/libc-start.c:360
    #14 0x5582d6e850f4 in _start (/root/targets/evince/build_asan/thumbnailer/evince-thumbnailer+0x30f4) (BuildId:
59918a2795a5711a59a8f95a150e203786973f2f)

previously allocated by thread T0 here:
    #0 0x76e0362fd9c7 in malloc ../../../../src/libsanitizer/asan/asan_malloc_linux.cpp:69
    #1 0x76e0361bd8d4 in _synctex_malloc ../cut-n-paste/synctex/synctex_parser_utils.c:72
    #2 0x76e0361b5361 in _synctex_new_ref ../cut-n-paste/synctex/synctex_parser.c:1647
    #3 0x76e0361b5361 in _synctex_new_ref ../cut-n-paste/synctex/synctex_parser.c:1647
    #4 0x76e0361b5361 in _synctex_parse_new_ref ../cut-n-paste/synctex/synctex_parser.c:4943
    #5 0x76e0361b5361 in __synctex_parse_sfi ../cut-n-paste/synctex/synctex_parser.c:5553
    #6 0x76e0361b5361 in _synctex_scan_content ../cut-n-paste/synctex/synctex_parser.c:5956
    #7 0x76e0361b5361 in synctex_scanner_parse ../cut-n-paste/synctex/synctex_parser.c:6086
    #8 0x76e0361bb373 in synctex_scanner_new_with_output_file ../cut-n-paste/synctex/synctex_parser.c:6015
    #9 0x76e036170425 in ev_document_initialize_synctex ../libdocument/ev-document.c:377
    #10 0x76e036171b81 in ev_document_load_full ../libdocument/ev-document.c:433
    #11 0x76e0361760a4 in ev_document_factory_get_document_full ../libdocument/ev-document-factory.c:335
    #12 0x5582d6e84950 in evince_thumbnailer_get_document ../thumbnailer/evince-thumbnailer.c:165
    #13 0x5582d6e84950 in main ../thumbnailer/evince-thumbnailer.c:278
    #14 0x76e035a2a1c9 in __libc_start_call_main ../sysdeps/nptl/libc_start_call_main.h:58
    #15 0x76e035a2a28a in __libc_start_main_impl ../csu/libc-start.c:360
    #16 0x5582d6e850f4 in _start (/root/targets/evince/build_asan/thumbnailer/evince-thumbnailer+0x30f4) (BuildId:
59918a2795a5711a59a8f95a150e203786973f2f)

SUMMARY: AddressSanitizer: heap-use-after-free ../cut-n-paste/synctex/synctex_parser.c:288 in synctex_node_free
Shadow bytes around the buggy address:
  0x506000005b00: 00 00 00 fa fa fa fa fa 00 00 00 00 00 00 00 00
  0x506000005b80: fa fa fa fa fd fd fd fd fd fd fd fa fa fa fa fa
  0x506000005c00: fd fd fd fd fd fd fd fa fa fa fa fa fd fd fd fd
```

```
=>0x506000005d80:[fd]fd fd fd fd fd fd fd fa fa fa fa fd fd fd fd
  0x506000005e00: fd fd fd fd fa fa fa fa fd fd fd fd fd fd fd fd
  0x506000005e80: fa fa fa fa fd fd fd fd fd fd fd fd fa fa fa fa
  0x506000005f00: fd fd fd fd fd fd fd fd fa fa fa fa fa fa fa fa
  0x506000005f80: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x506000006000: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
==1597649==ABORTING
```

Vulnerability Research

vulnerability research

Share:

Further Reading

Dec 10, 2020

Group Office CRM | Stored XSS via SVG File

Software:

https://sourceforge.net/projects/group-…

Dec 12, 2021

Deserialization of Untrusted Data in pytorch-lightning

Software:

[https://github.com/pytorchlightning/pyto…

Mar 1, 2024

Missing IP Address Control in isPublic() Function Leads to SSRF Bypass PoC

Software: NPM - Ip Package Vulnerability:

Missing IP Address Control CVE: CVE-2023-…

OLDER

Trivy: Arbitrary File Write via Path Traversal in Plugin Manager - CVE-2026-63328

NEWER

-