Source Code Audit on libjpeg-turbo
for Open Source Technology Improvement Fund (OSTIF)

Final Report and Management Summary

2023-07-12 *PUBLIC*

X41 D-SEC GmbH
Krefelder Str. 123
D-52070 Aachen
Amtsgericht Aachen: HRB19989

`https://x41-dsec.de/`
`info@x41-dsec.de`

Organized by the Open Source Technology Improvement Fund

Revision	*Date*	*Change*	*Author(s)*
1	2023-06-20	Final Report and Management Summary	MSc. H. Moesl, Dipl.-Ing. D. Gstir, R. Weinberger and D. Oberhollenzer

Contents

Dashboard

Target	
Customer	Open Source Technology Improvement Fund (OSTIF)
Name	libjpeg-turbo
Type	Source Code
Version	As deployed between 2023-04-17 and 2023-05-30
Engagement	
Type	Source Code Audit
Consultants	4: MSc. H. Moesl, Dipl.-Ing. D. Gstir, R. Weinberger and D. Oberhollenzer
Engagement Effort	25 person-days, 2023-04-17 to 2023-05-30
Total issues found	2

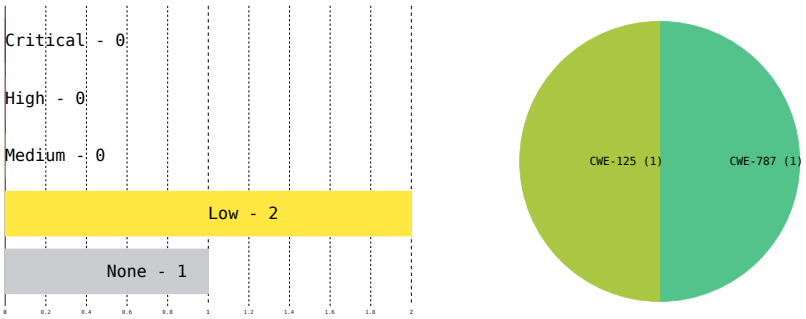

Figure 1: Issue Overview (l: Severity, r: CWE Distribution)

1 Executive Summary

In May 2023, X41 D-Sec GmbH performed a Source Code Audit against libjpeg-turbo to identify vulnerabilities and weaknesses in the source code. The test was organized by the Open Source Technology Improvement Fund (OSTIF)[1].

A total of two vulnerabilities were discovered during the test by X41. None were rated as critical, none were classified as high severity, none as medium, and two as low. Additionally, one issue without a direct security impact was identified.

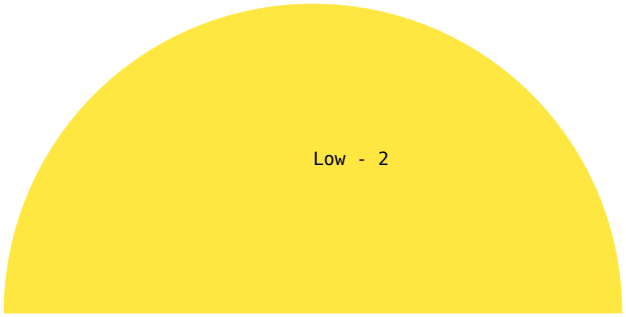

Figure 1.1: Issues and Severity

libjpeg-turbo is a JPEG image decoding software that uses SIMD instructions to accelerate baseline JPEG compression and decompression for various platforms, including x86 and ARM.

As it is common for compiled languages, a particular focus was placed on the identification of typical memory corruption vulnerabilities such as buffer overflows, information leakages, use-

[1] `https://ostif.org`

after-frees. Moreover, the testing team dedicated substantial focus and effort into examining the input validation associated with the various API calls, conducting variant analysis on previous bugs classified as security vulnerabilities, and generally reviewing memory management practices.

In a source code audit, the testers receive all available information about the target. The test was performed by four experienced security experts between 2023-04-17 and 2023-05-30.

As we conclude the audit process for the libjpeg-turbo library, it's important to highlight the excellent condition this library is in. After rigorous examination by multiple auditors, all reviewing the library independently, it is remarkable to note that very few areas of concern were identified. This signifies the strong robustness of the code, its quality, and its adherence to best practices with regard to secure programming principles.

This audit was conducted using different testing techniques and approaches, ensuring broad testing coverage. In terms of dynamic testing, the team developed several fuzz testing harnesses. Fuzz testing is, in general, essential for the overall security of the libjpeg-turbo project, especially since it is implemented in C, which is often prone to memory corruption vulnerabilities. For the purpose of this test, code coverage driven fuzz testing using AFL++ in combination with address space sanitizers (such as ASAN) was performed. It is highly recommended to incorporate the developed harnesses into the libjpeg-turbo project and to maintain the high standards and ensure that fuzzing remains an integral and fixed part of libjpeg-turbo, either using AFL++ or libFuzzer, resulting in better testing coverage.

Furthermore, both manual and tool-driven static source code analyses was performed. The former allowed the team to have a thorough and detailed understanding of the code, where the public facing API interface was under scrutiny concerning memory corruption flaws as well as secure coding standards in general. The manual auditing process was assisted by the usage of state-of-the-art tools CodeQL, which allows for advanced semantic code analysis including taint tracking and data flow analysis, adding an extra layer of comprehension to this audit.

In total, X41 was able to identify three issues through fuzz testing:

1. *tj3Decompress*: Out-of-bounds read in 2:1 upsampling code
2. *tj3Transform*: Memory corruption
3. *tj3Transform*: Out-of-bounds write

Again, it must be reiterated that this assessment provided valuable insights into the security posture at the time of testing, but it is important to note that any code audit is unable to guarantee that the software complex is free of additional bugs.

To sum up, the results of this audit speak volumes about the soundness of the libjpeg-turbo library.

It deserves high praise to find a library that stands up so well under such rigorous and diverse testing approaches. The combination of manual and automated testing methods, in conjunction with the use of tools like CodeQL, has ensured a level of scrutiny that leaves X41 confident in the resilience and reliability of libjpeg-turbo library. Nevertheless, due to the widespread usage of libjpeg-turbo, it is encouraged to perform recurring security audits, because new vulnerabilities may be introduced as more features are added and also changes within one part of the system may have unintentional security impact to other parts.

2 Introduction

The assessment comprised a security review of the *libjpeg-turbo* library, utilizing static source code analysis as well as dynamic testing using dedicated fuzz testing harnesses. The branch in scope for this inspection was the `main` branch with the commit id:

- `3a53627306233013dcec61a90f0e9ed302ea5156`

The main objective of this security assessment was the identification of vulnerabilities within the libjpeg-turbo code base. It must be noted that no communication channel with the maintainer of the libjpeg-turbo was available. Therefore, the testing team was unable to define specific areas of interest upfront.

From a programming style and software design perspective the code and design is clean and very well written, with security in mind.

2.1 Findings Overview

DESCRIPTION	SEVERITY	ID	REF
Out of Bounds Write in tj3Transform() for Non-Resizeable, Pre-Allocated Buffers	LOW	LJPGT-PT-23-01	4.1.1
Out-of-bounds Read in 2:1 Upsampling Code	LOW	LJPGT-PT-23-02	4.1.2
Memory Corruption in tj3Transform()	NONE	LJPGT-PT-23-100	4.2.1

Table 2.1: Security-Relevant Findings

2.2 Scope

Due to the absence of specifically outlined focus areas, the evaluation centered around a broad inspection, primarily looking for common memory corruption vulnerabilities.

2.3 Coverage

A security assessment attempts to find the most important or sometimes as many of the existing problems as possible, though it is practically never possible to rule out the possibility of additional weaknesses being found in the future.

A manual approach for code review is used by X41. This process was combined with fuzzing given the nature of libjpeg-turbo being exposed to parsing of potentially untrustworthy data.

The time allocated to X41 for this code review was sufficient to yield a reasonable coverage of the given scope.

2.3.1 Fuzzing

While conducting a source audit of the libjpeg-turbo library, it was found that some fuzz testing harnesses already exist and are part of OSS-Fuzz. Nevertheless, as not all API[1] functions had undergone fuzzing via OSS-Fuzz, X41 made the decision to focus on conducting further fuzzing.

[1] Application Programming Interface

For the fuzz testing, AFL++ [2] was used in two ways/modes:

- `argv` (command-line) fuzzing of the `cjpeg` and `djpeg`tools utilizing AFL++ persistent mode
- Fuzzing of selected interesting looking functions of libjpeg-turbo utilizing AFL++ persistent mode

Persistent mode fuzzing is a feature in the AFL++ fuzzer that keeps the target program running in the background and continuously feeds it with new test cases. This is in contrast to the default "one-shot" mode in which the fuzzer launches the target program with each new test case. By utilizing this approach, it was possible to achieve execution speed improvements of 10 to 20 times. Moreover, AFL++ has recently incorporated support for command-line interface (CLI[3]) fuzzing in persistent mode through the `AFL_INIT_ARGV_PERSISTENT` macro, rendering it an ideal choice for the CLI fuzzing of the `cjpeg` and `djpeg` CLI binaries.

2.3.1.1 Fuzzing Hardware

The fuzzing process was carried out on a system equipped with an AMD[4] Ryzen Threadripper Processor, which boasts 64 cores and 128GB of RAM[5].

2.3.1.2 CLI Fuzzing

Considering that libjpeg-turbo comprises various tools, such as `cjpeg` and `djpeg`, as a component of its code base, X41, decided to conduct command-line interface (CLI) fuzzing against these tools to detect any bugs associated with the parsing of `argv` parameters.

To conduct CLI fuzzing on each of the aforementioned tools, X41 generated a valid set of CLI parameters and utilized them as input test cases for the fuzzer. Additionally, X41 built the code base with address sanitization enabled (`-fsanitize=address`) to detect any memory management errors. To facilitate the fuzzer in quickly finding valid CLI parameters, X41 configured the AFL++ compiler to create a dictionary using the `AFL_LLVM_DICT2FILE` flag based on the compiled C code.

The fuzzer executed each of the aforementioned tools for a total of approximately 5 billion times. Despite the extensive number of executions, X41 was unable to identify any immediate crashes. Hence, it can be concluded that the code base for these tools and libjpeg-turbo is well tested

[2] `https://github.com/AFLplusplus/AFLplusplus/`
[3] Command-line Interface
[4] Advanced Micro Devices
[5] Random Access Memory

and written with security in mind, at least under the conditions and parameters we used for the fuzzing process.

However, it is worth noting that the absence of immediate crashes does not necessarily imply that the code is free from bugs or vulnerabilities.

2.3.1.3 Fuzzing of Selected Functions

During this project, X41 created fuzzing harnesses for the following functions:

- functions doing compression:
  - ***tj3Compress8()***
  - ***tj3Compress12()***
  - ***tj3Compress16()***
- functions doing JPEG[6] decompression:
  - ***tj3Decompress8()***
  - ***tj3Decompress12()***
  - ***tj3Decompress16()***
- functions doing JPEG transformations:
  - ***tj3Transform()***

The folder `testimages` already contained input test cases provided by OSS-Fuzz, which were utilized as a starting point for generating test cases using the `radamsa` tool[7]. X41 compiled the source code base with address sanitization enabled (`-fsanitize=address`).

The fuzzing harnesses executed each of the aforementioned functions approximately 5 billion times, but no memory memory corruptions or crashes were detected during this process.

Fuzzing the function `tj3Decompress12()` resulted in an out-of-bounds read, which is documented in the finding 4.1.1. This crash has also been identified mid-project by an independent security researcher[8].

Further, fuzzing of `tj3Transform()` revealed a memory corruption issue when executing JPEG transform operations. This is described in finding 4.2.1.

[6] Joint Photographic Experts Group
[7] `https://gitlab.com/akihe/radamsa`
[8] `https://github.com/libjpeg-turbo/libjpeg-turbo/issues/690`

2.4 Recommended Further Tests

X41 recommends to subject all newly developed code to regular source code audits. Given the complexity of the jpeg-turbo library, the code base would benefit from recurring security audits as changes within one part of the system may have unintentional security impact to other parts.

3 Rating Methodology for Security Vulnerabilities

Security vulnerabilities are given a purely technical rating by the testers as they are discovered during the test. Business factors and financial risks for Open Source Technology Improvement Fund (OSTIF) are beyond the scope of a code audit which focuses entirely on technical factors. Yet technical results from a code audit may be an integral part of a general risk assessment. A code audit is based on a limited time frame and only covers vulnerabilities and security issues which have been found in the given time, there is no claim for full coverage.

In total, five different ratings exist, which are as follows:

Severity Rating
None
Low
Medium
High
Critical

A low rating indicates that the vulnerability is either very hard for an attacker to exploit due to special circumstances, or that the impact of exploitation is limited, whereas findings with a medium rating are more likely to be exploited or have a higher impact. High and critical ratings are assigned when the testers deem the prerequisites realistic or trivial and the impact significant or very significant.

Findings with the rating 'none' are called informational findings and are related to security hardening, affect functionality, or other topics that are not directly related to security. X41 recommends to mitigate these issues as well, because they often become exploitable in the future. Doing so will strengthen the security of the system and is recommended for defense in depth.

3.1 Common Weakness Enumeration

The CWE[1] is a set of software weaknesses that allows the categorization of vulnerabilities and weaknesses in software. If applicable, X41 provides the CWE-ID for each vulnerability that is discovered during a test.

CWE is a very powerful method to categorize a vulnerability and to give general descriptions and solution advice on recurring vulnerability types. CWE is developed by *MITRE*[2]. More information can be found on the CWE website at `https://cwe.mitre.org/`.

[1] Common Weakness Enumeration
[2] `https://www.mitre.org`

4 Results

This chapter describes the results of this test. The security-relevant findings are documented in Section 4.1. Additionally, findings without a direct security impact are documented in Section 4.2.

4.1 Findings

The following subsections describe findings with a direct security impact that were discovered during the test.

4.1.1 LJPGT-PT-23-01: Out of Bounds Write in tj3Transform() for Non-Resizeable, Pre-Allocated Buffers

Severity:	LOW
CWE:	787 – Out-of-bounds Write
Affected Component:	turbojpeg.c:tj3Transform()

4.1.1.1 Description

In ***tj3Transform()***, if the buffer is pre-allocated by the user and the function is instructed to not resize it, the buffer size is overwritten with a maximum computed value using ***tj3JPEGBufSize()***:

```
1     if (this->noRealloc) {
2       alloc = FALSE;  dstSizes[i] = tj3JPEGBufSize(w, h, this->subsamp);
3     }
```

Listing 4.1: Buffer Size Recomputed

If the image is, e.g., rotated by 90 or 270 degrees, width and height are to be swapped and ***tj3JPEGBufSize()*** returns different results for transposed images.

The existing fuzzing code adheres this as well:

```
1   transforms[2].op = TJXOP_ROT90;
2   transforms[2].options = TJXOPT_TRIM | TJXOPT_COPYNONE | TJXOPT_ARITHMETIC;
3   dstBufs[2] =
4     (unsigned char *)malloc(tj3JPEGBufSize(height, width, jpegSubsamp));
```

Listing 4.2: Fuzzing Code in transform.cc

As a result, the assumed buffer size differs from the real size, leading to out-of-bounds writes when generating the transformed image.

This also affects code that uses a pre-allocated buffer that is too small for any other reason, as the buffer size passed to the function is ignored and replaced with an assumed maximum:

```
#include <turbojpeg.h>
#include <stdlib.h>
#include <stdint.h>
#include <assert.h>
#include <stdio.h>

void do_test(unsigned char *data, size_t size)
{
    tjhandle handle = NULL;
    unsigned char *dstBuf[1] = {NULL};
    size_t dstSizes[1] = {0};
    size_t maxBufSize = 0;
    int width = 0, height = 0, jpegSubsamp;
    tjtransform transform = {0};

    handle = tj3Init(TJINIT_TRANSFORM);
    assert(handle != NULL);

    assert(tj3DecompressHeader(handle, data, size) == 0);

    width = tj3Get(handle, TJPARAM_JPEGWIDTH);
    height = tj3Get(handle, TJPARAM_JPEGHEIGHT);
    jpegSubsamp = tj3Get(handle, TJPARAM_SUBSAMP);

    assert(width >= 1);
    assert(height >= 1);
    assert(jpegSubsamp < TJ_NUMSAMP);
    assert((uint64_t)width * height < 1048576);

    transform.op = TJXOP_ROT90;

    maxBufSize = tj3JPEGBufSize(height, width, jpegSubsamp);
    assert(maxBufSize > 0);

    dstBuf[0] = malloc(maxBufSize);
    assert(dstBuf[0] != NULL);

    //dstSizes[0] = maxBufSize;

    assert(tj3Set(handle, TJPARAM_NOREALLOC, 1) == 0);

    tj3Transform(handle, data, size, 1, dstBuf, dstSizes, &transform);

    free(dstBuf[0]);
    tj3Destroy(handle);
}

int main(int argc, char **argv)
```

```
49  {
50      unsigned char *imgdata;
51      long sz, nread;
52      FILE *fp;
53
54      assert(argc == 2);
55      fp = fopen(argv[1], "rb");
56      assert(fp != NULL);
57
58      assert(fseek(fp, 0, SEEK_END) == 0);
59      sz = ftell(fp);
60      assert(sz > 0);
61      assert(fseek(fp, 0, SEEK_SET) == 0);
62
63      imgdata = malloc(sz);
64      assert(imgdata != NULL);
65
66      nread = fread(imgdata, 1, sz, fp);
67      assert(nread == sz);
68      fclose(fp);
69
70      do_test(imgdata, sz);
71      free(imgdata);
72      return EXIT_SUCCESS;
73  }
```

Listing 4.3: Reproducer Code for the *tj3Transform()* Crash

The input file used by the reproducer code in listing 4.3 is contained in the appendix (see section A.1).

4.1.1.2 Solution Advice

X41 recommends to either take the transformation into account when computing the buffer size, or simply refrain from overwriting the buffer size argument, and require it to be non-zero if the *TJFLAG_NOREALLOC* flag is set.

4.1.2 LJPGT-PT-23-02: Out-of-bounds Read in 2:1 Upsampling Code

Severity:	LOW
CWE:	125 – Out-of-bounds Read
Affected Component:	jdmrgext.c:h2v2_merged_upsample_internal()

4.1.2.1 Description

An out-of-bounds read error in function ***h2v2_merged_upsample_internal()*** was discovered by the testing harness, which can be triggered by malformed JPEG files.

The fuzz testing produced three slightly different samples of lossless JPEG files (see appendix) which reproduce this issue using the `djpeg` CLI utility. When combined with code that prints the retrieved data, this can potentially cause in an information leakage.

However, due to time limitations, a thorough examination of whether this is feasible in this particular case was not conducted.

One can pertinently note that this issue was independently discovered and reported on GitHub by the user Shin-Yan [1].

The listing 4.4 shows the stack trace produced by samples 1 and 2:

```
1  $ ./djpeg-static -fast ../../libjpeg-turbo-decompress/case1.jpg
2  P6
3  1 1
4  4095
5  Premature end of JPEG file
6  AddressSanitizer:DEADLYSIGNAL
7  =================================================================
8  ==22626==ERROR: AddressSanitizer: SEGV on unknown address 0x62effffa194
9      (pc 0x55ed867b010d bp 0x7fffaf299d90 sp 0x7fffaf299cf0 T0)
10 ==22626==The signal is caused by a READ memory access.
11     #0 0x55ed867b010d in h2v2_merged_upsample_internal
       ↪   /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/jdmrgext.c
12     #1 0x55ed867b010d in h2v2_merged_upsample
       ↪   /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/jdmerge.c:383:5
13     #2 0x55ed867a677f in merged_2v_upsample
       ↪   /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/jdmerge.c:260:5
14     #3 0x55ed867803d1 in process_data_simple_main
       ↪   /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/jdmainct.c:317:3
15     #4 0x55ed8675b1a8 in jpeg12_read_scanlines
       ↪   /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/jdapistd.c:335:3
```

[1] https://github.com/libjpeg-turbo/libjpeg-turbo/issues/690

```
16      #5 0x55ed866e7b75 in
        ↪  main /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/djpeg.c:868:25
17      #6 0x7f4cfd04129c in __libc_start_main⌋
        ↪   (/lib64/libc.so.6+0x3529c) (BuildId: c8417d767baccfadb39b474e484d46947915cd8f)
18      #7 0x55ed86624359 in
        ↪  _start /home/abuild/rpmbuild/BUILD/glibc-2.31/csu/../sysdeps/x86_64/start.S:120
19
20  AddressSanitizer can not provide additional info.
21  ⌋
    ↪  SUMMARY: AddressSanitizer: SEGV /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/jdmrgext.c
22      in h2v2_merged_upsample_internal
```

Listing 4.4: Asan Output Produced by Sample 1 (Sample 2 Produces the Same Stack Trace)

Listing 4.5 shows the stack trace produced by sample 3:

```
$ ./djpeg-static -fast ../../libjpeg-turbo-decompress/case3.jpg
P6
64 1
4095
Premature end of JPEG file
AddressSanitizer:DEADLYSIGNAL
=================================================================
==9800==ERROR: AddressSanitizer: SEGV on unknown address 0x62effffa194
    (pc 0x5606f23ccdcd bp 0x7ffdd6fbec70 sp 0x7ffdd6fbebd0 T0)
==9800==The signal is caused by a READ memory access.
    #0 0x5606f23ccdcd in h2v2_merged_upsample_internal
    ↪  /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/jdmrgext.c
    #1 0x5606f23ccdcd in h2v2_merged_upsample
    ↪  /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/jdmerge.c:383:5
    #2 0x5606f23c877f in merged_2v_upsample
    ↪  /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/jdmerge.c:260:5
    #3 0x5606f23a23d1 in process_data_simple_main
    ↪  /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/jdmainct.c:317:3
    #4 0x5606f237d1a8 in jpeg12_read_scanlines
    ↪  /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/jdapistd.c:335:3
    #5 0x5606f2309b75 in
    ↪  main /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/djpeg.c:868:25
    #6 0x7fa7d7f9829c in __libc_start_main
    ↪  (/lib64/libc.so.6+0x3529c) (BuildId: c8417d767baccfadb39b474e484d46947915cd8f)
    #7 0x5606f2246359 in
    ↪  _start /home/abuild/rpmbuild/BUILD/glibc-2.31/csu/../sysdeps/x86_64/start.S:120

AddressSanitizer can not provide additional info.

↪  SUMMARY: AddressSanitizer: SEGV /home/dags/work/rootsys/libjpeg-turbo_2023-05/libjpeg-turbo/jdmrgext.c
    in h2v2_merged_upsample_internal
==9800==ABORTING
```

Listing 4.5: Asan Output Produced by Sample 3

In all samples the faulty read occurs when accessing ***cinfo->sample_range_limit*** for the red color. The inputs leading to the issue and the AFL++ test harness can be found in the appendix (see section A.4).

4.1.2.2 Solution Advice

X41 recommends adding boundary checks to mitigate the risk of reading out-of-bounds.

4.2 Informational Notes

The following observations do not have a direct security impact, but are related to security hardening, affect functionality, or other topics that are not directly related to security. X41 recommends to mitigate these issues as well, because they often become exploitable in the future. Doing so will strengthen the security of the system and is recommended for defense in depth.

4.2.1 LJPGT-PT-23-100: Memory Corruption in tj3Transform()

Affected Component:	turbojpeg.c:tj3Transform()

4.2.1.1 Note

This informational note was previously a finding. It was re-rated because the API was used incorrectly.

4.2.1.2 Description

In ***tj3Transform()***, if the buffer is pre-allocated by the user and the function is instructed to not resize it, a memory corruption error is encountered.

This error occurs on any valid JPEG transform operation (except *TJXOP_NONE*) and results in the following GDB[2] output:

```
double free or corruption (out)

Program received signal SIGABRT, Aborted.
0x00007f617116ea7c in pthread_kill () from /lib/x86_64-linux-gnu/libc.so.6
#0  0x00007f617116ea7c in pthread_kill () from /lib/x86_64-linux-gnu/libc.so.6
#1  0x00007f617111a476 in raise () from /lib/x86_64-linux-gnu/libc.so.6
#2  0x00007f61711007f3 in abort () from /lib/x86_64-linux-gnu/libc.so.6
#3  0x00007f61711616f6 in ?? () from /lib/x86_64-linux-gnu/libc.so.6
#4  0x00007f617178d7c in ?? () from /lib/x86_64-linux-gnu/libc.so.6
#5  0x00007f617117aef0 in ?? () from /lib/x86_64-linux-gnu/libc.so.6
#6  0x00007f617117d4d3 in free () from /lib/x86_64-linux-gnu/libc.so.6
#7  0x000055e705db99d1 in free_pool (cinfo=0x55e706f89a98, pool_id=<optimized out>)
    at /src/jmemmgr.c:1142
#8  0x000055e705da99fd in jpeg_abort (cinfo=0x944) at /src/jcomapi.c:44
#9  0x000055e705d7faf6 in tj3Transform (handle=handle@entry=0x55e706f89890,
```

[2] GNU Debugger

```
16          jpegBuf=jpegBuf@entry=0x7ffe462fd7b3 "\377\330\377\340", jpegSize=<optimized out>,
            ↪  n=n@entry=1,
17          dstBufs=dstBufs@entry=0x7ffe462fd730, dstSizes=dstSizes@entry=0x7ffe462fd748,
            ↪  t=<optimized out>)
18          at /src/turbojpeg.c:2771
19      #10 0x000055e705d6f7bd in do_test (data=0x7ffe462fd7b3 "\377\330\377\340",
20          data@entry=0x7ffe462fd7b0 "\353\212o\377\330\377\340", size=3807, size@entry=3810)
21          at /src/afl_fuzz/fuzz_transform.c:117
22      #11 0x000055e705d6fb05 in main (argc=<optimized out>, argv=<optimized out>)
23          at /src/afl_fuzz/fuzz_transform.c:215
```

Listing 4.6: Crash Output

The crash was detected through the use of specially crafted fuzz testing harnesses, leveraging AFL++. However, due to the limited time scope of the evaluation, it wasn't possible to pinpoint the root cause of this crash.

The crashing input and the AFL++ test harness can be found in the appendix (see section A.2).

4.2.1.3 Solution Advice

X41 recommends resolving this memory corruption issue after conducting further analysis to identify the root cause of this issue.

5 About X41 D-Sec GmbH

X41 D-Sec GmbH is an expert provider for application security and penetration testing services. Having extensive industry experience and expertise in the area of information security, a strong core security team of world-class security experts enables X41 D-Sec GmbH to perform premium security services.

X41 has the following references that show their experience in the field:

- Source code audit of the Git source code version control system[1]
- Review of the Mozilla Firefox updater[2]
- X41 Browser Security White Paper[3]
- Review of Cryptographic Protocols (Wire)[4]
- Identification of flaws in Fax Machines[5,6]
- Smartcard Stack Fuzzing[7]

The testers at X41 have extensive experience with penetration testing and red teaming exercises in complex environments. This includes enterprise environments with thousands of users and vendor infrastructures such as the Mozilla Firefox Updater (Balrog).

Fields of expertise in the area of application security encompass security-centered code reviews, binary reverse-engineering and vulnerability-discovery. Custom research and IT security consulting, as well as support services, are the core competencies of X41. The team has a strong technical background and performs security reviews of complex and high-profile applications such as Google Chrome and Microsoft Edge web browsers.

X41 D-Sec GmbH can be reached via `https://x41-dsec.de` or `mailto:info@x41-dsec.de`.

[1] `https://x41-dsec.de/security/research/news/2023/01/17/git-security-audit-ostif/`
[2] `https://blog.mozilla.org/security/2018/10/09/trusting-the-delivery-of-firefox-updates/`
[3] `https://browser-security.x41-dsec.de/X41-Browser-Security-White-Paper.pdf`
[4] `https://www.x41-dsec.de/reports/Kudelski-X41-Wire-Report-phase1-20170208.pdf`
[5] `https://www.x41-dsec.de/lab/blog/fax/`
[6] `https://2018.zeronights.ru/en/reports/zero-fax-given/`
[7] `https://www.x41-dsec.de/lab/blog/smartcards/`

Acronyms

A Appendix

A.1 Fuzzing Harnesses

For additional coverage, this section provides all test harnesses used during the fuzzing campaign of the `libjpeg-turbo` library.

A.1.0.1 Fuzzing of tj3Transform() - Out of Bounds Write

```
00000000: ffd8 ffe0 0010 4a46 3244 3232 3232 3201  ......JF2D22222.
00000010: 0001 0000 ffdb 0043 0008 0606 0700 0004  .......C........
00000020: 0007 0709 0908 0a0c 140d 0c0b 0b0c 1912  ................
00000030: 130f 5f1d 1a1f 1e1d 1a1c 1c20 242e 2720  .._........ $.'
00000040: 2211 2933 3232 3331 3232 8032 3232 1f27  ".)3223122.222.'
00000050: 393d 381a 3c05 feff 05ff db00 4301 0509  9=8.<.......C...
00000060: 090c 0b0c 180d 0d18 3221 1c21 3232 6a25  ........2!.!22j%
00000070: 3232 3232 0c40 327f 1c28 3729 2c30 3134  2222.@2..(7),014
00000080: 3434 3232 3232 3232 3203 0303 0303 f803  442222222.......
00000090: 0303 0332 3232 3232 3232 3232 3232 ffc9  ...22222222222..
000000a0: 0011 0c00 0100 0d03 0112 0002 1101 0311  ................
000000b0: 01ff cc00 0a00 1010 0514 1011 05ff da00  ................
000000c0: 0c03 0100 0211 03eb 8a13 ffd8 ffe0 ff00  ................
000000d0: 4a46 3232 3232 3232 3201 0001 0000 ffdb  JF2222222.......
000000e0: 0004 0007 0709 0908 0a0c 140d 0c0b 0b0c  ................
000000f0: 1912 130f 5f1d 1a1f 1e1d 1a1c 1c20 242e  ...._........ $.
00000100: 2720 222c eb8a 5aff d8ff e000 104a 4632  ' ",..Z......JF2
00000110: 4432 3232 3232 0060 00ff d8ff e000 104a  D22222.`.......J
00000120: 4652 4600 0101 0000 0100 0100 00ff db00  FRF.............
00000130: 4300 0806 0607 0605 0807 0707 0909 080a  C...............
00000140: 0c14 0d0c 0b0b 0c19 1213 0f14 1d1a 1f1e  ................
00000150: 1d1a 1c1c 2024 2e27 2022 2c23 1c1c 2837  .... $.' ",#..(7
00000160: 292c 3031 3432 341f 2739 3d38 323c 2e33  ),01424.'9=82<.3
00000170: 3432 ffdb 0043 0109 0909 0c0b 0c18 0d0d  42...C..........
00000180: 1832 211c 2132 3232 3232 3232 3232 3232  .2!.!22222222222
00000190: 3232 3232 3232 3232 3232 3232 3232 3232  2222222222222222
000001a0: 3232 3232 3232 3232 3232 3232 3232 3232  2222222222222222
000001b0: 3232 3232 3232 32ff c100 1108 0095 00e3  2222222.........
000001c0: 0301 2200 0211 0103 1101 ffcc 000a 0010  .."."...........
000001d0: 1005 0110 1105 ffda 000c 0301 0002 1103  ................
```

```
000001e0: 1100 3f00 ff00 bab9 3d0c dd51 ad02 5dd3  ..?.....=..Q..].
000001f0: b0cf b3d9 b612 7b36 81b9 3553 5cd4 bacc  ......{6..5S\...
00000200: 5234 b41c 668d 4ce2 c7b8 b681 ed46 f258  R4..f.L......F.X
00000210: aed9 03ab 77cc cad1 f44b 45a9 ee66 2606  ....w....KE..f&.
00000220: f9d0 e36a d82d 06e2 a8c2 0318 ecc0 8d36  ...j.-.........6
00000230: 241b 5fcc eab0 b995 c3a8 b052 3948 1fcc  $._........R9H..
00000240: 9fe7 8c3e ad7f 5b6e 3d89 f874 68fc 2d80  ...>..[n=..th.-.
00000250: 13a5 e748 7226 44b8 3d69 06c2 2c7b b96d  ...Hr&D.=i..,{.m
00000260: 5bd4 f614 0133 df0f e566 3abc 0cab 8f4c  [....3...f:....L
00000270: f591 7c23 7ff2 c0b4 af0f 8a16 fed9 b318  ..|#............
00000280: a24f 9404 bf8a f0c1 8ca6 01a5 d1a1 32af  .O............2.
00000290: 4f5f d5f3 9453 ae16 26af ca02 5042 6f1b  O_...S..&...PBo.
000002a0: 578f 81bb 4d1d 6e1d efe1 27f8 f14c 8577  W...M.n...'..L.w
000002b0: f50a 923a 3ba0 8ff9 700e dba0 31f9 85f2  ...:;...p...1...
000002c0: e7fc 2eb3 5798 ef2c e774 3819 a1a8 83e3  ....W..,.t8.....
000002d0: 4f4e f958 bbe6 0eaf 4514 3fe8 7993 cfce  ON.X....E.?.y...
000002e0: 3942 5bb6 b97e 6a5f 2241 30e9 7548 326b  9B[..~j_"A0.uH2k
000002f0: b9d5 061e c316 641c 80c1 3b4f 8987 0ca7  ......d...;O....
00000300: fba4 b1f9 92a0 1b92 ccdf 0934 5b4a 5af2  ...........4[JZ.
00000310: 0067 4c1a 6f16 47cf 9bdd 536f 9428 9870  .gL.o.G...So.(.p
00000320: 635f 4930 ed1f 8852 f9d0 1ce6 b3d9 b612  c_I0...R........
00000330: 3349 c0cb 3ef0 a0d2 641b 2d51 3280 8095  3I..>...d.-Q2...
00000340: 86fc 2715 ba91 eb80 938d 3cf3 0e1e fc82  ..'.......<.....
00000350: ed88 6c53 61ef eeb3 c370 254e c662 3333  ..lSa....p%N.b33
00000360: cf51 7a30 4a2c 499d 9c8c 58de 49fe f5cc  .Qz0J,I...X.I...
00000370: 2186 1db7 8859 31e4 d3c5 56f7 288e 3768  !....Y1...V.(.7h
00000380: e2c1 0db9 23bf 6d1f eeba bb52 f19a 4a54  ....#.m....R..JT
00000390: d167 3c37 1b81 d8f0 cb1d 4dc1 8291 5ec4  .g<7......M...^.
000003a0: 5642 7046 e40d 9e48 9e87 6e11 f2e0 cb21  VBpF...H..n....!
000003b0: 0e71 4943 ade8 01ac 3508 d3b2 e240 fb66  .qIC....5....@.f
000003c0: b8e7 2ace 8f4f efd9 fab5 6515 1e62 2e90  ..*..O....e..b..
000003d0: 5a1c 6cea 45c6 0a58 3fb9 98d3 4456 a610  Z.l.E..X?...DV..
000003e0: a37d 1056 c5bc 1029 e4da db5b 532d 43f0  .}.V...)...[S-C.
000003f0: e596 7497 fdf5 4e77 fd14 48ae c5f2 9f26  ..t...Nw..H....&
00000400: 320e ed26 ed16 71c8 83e9 ce7a 3951 a71b  2..&..q....z9Q..
00000410: 5c1d c3fe 107e cdd3 12b2 7061 777f 6bd1  \....~....paw.k.
00000420: 467a 1d4a b94e 05c4 5710 296d 475d fdba  Fz.J.N..W.)mG]..
00000430: 90a1 1943 ebfd f3ba 1f95 bec0 fac3 2138  ...C..........!8
00000440: fcc3 8022 2a63 1460 4d31 287d 48cc dca7  ..."*c.`M1(}H...
00000450: caeb 6a45 e1c0 8b3c 9580 f5b9 e89c 238d  ..jE...<......#.
00000460: 2964 0788 20fc caf4 af34 4bd7 6e91 b504  )d.. ....4K.n...
00000470: a9a6 60ef 3866 cf03 cbc1 eed6 a689 933f  ..`.8f.........?
00000480: 74c8 a8f6 2b51 7771 c2b3 a82a c20e 5065  t...+Qwq...*..Pe
00000490: 5cc3 c7fe 7501 68a1 7568 f6d1 27c8 32ef  \...u.h.uh..'.2.
000004a0: f6c8 2962 2a6e 40f8 e9d6 00b0 5795 ae33  ..)b*n@.....W..3
000004b0: ca67 1cae 921a db39 3b15 66b4 cf63 dd47  .g.....9;.f..c.G
000004c0: 120a b16e 698d 6853 74b6 a0a0 514f f376  ...ni.hSt...QO.v
000004d0: f983 4917 63bc 0135 918c aa4f 099f ec38  ..I.c..5...O...8
000004e0: 86e5 dba0 9e39 dc80 51ca c985 6351 1d5e  .....9..Q...cQ.^
000004f0: de0b 581c 4ccb 0791 1c0c d857 b894 64cf  ..X.L......W..d.
00000500: 9fe5 6541 2d8f a072 040e ecca 5ba0 0b1a  ..eA-..r....[...
00000510: d6e8 5405 51c3 db1c e6a6 a927 109a 1e83  ..T.Q......'....
```

```
00000520: dde2 634d 9d1e 161b 530e 80c6 d388 b986  ..cM....S.......
00000530: 3960 e93c 6c37 8dd7 3bf6 668e 5383 ccfe  9`.<l7..;.f.S...
00000540: 9e29 7ab2 7468 7fa8 cc85 e8de 94c8 3dfb  .)z.th........=.
00000550: b0a6 c340 6518 fb10 e438 c77a f072 3ecf  ...@e....8.z.r>.
00000560: b81a f258 6d6a 7fa5 eefb 258d bf92 ab21  ...Xmj....%....!
00000570: 237f bace 20c1 d642 f7aa e464 2b0f 318c  #... ..B...d+.1.
00000580: a00f f9a9 64ea e768 be35 e82e ac64 cb94  ....d..h.5...d..
00000590: 1507 bd5a 9774 a823 4e83 17d4 9f72 2106  ...Z.t.#N....r!.
000005a0: b217 3f92 d433 99d5 6607 9bc9 45d8 2a7c  ..?..3..f...E.*|
000005b0: 8ac4 8c2c 24cc f8ef c8fc 8904 3643 8069  ...,$.......6C.i
000005c0: 8a9c 3707 83a9 d762 c1b2 af14 9526 598e  ..7....b.....&Y.
000005d0: 8465 29d1 8432 5ee3 ec12 eb94 0366 1a0d  .e)..2^......f..
000005e0: 2668 1623 bfcb 4283 7f1b 3908 b683 df37  &h.#..B...9....7
000005f0: 6092 2928 4dbb b20b a11d f9d3 6903 3eef  `.)(M.......i.>.
00000600: 7508 8afd 47af 03a0 8415 ec48 fb35 240d  u...G......H.5$.
00000610: 4708 29d7 e38f 29c2 e5fa 0217 09ea 0b67  G.)...)........g
00000620: 831c 383f 1b3a 239a fbb2 6175 8b1e 55fb  ..8?.:#...au..U.
00000630: eb9b 1fa2 3762 6698 8881 96cc c80e fda7  ....7bf.........
00000640: cd0b 3b1f 74f7 c1a2 4e20 3029 7059 4a19  ..;.t...N 0)pYJ.
00000650: 4d4c 6593 e7f5 b09e c1cc 419b f6f1 78c9  MLe.......A...x.
00000660: 2517 8a8b 7b38 a968 1117 e8e5 753e e061  %...{8.h....u>.a
00000670: 1613 c8b0 977a 6e8f fc50 2cc4 fde8 2a4a  .....zn..P,...*J
00000680: 63dc b13d 9512 89e8 2000 0000 f2ad aba0  c..=.... .......
00000690: 3c82 bac4 4136 5050 42b1 38dd 58bc fd31  <...A6PPB.8.X..1
000006a0: 8fc8 c600 a2e2 1f6d 3954 8498 2196 3c72  .......m9T..!.<r
000006b0: a64f 24cf 74b6 da4d 2b1b 135b d1f1 efc8  .O$.t..M+..[....
000006c0: 08df 9d1a df0c 1c3b c782 31fc 227c 49df  .......;..1."|I.
000006d0: 28bf 8d59 880e 48c1 4d55 3105 3937 4d8e  (..Y..H.MU1.97M.
000006e0: 1cd6 57b6 db83 1f8e c4d7 6403 3fdc 3981  ..W.......d.?.9.
000006f0: aac9 36d3 6386 3892 88f3 02a1 bb31 b6dc  ..6.c.8......1..
00000700: ad5f d4e8 45e3 ee41 8df9 5224 f429 9569  ._..E..A..R$.).i
00000710: 5e13 ae6a 6a77 5c2c 26a7 331f 5644 ca2f  ^..jjw\,&.3.VD./
00000720: 9e6e be07 18be 0c15 685b 235e 11b1 768d  .n......h[#^..v.
00000730: 6bda 62b1 2d68 cee2 21da d88c 4010 e086  k.b.-h..!...@...
00000740: e4c5 2c23 6de8 cc8f 835f e5f1 2660 db32  ..,#m...._..&`.2
00000750: 0757 0707 483e 5b10 8f3e f1f1 4ac3 488a  .W..H>[..>..J.H.
00000760: ecd8 ec55 4c13 5464 32e5 34c3 6cb5 d165  ...UL.Td2.4.l..e
00000770: dcd7 07b6 3e34 b2e9 ff00 f7f7 fd54 5077  ....>4.......TPw
00000780: 820e 3930 6160 a0ed 1329 e741 e44f 55f6  ..90a`...).A.OU.
00000790: b87f 627d 7e54 3d39 bd85 a3dc b7b3 f3c3  ..b}~T=9........
000007a0: 1286 3ed0 693c 087b f8c6 3d97 9f9d f22a  ..>.i<.{..=....*
000007b0: 7710 edd4 a128 a3d8 65f7 0186 a16f 195c  w....(..e....o.\
000007c0: 3e47 8c1d 7f6c 5d50 054a 418a 1f23 cc6a  >G...l]P.JA..#.j
000007d0: c83c 898c 0e5a d551 e407 25e5 bf76 33f0  .<...Z.Q..%..v3.
000007e0: b390 8e98 5365 3a53 a12f 7101 f171 c2be  ....Se:S./q..q..
000007f0: 0858 f733 59ec 113a 75af d73f 18ed 3e59  .X.3Y..:u..?..>Y
00000800: d04f 7948 f5ca da3a d7a8 9fd7 3a9a d123  .OyH...:....:..#
00000810: afc4 d6fa 8da0 e708 aaad 6891 156a 32c4  ..........h..j2.
00000820: 0e77 2b98 22fd 72c7 3fcc 46b0 adde 7136  .w+.".r.?.F...q6
00000830: addb a1a7 a767 1f4f d726 005a 11f6 478c  .....g.O.&.Z..G.
00000840: d38e 2bc6 8e1a b625 d87a e29d 017d 93b7  ..+....%.z...}..
00000850: 31da 09db f14a af12 c2a0 2ea2 67c0 1be9  1....J......g...
```

```
00000860: dcfd 0dd7 31d9 f239 76f6 28bb 734a 2449  ....1..9v.(.sJ$I
00000870: ca07 2d54 d8b1 c513 6369 9512 449d bb05  ..-T....ci..D...
00000880: 428c 6a20 93fb 5190 1f87 c433 a72f ff00  B.j ..Q....3./..
00000890: 61e3 2844 1dd9 a89e b443 46fb 66c7 1b16  a.(D.....CF.f...
000008a0: 19e4 955f cfbd 0356 67b7 82da 239a 2bb7  ..._...Vg...#.+.
000008b0: 8548 2118 f350 5ddc 2017 0f59 8942 23d0  .H!..P]. ..Y.B#.
000008c0: e401 eca5 6ae8 7041 e5a3 d46d cd3f c575  ....j.pA...m.?.u
000008d0: bf7a 2c17 8430 6881 5413 4fcc ba04 2faa  .z,..0h.T.O.../.
000008e0: 5b3e dbad 4f1f 29ba aa2e 2bdf 0b78 b083  [>..O.)...+..x..
000008f0: 08ed 8741 33fc 53ea 9bf3 335f f5f8 dac5  ...A3.S...3_....
00000900: 568b 24c4 bdee 6e55 36a7 89bd 1c22 365a  V.$...nU6....”6Z
00000910: a51c 376b cc72 5b5a 6200 fcda 4223 1e57  ..7k.r[Zb...B#.W
00000920: 44b6 82b1 7c8b ecba 980e 5efa 76d7 eb8f  D...|.....^.v...
00000930: ba02 f00c aa88 7e09 7bfc 983b 2dc5 7ce3  ......~.{..;-.|.
00000940: a1b3 d04c b092 59dc 0914 35af 836d 9d8c  ...L..Y...5..m..
00000950: 25d5 dc45 f09c b1c1 797b fb6d 3d24 36ef  %..E....y{.m=$6.
00000960: 9c5d 841e c76e ea95 3418 c088 a9a3 abcc  .]...n..4.......
00000970: 3231 fc36 5f37 639e 5ac0 4d38 a136 299d  21.6_7c.Z.M8.6).
00000980: 8728 970a d4d6 eb65 915c 141c b03d 2766  .(.....e.\...=’f
00000990: 26c4 afa3 befd 7b1d 992d c260 abe3 c196  &.....{..-.`....
000009a0: d804 4fd6 1545 92e2 d1bd 45e4 efa0 85b5  ..O..E....E.....
000009b0: 1cc7 2594 c02f b1fd 6128 6b42 1f26 75c8  ..%../..a(kB.&u.
000009c0: b775 dd14 aa3a 93db 2605 a970 a3f7 76b7  .u...:..&..p..v.
000009d0: ff00 367c c2cd faf0 4639 edab 3853 3dcc  ..6|....F9..8S=.
000009e0: df85 d7fe 6001 c492 a05c ef9a 259c 9ecb  ....`....\..%...
000009f0: b6b3 c982 6096 c9cf 2786 4b9d 3192 ec72  ....`...’.K.1..r
00000a00: 04db bc5d e6de fc39 4b66 56af d5f7 2fb6  ...]...9KfV.../.
00000a10: 60ce b9ae 154b e797 df73 7709 71b4 69ed  `....K...sw.q.i.
00000a20: e33f 0f35 c637 78ed 811d 877c ec50 5978  .?.5.7x....|.PYx
00000a30: 7231 b1c6 54d8 d503 3eef 7508 8afd 47af  r1..T...>.u...G.
00000a40: 03a0 8415 ec48 fb35 240d 4708 29d7 e38f  .....H.5$.G.)...
00000a50: 29c2 e5fa 0231 52fc c95c f9e9 1c29 3351  )....1R..\...)3Q
00000a60: 341f 5adc cd1d e828 060c 115e 0844 9d41  4.Z....(...^.D.A
00000a70: 6959 ef2b dae0 ecb2 6982 926a eb94 c928  iY.+....i..j...(
00000a80: d902 81c4 8c1d 7f6c 5d50 054a 418a 2367  .......l]P.JA.#g
00000a90: 0c2a dd6c f574 8d79 cb3b 59dc 4fcc ba04  .*.l.t.y.;Y.O...
00000aa0: 2faa 5b3e dbad 4f1f 29ba aa2e 2bdf 0b78  /.[>..O.)...+..x
00000ab0: 55ae 6658 aab4 774e 04af 9be4 21eb c737  U.fX..wN....!..7
00000ac0: deb4 407f 7126 0dc7 15ef a27e d97d 54dc  ..@.q&.....~.}T.
00000ad0: 8da5 7da0 cdff ffff ffe4 0a61 061c 097a  ..}........a...z
00000ae0: a24d fcb2 329c cbc1 3bdb d629 172d 5063  .M..2...;..).-Pc
00000af0: 535c 04a9 a660 ef38 66cf 03cb c1ee d6a6  S\...`.8f.......
00000b00: 8993 3f74 9d86 0798 acac 8f62 9e03 5c4a  ..?t.......b..\J
00000b10: ba76 9e08 bc5c a691 3ce1 3af7 0754 3a43  .v...\..<.:..T:C
00000b20: 40c9 0c16 fac0 f8da 2258 6157 9175 e199  @.......”XaW.u..
00000b30: e956 3ce5 9eff 007c 3926 a8fd 0080 4ea4  .V<....|9&....N.
00000b40: 4d0a 2d06 dc63 61ae 4eb3 770c 0156 a63f  M.-..ca.N.w..V.?
00000b50: 6bc8 ca7e 5bdd 81d4 3ee8 e575 3ee0 6116  k..~[...>..u>.a.
00000b60: 13c8 b097 7a6e 8ffc 502c c4fd e82a 4a63  ....zn..P,...*Jc
00000b70: 2870 b96c b350 a350 c543 4431 0ce9 8fb7  (p.l.P.P.CD1....
00000b80: b4bb 4b0e 4dc9 f274 98e9 ef2a 3abd cc3c  ..K.M..t...*:..<
00000b90: 38a1 65ad 438d 9b00 f1de fc52 cf16 6195  8.e.C......R..a.
```

```
00000ba0: c847 1c9f 4d53 a278 dcdc dcdc dcdc dcdc  .G..MS.x........
00000bb0: dcdc dcdc dcdc dcdc dcdc 4c5c b4dc 2d1f  ..........L\..-.
00000bc0: cd7a 7187 6034 b30b eeef ba3b 02cf afc8  .zq.`4.....;....
00000bd0: 6648 eb19 6c46 3e35 542f 9d08 ff00 a917  fH..lF>5T/......
00000be0: 50c7 75a7 f7cd 7d95 f79f e221 1a0b 3358  P.u...}....!..3X
00000bf0: 7af3 71c6 8ee9 b1ee 917a 3dc0 8b60 dc80  z.q......z=..`..
00000c00: 30c8 5ba2 03c7 cfb6 fc68 fb4a ef6a 97a7  0.[......h.J.j..
00000c10: b5b7 ee67 9e44 55ff 00a6 e811 2a87 de82  ...g.DU.....*...
00000c20: 7386 2bb0 82ac 7ad0 3e4a 6c37 b9af b53a  s.+...z.>Jl7...:
00000c30: 6015 1278 c105 007f 649e e3c7 96c1 6f06  `..x....d.....o.
00000c40: c8cc 8afa 52bf 34ad aae1 e010 7fd2 dec2  ....R.4.........
00000c50: 3345 0161 c7cc 79ab 7b13 250d 6859 9f0a  3E.a..y.{.%.hY..
00000c60: 800b 650e fa9d 1067 52b2 55e7 9772 2221  ..e....gR.U..r"!
00000c70: 5004 a17b eb22 25d1 544e 54d4 2e59 840b  P..{."%.TNT..Y..
00000c80: 12c1 0c80 180d ed6c ebd8 eb4c 0c1c 5eb4  .......l...L..^.
00000c90: f50f c86e 7fb4 7742 90e6 9d31 7628 379c  ...n..wB...1v(7.
00000ca0: f061 4a7a 37e7 3b05 a854 9a05 45be d402  .aJz7.;..T..E...
00000cb0: 337e 5311 e633 ac91 076b 1998 836b 62f4  3~S..3...k...kb.
00000cc0: da28 10bf 8486 39e9 be69 7995 2a39 3add  .(....9..iy.*9:.
00000cd0: 97de 7bf1 4511 0d07 a3df 3500 0004 00a1  ..{.E.....5.....
00000ce0: 4cd6 1c17 610f 4f7c 27d1 1c55 5bb5 265c  L...a.O|'..U[.&\
00000cf0: 8744 ff00 5eff 00ae d6cb 1ca7 f056 f90c  .D..^........V..
00000d00: 2992 53c3 bd0e cc9c b67b 7df8 b951 2f86  ).S......{}..Q/.
00000d10: acab b638 0a48 aa8c a5a6 da06 2738 4ee1  ...8.H......'8N.
00000d20: d079 526f d962 ca57 39c9 57d5 3a66 1b1f  .yRo.b.W9.W.:f..
00000d30: 1e1d 1a1c 1c20 242e 2720 ba5d f610 8c9f  ..... $.' .]....
00000d40: 029f fead cd8d 70e2 4a7a d997 28e3 e996  ......p.Jz..(...
00000d50: 0c0e 3ad2 f09c edbf 1256 2ad8 521a c671  ..:......V*.R..q
00000d60: b564 c907 67be 4610 2842 18df 19b5 77a5  .d..g.F.(B....w.
00000d70: 4bba 8d6e 7903 d332 4086 f7f1 454d 9f25  K..ny..2@...EM.%
00000d80: c7d4 5533 10ff 00f0 561d 8ba1 9339 b34d  ..U3....V....9.M
00000d90: 9f34 983f ed5d 1607 c08d 6b43 0df7 ae6e  .4.?.]....kC...n
00000da0: f457 ab88 5d2c b96f 38e3 25db 719e 1599  .W..],.o8.%.q...
00000db0: 379d 0966 3b93 6df3 83f1 f1a3 a648 09bb  7..f;.m......H..
00000dc0: fba0 81f3 2e2e 52dd 0f29 9003 5564 eec2  ......R..)..Ud..
00000dd0: 0942 d1e2 4e73 7491 4812 4d22 f951 5919  .B..Nst.H.M".QY.
00000de0: be54 0a2c 9426 e34c da7d 36b4 2785 176c  .T.,.&.L.}6.'..l
00000df0: 1b6c e40c eae7 7477 ade1 7e46 c6a2 e6dc  .l....tw..~F....
00000e00: ed46 143d a98e 4e7c 9ec8 619f b001 2d4b  .F.=..N|..a...-K
00000e10: 2729 8e06 c36a 2769 018b 3769 a69c 51a5  ')...j'i..7i..Q.
00000e20: 4bd7 ebe6 74fc b01c 59e0 79fa 7d60 bb95  K...t...Y.y.}`..
00000e30: a235 35a8 b03f 4548 ed24 64d2 9a61 9a79  .55..?EH.$d..a.y
00000e40: 1c45 44df 6fde 1048 767b 03aa f960 f2b7  .ED.o..Hv{...`..
00000e50: 892f 78f4 cbe3 492c d7a8 9fd7 3a9a d123  ./x...I,....:..#
00000e60: afc4 d6fa 8d0b 7b55 2d47 a4e6 e3ff 00ed  ......{U-G......
00000e70: 7882 e55c f04b 5f44 3cb6 9ccb cb20 21bf  x..\.K_D<.... !.
00000e80: e47e 0e4b 9515 1348 10cb a688 51ea d4bd  .~.K...H....Q...
00000e90: 1def cff0 cdb9 59bc 30b4 e477 9936 1760  ......Y.0..w.6.`
00000ea0: 480d 4147 98d6 37e7 7bca 29b0 b807 3627  H.AG..7.{.)...6'
00000eb0: d712 e58e 9de8 5128 f074 ca51 5087 d8b9  ......Q(.t.QP...
00000ec0: 7d5f d402 9ded e03e 1c7e 13e7 c5af 4295  }_.....>.~....B.
00000ed0: a886 ccb3 d8a0 0fab 68cf adc8 cf55 cb    ........h....U.
```

Listing A.1: Input File for `tj3Transform()` Out-of-Bounds Write Crash

A.1.0.2 Fuzzing of tj3Transform() - Memory Corruption

```
#include <turbojpeg.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <limits.h>
#include <unistd.h>
#include <sys/types.h>

/* this lets the source compile without afl-clang-fast/lto */
#ifndef __AFL_FUZZ_TESTCASE_LEN

ssize_t fuzz_len;
unsigned char fuzz_buf[1024000];

#define __AFL_FUZZ_TESTCASE_LEN fuzz_len
#define __AFL_FUZZ_TESTCASE_BUF fuzz_buf
#define __AFL_FUZZ_INIT() void sync(void);
#define __AFL_LOOP(x) \
    ((fuzz_len = read(0, fuzz_buf, sizeof(fuzz_buf))) > 0 ? 1 : 0)
#define __AFL_INIT() sync()

#endif

typedef struct TestTJParam {
    int index;
    char *name;
} TestTJParam;

static TestTJParam testParams[] = {
    {TJXOPT_PERFECT, "TJXOPT_PERFECT"},
    {TJXOPT_TRIM, "TJXOPT_TRIM"},
    {TJXOPT_CROP, "TJXOPT_CROP"},
    {TJXOPT_GRAY, "TJXOPT_GRAY"},
    {TJXOPT_NOOUTPUT, "TJXOPT_NOOUTPUT"},
    {TJXOPT_PROGRESSIVE, "TJXOPT_PROGRESSIVE"},
    {TJXOPT_COPYNONE, "TJXOPT_COPYNONE"},
    {TJXOPT_ARITHMETIC, "TJXOPT_ARITHMETIC"},
    {TJXOPT_OPTIMIZE, "TJXOPT_OPTIMIZE"}
};
static unsigned int testParamLen = sizeof(testParams)/sizeof(TestTJParam);

__AFL_FUZZ_INIT();

```

```
44      void do_test(unsigned char *data, size_t size)
45      {
46          tjhandle handle = NULL;
47          unsigned char *dstBuf[1] = {NULL};
48          size_t dstSizes[1] = {0};
49          size_t maxBufSize = 0;
50          int width = 0, height = 0, jpegSubsamp, i, t;
51          tjtransform transform = {0};
52
53          unsigned short transform_options = (data[0] << 8) | (data[1]);
54
55
56          /*
57          IMPORTANT NOTE: first 3 bytes of the test file contain transform options and operation!
58          */
59          int op = data[2] % TJ_NUMXOP;
60
61          if ((handle = tj3Init(TJINIT_TRANSFORM)) == NULL) {
62              goto bailout;
63          }
64
65          data += 3;
66          size -= 3;
67
68          if (tj3DecompressHeader(handle, data, size) < 0) {
69              goto bailout;
70          }
71
72          width = tj3Get(handle, TJPARAM_JPEGWIDTH);
73          height = tj3Get(handle, TJPARAM_JPEGHEIGHT);
74          jpegSubsamp = tj3Get(handle, TJPARAM_SUBSAMP);
75
76          /* Ignore 0-pixel images and images larger than 1 Megapixel.  Casting width
77             to (uint64_t) prevents integer overflow if width * height > INT_MAX. */
78          if (width < 1 || height < 1 || (uint64_t)width * height > 1048576) {
79              goto bailout;
80          }
81
82          tj3Set(handle, TJPARAM_SCANLIMIT, 500);
83
84          if ((transform_options & TJXOPT_CROP) > 0)
85          {
86              transform.r.w = (width + 1) / 2;
87              transform.r.h = (height + 1) / 2;
88
89              width = transform.r.w;
90              height = transform.r.h;
91          }
92
93          transform.options = transform_options;
94          transform.op = op;
95
```

```
96          maxBufSize = tj3JPEGBufSize(width, height, jpegSubsamp);
97          if (!maxBufSize)
98              goto bailout;
99
100         if ((op & TJXOP_ROT90) > 0 || ((op & TJXOP_ROT270) > 0))
101         {
102             dstBuf[0] =
103                 (unsigned char *)malloc(tj3JPEGBufSize(height, width, jpegSubsamp));
104             if (!dstBuf[0]) {
105                 goto bailout;
106             }
107         }
108         else
109         {
110             dstBuf[0] =
111                 (unsigned char *)malloc(tj3JPEGBufSize(width, height, jpegSubsamp));
112             if (!dstBuf[0]) {
113                 goto bailout;
114             }
115         }
116
117         if (tj3Set(handle, TJPARAM_NOREALLOC, 1) < 0) {
118             goto bailout;
119         }
120
121         if (tj3Transform(handle, data, size, 1, dstBuf, dstSizes,
122                          &transform) == 0)
123         {
124             /* Touch all of the output pixels in order to catch uninitialized reads
125                when using MemorySanitizer. */
126
127             int sum = 0;
128
129             for (i = 0; i < dstSizes[0]; i++)
130                 sum += dstBuf[0][i];
131
132             /* Prevent the code above from being optimized out.  This test should
133                 never be true, but the compiler doesn't know that. */
134             if (sum > 255 * maxBufSize) {
135                 goto bailout;
136             }
137         }
138
139     bailout:
140         free(dstBuf[0]);
141         tj3Destroy(handle);
142     }
143
144     void make_testcase(const char *filepath)
145     {
146         int options = TJXOPT_PROGRESSIVE | TJXOPT_COPYNONE;
147
```

```
148         unsigned char data[200*1024] = {0};
149         int nread = 0;
150 
151         data[0] = (char)(options >> 8);
152         data[1] = (char)(options);
153         data[2] = TJXOP_NONE;
154 
155         FILE *fp = fopen("testcase", "wb");
156         if (fp)
157         {
158             fwrite(data, 1, 3, fp);
159 
160             FILE *fJpeg = fopen(filepath, "rb");
161             if (fJpeg)
162             {
163                 nread = fread(data + 3, 1, sizeof(data) - 3, fJpeg);
164                 fclose(fJpeg);
165 
166                 fwrite(data + 3, 1, nread, fp);
167 
168                 nread += 3;
169             } else {
170                 fclose(fp);
171                 return;
172             }
173 
174             fclose(fp);
175 
176             do_test(data, nread);
177         }
178     }
179 
180     //#define MAKE_TESTCASE 1
181     //#define LOAD_TESTFILE 1
182     //#define TEST_WITH_PARAM 1
183 
184     int main(int argc, char **argv)
185     {
186     #ifdef MAKE_TESTCASE
187          make_testcase(argv[1]);
188         return 0;
189     #endif
190 
191     #ifdef LOAD_TESTFILE
192         FILE *fp = fopen(argv[1], "rb");
193         if (fp)
194         {
195             unsigned char imgdata[200 * 1024];
196             int nread = fread(imgdata, 1, sizeof(imgdata), fp);
197             fclose(fp);
198 
199             if (nread < 4)
```

```
200             {
201                 printf("not enough data for test!");
202             }
203 
204             unsigned short transform_options = (imgdata[0] << 8) | (imgdata[1]);
205 
206             int op = imgdata[2] % TJ_NUMXOP;
207 
208             printf("Operation: %02x\n", op);
209             printf("TransformOptions: %02x\n", transform_options);
210 
211             for (int i=0; i<testParamLen; i++)
212             {
213                 if (transform_options & testParams[i].index) {
214                     printf("Setting %s to 1\n", testParams[i].name);
215                 }
216             }
217 
218             do_test(imgdata, nread);
219         }
220         else
221         {
222             printf("Error opening file\n");
223         }
224 
225         return 0;
226     #elif TEST_WITH_PARAM
227       FILE *fp = fopen(argv[1], "rb");
228       if (fp)
229       {
230         unsigned char data[200*1024] = {0};
231         int nread = fread(data + 3, 1, sizeof(data) - 3, fp);
232         fclose(fp);
233 
234         // test param: no transform options, OP: ROT180
235         data[0] = 0;
236         data[1] = 0;
237         data[2] = TJXOP_ROT180;
238 
239         do_test(data, nread + 3);
240       }
241       return 0;
242     #endif
243         ssize_t len;        /* how much input did we read? */
244         unsigned char *buf; /* test case buffer pointer    */
245 
246         __AFL_INIT();
247         buf = __AFL_FUZZ_TESTCASE_BUF; // this must be assigned before __AFL_LOOP!
248 
249         while (__AFL_LOOP(UINT_MAX))
250         {
251             len = __AFL_FUZZ_TESTCASE_LEN;
```

```
252             if (len < 4)
253                 continue;
254
255             unsigned char *buf_new = buf;
256
257             do_test(buf_new, len);
258         }
259
260         return 0;
261     }
```

Listing A.2: tj3Transform() Memory Safety

A crashing test input (first 3 bytes contain transform_options and operation, in this case options=0 and op=6 (*TXJOP_ROT180*)) is shown below. The file can be loaded by setting `#define LOAD_TESTFILE 1` in the test harness before compilation.

```
00000000: 0000 06ff d8ff e000 104a 4632 4432 3232  .........JF2D222
00000010: 3232 0100 0100 00ff db00 4300 0806 0607  22........C.....
00000020: 0000 0400 0707 0909 080a 0c14 0d0c 0b0b  ................
00000030: 0c19 1213 0f5f 1d1a 1f1e 1d1a 1c1c 2024  ....._........ $
00000040: 2e27 2022 1129 3332 3233 3132 3280 3232  .' ".)3223122.22
00000050: 321f 2739 3d38 1a3c 05fe ff05 ffdb 0043  2.'9=8.<.......C
00000060: 0105 0909 0c0b 0c18 0d0d 1832 211c 2132  ...........2!.!2
00000070: 326a 2532 3232 320c 4032 7f1c 2837 292c  2j%2222.@2..(7),
00000080: 3031 3434 3432 3232 3232 3232 0303 0303  014442222222....
00000090: 03f8 0303 0303 3232 3232 3232 3232 3232  ......2222222222
000000a0: 32ff c900 110c 0001 000d 0301 1200 0211  2...............
000000b0: 0103 1101 ffcc 000a 0010 1005 1410 1105  ................
000000c0: ffda 000c 0301 0002 1103 eb8a 13ff d8ff  ................
000000d0: e0ff 004a 4632 3232 3232 3232 0100 0100  ...JF2222222....
000000e0: 00ff db00 0400 0707 0909 080a 0c14 0d0c  ................
000000f0: 0b0b 0c19 1213 0f5f 1d1a 1f1e 1d1a 1c1c  ......._........
00000100: 2024 2e27 2022 2ceb 8a5a ffd8 ffe0 0010   $.' ",..Z......
00000110: 4a46 3244 3232 3232 3200 6000 ffd8 ffe0  JF2D22222.`.....
00000120: 0010 4a46 5246 0001 0100 0001 0001 0000  ..JFRF..........
00000130: ffdb 0043 0008 0606 0706 0508 0707 0709  ...C............
00000140: 0908 0a0c 140d 0c0b 0b0c 1912 130f 141d  ................
00000150: 1a1f 1e1d 1a1c 1c20 242e 2720 222c 231c  ....... $.' ",#.
00000160: 1c28 3729 2c30 3134 3234 1f27 393d 3832  .(7),01424.'9=82
00000170: 3c2e 3334 32ff db00 4301 0909 090c 0b0c  <.342...C.......
00000180: 180d 0d18 3221 1c21 3232 3232 3232 3232  ....2!.!22222222
00000190: 3232 3232 3232 3232 3232 3232 3232 3232  2222222222222222
000001a0: 3232 3232 3232 3232 3232 3232 3232 3232  2222222222222222
000001b0: 3232 3232 3232 3232 3232 ffc1 0011 0800  2222222222......
000001c0: 9500 e303 0122 0002 1101 0311 01ff cc00  ....."..........
000001d0: 0a00 1010 0501 1011 05ff da00 0c03 0100  ................
000001e0: 0211 0311 003f 00ff 00ba b93d 0cdd 51ad  .....?.....=..Q.
000001f0: 025d d3b0 cfb3 d9b6 127b 3681 b935 535c  .].......{6..5S\
00000200: d4ba cc52 34b4 1c66 8d4c e2c7 b8b6 81ed  ...R4..f.L......
```

```
00000210: 46f2 58ae d903 ab77 ccca d1f4 4b45 a9ee  F.X....w....KE..
00000220: 6626 06f9 d0e3 6ad8 2d06 e2a8 c203 18ec  f&....j.-.......
00000230: c08d 3624 1b5f ccea b0b9 95c3 a8b0 5239  ..6$._........R9
00000240: 481f cc9f e78c 3ead 7f5b 6e3d 89f8 7468  H.....>..[n=..th
00000250: fc2d 8013 a5e7 4872 2644 b83d 6906 c22c  .-....Hr&D.=i..,
00000260: 7bb9 6d5b d4f6 1401 33df 0fe5 663a bc0c  {.m[....3...f:..
00000270: ab8f 4cf5 917c 237f f2c0 b4af 0f8a 16fe  ..L..|#.........
00000280: d9b3 18a2 4f94 04bf 8af0 c18c a601 a5d1  ....O...........
00000290: a132 af4f 5fd5 f394 53ae 1626 afca 0250  .2.O_...S..&...P
000002a0: 426f 1b57 8f81 bb4d 1d6e 1def e127 f8f1  Bo.W...M.n...'..
000002b0: 4c85 77f5 0a92 3a3b a08f f970 0edb a031  L.w...:;...p...1
000002c0: f985 f2e7 fc2e b357 98ef 2ce7 7438 19a1  .......W..,.t8..
000002d0: a883 e34f 4ef9 58bb e60e af45 143f e879  ...ON.X....E.?.y
000002e0: 93cf ce39 425b b6b9 7e6a 5f22 4130 e975  ...9B[..~j_"A0.u
000002f0: 4832 6bb9 d506 1ec3 1664 1c80 c13b 4f89  H2k......d...;O.
00000300: 870c a7fb a4b1 f992 a01b 92cc df09 345b  ..............4[
00000310: 4a5a f200 674c 1a6f 1647 cf9b dd53 6f94  JZ..gL.o.G...So.
00000320: 2898 7063 5f49 30ed 1f88 52f9 d01c e6b3  (.pc_I0...R.....
00000330: d9b6 1233 49c0 cb3e f0a0 d264 1b2d 5132  ...3I..>...d.-Q2
00000340: 8080 9586 fc27 15ba 91eb 8093 8d3c f30e  .....'.......<..
00000350: 1efc 82ed 886c 5361 efee b3c3 7025 4ec6  .....lSa....p%N.
00000360: 6233 33cf 517a 304a 2c49 9d9c 8c58 de49  b33.Qz0J,I...X.I
00000370: fef5 cc21 861d b788 5931 e4d3 c556 f728  ...!....Y1...V.(
00000380: 8e37 68e2 c10d b923 bf6d 1fee babb 52f1  .7h....#.m....R.
00000390: 9a4a 54d1 673c 371b 81d8 f0cb 1d4d c182  .JT.g<7......M..
000003a0: 915e c456 4270 46e4 0d9e 489e 876e 11f2  .^.VBpF...H..n..
000003b0: e0cb 210e 7149 43ad e801 ac35 08d3 b2e2  ..!.qIC....5....
000003c0: 40fb 66b8 e72a ce8f 4fef d9fa b565 151e  @.f..*..O....e..
000003d0: 622e 905a 1c6c ea45 c60a 583f b998 d344  b..Z.l.E..X?...D
000003e0: 56a6 10a3 7d10 56c5 bc10 29e4 dadb 5b53  V...}.V...)...[S
000003f0: 2d43 f0e5 9674 97fd f54e 77fd 1448 aec5  -C...t...Nw..H..
00000400: f29f 2632 0eed 26ed 1671 c883 e9ce 7a39  ..&2..&..q....z9
00000410: 51a7 1b5c 1dc3 fe10 7ecd d312 b270 6177  Q..\....~....paw
00000420: 7f6b d146 7a1d 4ab9 4e05 c457 1029 6d47  .k.Fz.J.N..W.)mG
00000430: 5dfd ba90 a119 43eb fdf3 ba1f 95be c0fa  ].....C.........
00000440: c321 38fc c380 222a 6314 604d 3128 7d48  .!8..."*c.`M1(}H
00000450: ccdc a7ca eb6a 45e1 c08b 3c95 80f5 b9e8  .....jE...<.....
00000460: 9c23 8d29 6407 8820 fcca f4af 344b d76e  .#.)d.. ....4K.n
00000470: 91b5 04a9 a660 ef38 66cf 03cb c1ee d6a6  .....`.8f.......
00000480: 8993 3f74 c8a8 f62b 5177 71c2 b3a8 2ac2  ..?t...+Qwq...*.
00000490: 0e50 655c c3c7 fe75 0168 a175 68f6 d127  .Pe\...u.h.uh..'
000004a0: c832 eff6 c829 622a 6e40 f8e9 d600 b057  .2...)b*n@.....W
000004b0: 95ae 33ca 671c ae92 1adb 393b 1566 b4cf  ..3.g.....9;.f..
000004c0: 63dd 4712 0ab1 6e69 8d68 5374 b6a0 a051  c.G...ni.hSt...Q
000004d0: 4ff3 76f9 8349 1763 bc01 3591 8caa 4f09  O.v..I.c..5...O.
000004e0: 9fec 3886 e5db a09e 39dc 8051 cac9 8563  ..8.....9..Q...c
000004f0: 511d 5ede 0b58 1c4c cb07 911c 0cd8 57b8  Q.^..X.L......W.
00000500: 9464 cf9f e565 412d 8fa0 7204 0eec ca5b  .d...eA-..r....[
00000510: a00b 1ad6 e854 0551 c3db 1ce6 a6a9 2710  .....T.Q......'.
00000520: 9a1e 83dd e263 4d9d 1e16 1b53 0e80 c6d3  .....cM....S....
00000530: 88b9 8639 60e9 3c6c 378d d73b f666 8e53  ...9`.<l7..;.f.S
00000540: 83cc fe9e 297a b274 687f a8cc 85e8 de94  ....)z.th.......
```

```
00000550: c83d fbb0 a6c3 4065 18fb 10e4 38c7 7af0  .=....@e....8.z.
00000560: 723e cfb8 1af2 586d 6a7f a5ee fb25 8dbf  r>....Xmj....%..
00000570: 92ab 2123 7fba ce20 c1d6 42f7 aae4 642b  ..!#... ..B...d+
00000580: 0f31 8ca0 0ff9 a964 eae7 68be 35e8 2eac  .1.....d..h.5...
00000590: 64cb 9415 07bd 5a97 74a8 234e 8317 d49f  d.....Z.t.#N....
000005a0: 7221 06b2 173f 92d4 3399 d566 079b c945  r!...?..3..f...E
000005b0: d82a 7c8a c48c 2c24 ccf8 efc8 fc89 0436  .*|...,$.......6
000005c0: 4380 698a 9c37 0783 a9d7 62c1 b2af 1495  C.i..7....b.....
000005d0: 2659 8e84 6529 d184 325e e3ec 12eb 9403  &Y..e)..2^......
000005e0: 661a 0d26 6816 23bf cb42 837f 1b39 08b6  f..&h.#..B...9..
000005f0: 83df 3760 9229 284d bbb2 0ba1 1df9 d369  ..7`.)(M.......i
00000600: 033e ef75 088a fd47 af03 a084 15ec 48fb  .>.u...G......H.
00000610: 3524 0d47 0829 d7e3 8f29 c2e5 fa02 1709  5$.G.)...)......
00000620: ea0b 6783 1c38 3f1b 3a23 9afb b261 758b  ..g..8?.:#...au.
00000630: 1e55 fbeb 9b1f a237 6266 9888 8196 ccc8  .U.....7bf......
00000640: 0efd a7cd 0b3b 1f74 f7c1 a24e 2030 2970  .....;.t...N 0)p
00000650: 594a 194d 4c65 93e7 f5b0 9ec1 cc41 9bf6  YJ.MLe.......A..
00000660: f178 c925 178a 8b7b 38a9 6811 17e8 e575  .x.%...{8.h....u
00000670: 3ee0 6116 13c8 b097 7a6e 8ffc 502c c4fd  >.a.....zn..P,..
00000680: e82a 4a63 dcb1 3d95 1289 e820 0000 00f2  .*Jc..=.... ....
00000690: adab a03c 82ba c441 3650 5042 b138 dd58  ...<...A6PPB.8.X
000006a0: bcfd 318f c8c6 00a2 e21f 6d39 5484 9821  ..1.......m9T..!
000006b0: 963c 72a6 4f24 cf74 b6da 4d2b 1b13 5bd1  .<r.O$.t..M+..[.
000006c0: f1ef c808 df9d 1adf 0c1c 3bc7 8231 fc22  ..........;..1."
000006d0: 7c49 df28 bf8d 5988 0e48 c14d 5531 0539  |I.(..Y..H.MU1.9
000006e0: 374d 8e1c d657 b6db 831f 8ec4 d764 033f  7M...W.......d.?
000006f0: dc39 81aa c936 d363 8638 9288 f302 a1bb  .9...6.c.8......
00000700: 31b6 dcad 5fd4 e845 e3ee 418d f952 24f4  1..._..E..A..R$.
00000710: 2995 695e 13ae 6a6a 775c 2c26 a733 1f56  ).i^..jjw\,&.3.V
00000720: 44ca 2f9e 6ebe 0718 be0c 1568 5b23 5e11  D./.n......h[#^.
00000730: b176 8d6b da62 b12d 68ce e221 dad8 8c40  .v.k.b.-h..!...@
00000740: 10e0 86e4 c52c 236d e8cc 8f83 5fe5 f126  .....,#m...._..&
00000750: 60db 3207 5707 0748 3e5b 108f 3ef1 f14a  `.2.W..H>[..>..J
00000760: c348 8aec d8ec 554c 1354 6432 e534 c36c  .H....UL.Td2.4.l
00000770: b5d1 65dc d707 b63e 34b2 e9ff 00f7 f7fd  ..e....>4.......
00000780: 5450 7782 0e39 3061 60a0 ed13 29e7 41e4  TPw..90a`...).A.
00000790: 4f55 f6b8 7f62 7d7e 543d 39bd 85a3 dcb7  OU...b}~T=9.....
000007a0: b3f3 c312 863e d069 3c08 7bf8 c63d 979f  .....>.i<.{..=..
000007b0: 9df2 2a77 10ed d4a1 28a3 d865 f701 86a1  ..*w....(..e....
000007c0: 6f19 5c3e 478c 1d7f 6c5d 5005 4a41 8a1f  o.\>G...l]P.JA..
000007d0: 23cc 6ac8 3c89 8c0e 5ad5 51e4 0725 e5bf  #.j.<...Z.Q..%..
000007e0: 7633 f0b3 908e 9853 653a 53a1 2f71 01f1  v3.....Se:S./q..
000007f0: 71c2 be08 58f7 3359 ec11 3a75 afd7 3f18  q...X.3Y..:u..?.
00000800: ed3e 59d0 4f79 48f5 cada 3ad7 a89f d73a  .>Y.OyH...:....:
00000810: 9ad1 23af c4d6 fa8d a0e7 08aa ad68 9115  ..#..........h..
00000820: 6a32 c40e 772b 9822 fd72 c73f cc46 b0ad  j2..w+."r.?.F..
00000830: de71 36ad dba1 a7a7 671f 4fd7 2600 5a11  .q6.....g.O.&.Z.
00000840: f647 8cd3 8e2b c68e 1ab6 25d8 7ae2 9d01  .G...+....%.z...
00000850: 7d93 b731 da09 dbf1 4aaf 12c2 a02e a267  }..1....J......g
00000860: c01b e9dc fd0d d731 d9f2 3976 f628 bb73  .......1..9v.(.s
00000870: 4a24 49ca 072d 54d8 b1c5 1363 6995 1244  J$I..-T....ci..D
00000880: 9dbb 0542 8c6a 2093 fb51 901f 87c4 33a7  ...B.j ..Q....3.
```

```
00000890: 2fff 0061 e328 441d d9a8 9eb4 4346 fb66  /..a.(D.....CF.f
000008a0: c71b 1619 e495 5fcf bd03 5667 b782 da23  ......_...Vg...#
000008b0: 9a2b b785 4821 18f3 505d dc20 170f 5989  .+..H!..P]. ..Y.
000008c0: 4223 d0e4 01ec a56a e870 41e5 a3d4 6dcd  B#.....j.pA...m.
000008d0: 3fc5 75bf 7a2c 1784 3068 8154 134f ccba  ?.u.z,..0h.T.O..
000008e0: 042f aa5b 3edb ad4f 1f29 baaa 2e2b df0b  ./.[>..O.)...+..
000008f0: 78b0 8308 ed87 4133 fc53 ea9b f333 5ff5  x.....A3.S...3_.
00000900: f8da c556 8b24 c4bd ee6e 5536 a789 bd1c  ...V.$...nU6....
00000910: 2236 5aa5 1c37 6bcc 725b 5a62 00fc da42  "6Z..7k.r[Zb...B
00000920: 231e 5744 b682 b17c 8bec ba98 0e5e fa76  #.WD...|.....^.v
00000930: d7eb 8fba 02f0 0caa 887e 097b fc98 3b2d  .........~.{..;-
00000940: c57c e3a1 b3d0 4cb0 9259 dc09 1435 af83  .|....L..Y...5..
00000950: 6d9d 8c25 d5dc 45f0 9cb1 c179 7bfb 6d3d  m..%..E....y{.m=
00000960: 2436 ef9c 5d84 1ec7 6eea 9534 18c0 88a9  $6..]...n..4....
00000970: a3ab cc32 31fc 365f 3763 9e5a c04d 38a1  ...21.6_7c.Z.M8.
00000980: 3629 9d87 2897 0ad4 d6eb 6591 5c14 1cb0  6)..(.....e.\...
00000990: 3d27 6626 c4af a3be fd7b 1d99 2dc2 60ab  ='f&.....{..-.`.
000009a0: e3c1 96d8 044f d615 4592 e2d1 bd45 e4ef  .....O..E....E..
000009b0: a085 b51c c725 94c0 2fb1 fd61 286b 421f  .....%../..a(kB.
000009c0: 2675 c8b7 75dd 14aa 3a93 db26 05a9 70a3  &u..u...:..&..p.
000009d0: f776 b7ff 0036 7cc2 cdfa f046 39ed ab38  .v...6|....F9..8
000009e0: 533d ccdf 85d7 fe60 01c4 92a0 5cef 9a25  S=.....`....\..%
000009f0: 9c9e cbb6 b3c9 8260 96c9 cf27 864b 9d31  .......`...'.K.1
00000a00: 92ec 7204 dbbc 5de6 defc 394b 6656 afd5  ..r...]...9KfV..
00000a10: f72f b660 ceb9 ae15 4be7 97df 7377 0971  ./.`....K...sw.q
00000a20: b469 ede3 3f0f 35c6 3778 ed81 1d87 7cec  .i..?.5.7x....|.
00000a30: 5059 7872 31b1 c654 d8d5 033e ef75 088a  PYxr1..T...>.u..
00000a40: fd47 af03 a084 15ec 48fb 3524 0d47 0829  .G......H.5$.G.)
00000a50: d7e3 8f29 c2e5 fa02 3152 fcc9 5cf9 e91c  ...)....1R..\...
00000a60: 2933 5134 1f5a dccd 1de8 2806 0c11 5e08  )3Q4.Z....(...^.
00000a70: 449d 4169 59ef 2bda e0ec b269 8292 6aeb  D.AiY.+....i..j.
00000a80: 94c9 28d9 0281 c48c 1d7f 6c5d 5005 4a41  ..(.......l]P.JA
00000a90: 8a23 670c 2add 6cf5 748d 79cb 3b59 dc4f  .#g.*.l.t.y.;Y.O
00000aa0: ccba 042f aa5b 3edb ad4f 1f29 baaa 2e2b  .../.[>..O.)...+
00000ab0: df0b 7855 ae66 58aa b477 4e04 af9b e421  ..xU.fX..wN....!
00000ac0: ebc7 37de b440 7f71 260d c715 efa2 7ed9  ..7..@.q&.....~.
00000ad0: 7d54 dc8d a57d a0cd ffff ffff e40a 6106  }T...}........a.
00000ae0: 1c09 7aa2 4dfc b232 9ccb c13b dbd6 2917  ..z.M..2...;..).
00000af0: 2d50 6353 5c04 a9a6 60ef 3866 cf03 cbc1  -PcS\...`.8f....
00000b00: eed6 a689 933f 749d 8607 98ac ac8f 629e  .....?t.......b.
00000b10: 035c 4aba 769e 08bc 5ca6 913c e13a f707  .\J.v...\..<.:..
00000b20: 543a 4340 c90c 16fa c0f8 da22 5861 5791  T:C@........"XaW.
00000b30: 75e1 99e9 563c e59e ff00 7c39 26a8 fd00  u...V<....|9&...
00000b40: 804e a44d 0a2d 06dc 6361 ae4e b377 0c01  .N.M.-..ca.N.w..
00000b50: 56a6 3f6b c8ca 7e5b dd81 d43e e8e5 753e  V.?k..~[...>..u>
00000b60: e061 1613 c8b0 977a 6e8f fc50 2cc4 fde8  .a.....zn..P,...
00000b70: 2a4a 6328 70b9 6cb3 50a3 50c5 4344 310c  *Jc(p.l.P.P.CD1.
00000b80: e98f b7b4 bb4b 0e4d c9f2 7498 e9ef 2a3a  .....K.M..t...*:
00000b90: bdcc 3c38 a165 ad43 8d9b 00f1 defc 52cf  ..<8.e.C......R.
00000ba0: 1661 95c8 471c 9f4d 53a2 78dc dcdc dcdc  .a..G..MS.x.....
00000bb0: dcdc dcdc dcdc dcdc dcdc dcdc dc4c 5cb4  .............L\.
00000bc0: dc2d 1fcd 7a71 8760 34b3 0bee efba 3b02  .-..zq.`4.....;.
```

```
00000bd0: cfaf c866 48eb 196c 463e 3554 2f9d 08ff  ...fH..lF>5T/...
00000be0: 00a9 1750 c775 a7f7 cd7d 95f7 9fe2 211a  ...P.u...}....!.
00000bf0: 0b33 587a f371 c68e e9b1 ee91 7a3d c08b  .3Xz.q......z=..
00000c00: 60dc 8030 c85b a203 c7cf b6fc 68fb 4aef  `..0.[......h.J.
00000c10: 6a97 a7b5 b7ee 679e 4455 ff00 a6e8 112a  j.....g.DU.....*
00000c20: 87de 8273 862b b082 ac7a d03e 4a6c 37b9  ...s.+...z.>Jl7.
00000c30: afb5 3a60 1512 78c1 0500 7f64 9ee3 c796  ..:`..x....d....
00000c40: c16f 06c8 cc8a fa52 bf34 adaa e1e0 107f  .o.....R.4......
00000c50: d2de c233 4501 61c7 cc79 ab7b 1325 0d68  ...3E.a..y.{.%.h
00000c60: 599f 0a80 0b65 0efa 9d10 6752 b255 e797  Y....e....gR.U..
00000c70: 7222 2150 04a1 7beb 2225 d154 4e54 d42e  r"!P..{."%.TNT..
00000c80: 5984 0b12 c10c 8018 0ded 6ceb d8eb 4c0c  Y.........l...L.
00000c90: 1c5e b4f5 0fc8 6e7f b477 4290 e69d 3176  .^....n..wB...1v
00000ca0: 2837 9cf0 614a 7a37 e73b 05a8 549a 0545  (7..aJz7.;..T..E
00000cb0: bed4 0233 7e53 11e6 33ac 9107 6b19 9883  ...3~S..3...k...
00000cc0: 6b62 f4da 2810 bf84 8639 e9be 6979 952a  kb..(....9..iy.*
00000cd0: 393a dd97 de7b f145 110d 07a3 df35 0000  9:...{.E.....5..
00000ce0: 0400 a14c d61c 1761 0f4f 7c27 d11c 555b  ...L...a.O|'..U[
00000cf0: b526 5c87 44ff 005e ff00 aed6 cb1c a7f0  .&\.D..^........
00000d00: 56f9 0c29 9253 c3bd 0ecc 9cb6 7b7d f8b9  V..).S......{}..
00000d10: 512f 86ac abb6 380a 48aa 8ca5 a6da 0627  Q/....8.H......'
00000d20: 384e e1d0 7952 6fd9 62ca 5739 c957 d53a  8N..yRo.b.W9.W.:
00000d30: 661b 1f1e 1d1a 1c1c 2024 2e27 20ba 5df6  f....... $.' .].
00000d40: 108c 9f02 9ffe adcd 8d70 e24a 7ad9 9728  .........p.Jz..(
00000d50: e3e9 960c 0e3a d2f0 9ced bf12 562a d852  .....:......V*.R
00000d60: 1ac6 71b5 64c9 0767 be46 1028 4218 df19  ..q.d..g.F.(B...
00000d70: b577 a54b ba8d 6e79 03d3 3240 86f7 f145  .w.K..ny..2@...E
00000d80: 4d9f 25c7 d455 3310 ff00 f056 1d8b a193  M.%..U3....V....
00000d90: 39b3 4d9f 3498 3fed 5d16 07c0 8d6b 430d  9.M.4.?.]....kC.
00000da0: f7ae 6ef4 57ab 885d 2cb9 6f38 e325 db71  ..n.W..],.o8.%.q
00000db0: 9e15 9937 9d09 663b 936d f383 f1f1 a3a6  ...7..f;.m......
00000dc0: 4809 bbfb a081 f32e 2e52 dd0f 2990 0355  H........R..)..U
00000dd0: 64ee c209 42d1 e24e 7374 9148 124d 22f9  d...B..Nst.H.M".
00000de0: 5159 19be 540a 2c94 26e3 4cda 7d36 b427  QY..T.,.&.L.}6.'
00000df0: 8517 6c1b 6ce4 0cea e774 77ad e17e 46c6  ..l.l....tw..~F.
00000e00: a2e6 dced 4614 3da9 8e4e 7c9e c861 9fb0  ....F.=..N|..a..
00000e10: 012d 4b27 298e 06c3 6a27 6901 8b37 69a6  .-K')...j'i..7i.
00000e20: 9c51 a54b d7eb e674 fcb0 1c59 e079 fa7d  .Q.K...t...Y.y.}
00000e30: 60bb 95a2 3535 a8b0 3f45 48ed 2464 d29a  `...55..?EH.$d..
00000e40: 619a 791c 4544 df6f de10 4876 7b03 aaf9  a.y.ED.o..Hv{...
00000e50: 60f2 b789 2f78 f4cb e349 2cd7 a89f d73a  `.../x...I,....:
00000e60: 9ad1 23af c4d6 fa8d 0b7b 552d 47a4 e6e3  ..#......{U-G...
00000e70: ff00 ed78 82e5 5cf0 4b5f 443c b69c cbcb  ...x..\.K_D<....
00000e80: 2021 bfe4 7e0e 4b95 1513 4810 cba6 8851   !..~.K...H....Q
00000e90: ead4 bd1d efcf f0cd b959 bc30 b4e4 7799  .........Y.0..w.
00000ea0: 3617 6048 0d41 4798 d637 e77b ca29 b0b8  6.`H.AG..7.{.)..
00000eb0: 0736 27d7 12e5 8e9d e851 28f0 74ca 5150  .6'......Q(.t.QP
00000ec0: 87d8 b97d 5fd4 029d ede0 3e1c 7e13 e7c5  ...}_.....>.~...
00000ed0: af42 95a8 86cc b3d8 a00f ab68 cfad c8cf  .B.........h....
00000ee0: 55cb                                     U.
```

Listing A.3: tj3Transform() Crash Input

A.1.0.3 Fuzzing of tj3Decompress()

```
#include <turbojpeg.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <limits.h>
#include <unistd.h>
#include <sys/types.h>

/* this lets the source compile without afl-clang-fast/lto */
#ifndef __AFL_FUZZ_TESTCASE_LEN

ssize_t fuzz_len;
unsigned char fuzz_buf[1024000];

#define __AFL_FUZZ_TESTCASE_LEN fuzz_len
#define __AFL_FUZZ_TESTCASE_BUF fuzz_buf
#define __AFL_FUZZ_INIT() void sync(void);
#define __AFL_LOOP(x) \
    ((fuzz_len = read(0, fuzz_buf, sizeof(fuzz_buf))) > 0 ? 1 : 0)
#define __AFL_INIT() sync()

#endif

__AFL_FUZZ_INIT();

#define MINIMAL_TESTCASE_LEN 4

typedef struct TestTJParam
{
    int index;
    char *name;
} TestTJParam;

static TestTJParam testParams[] = {
    {TJPARAM_FASTUPSAMPLE, "TJPARAM_FASTUPSAMPLE"},
    {TJPARAM_LOSSLESSPT, "TJPARAM_LOSSLESSPT"},
    {TJPARAM_BOTTOMUP, "TJPARAM_BOTTOMUP"},
    {TJPARAM_FASTDCT, "TJPARAM_FASTDCT"},
    {TJPARAM_LOSSLESS, "TJPARAM_LOSSLESS"},
    {TJPARAM_OPTIMIZE, "TJPARAM_OPTIMIZE"},
    {TJPARAM_PROGRESSIVE, "TJPARAM_PROGRESSIVE"},
    {TJPARAM_ARITHMETIC, "TJPARAM_ARITHMETIC"},
    {TJPARAM_NOREALLOC, "TJPARAM_NOREALLOC"},
    {TJPARAM_RESTARTROWS, "TJPARAM_RESTARTROWS"}};

```

```
46      static unsigned int testParamLen = sizeof(testParams)/sizeof(TestTJParam);
47
48
49      void do_test_with_params(const unsigned char *data, size_t size, enum TJPF tjpf, int
        ↪  fastupsample, int bottomup, int fastdct)
50      {
51          tjhandle handle = NULL;
52          void *dstBuf = NULL;
53          int width = 0, height = 0, precision, sampleSize, pfi;
54
55          char env[18] = "JSIMD_FORCENONE=1";
56
57          /* The libjpeg-turbo SIMD extensions produce false positives with
58              MemorySanitizer. */
59          putenv(env);
60
61          if ((handle = tj3Init(TJINIT_DECOMPRESS)) == NULL)
62              goto bailout;
63
64          if (tj3DecompressHeader(handle, data, size) < 0)
65          {
66              goto bailout;
67          }
68
69          width = tj3Get(handle, TJPARAM_JPEGWIDTH);
70          height = tj3Get(handle, TJPARAM_JPEGHEIGHT);
71          precision = tj3Get(handle, TJPARAM_PRECISION);
72          sampleSize = (precision > 8 ? 2 : 1);
73
74          /* Ignore 0-pixel images and images larger than 1 Megapixel, as Google's
75             OSS-Fuzz target for libjpeg-turbo did.  Casting width to (uint64_t)
76             prevents integer overflow if width * height > INT_MAX. */
77          if (width < 1 || height < 1 || (uint64_t)width * height > 1048576)
78          {
79              goto bailout;
80          }
81
82          tj3Set(handle, TJPARAM_SCANLIMIT, 500);
83
84          int w = width, h = height;
85          int i;
86          int64_t sum = 0;
87
88          tj3Set(handle, TJPARAM_FASTUPSAMPLE, fastupsample);
89          tj3Set(handle, TJPARAM_BOTTOMUP, bottomup);
90          tj3Set(handle, TJPARAM_FASTDCT, fastdct);
91
92          if (!tj3Get(handle, TJPARAM_LOSSLESS))
93          {
94              tjscalingfactor sf = {1, 2};
95              tj3SetScalingFactor(handle, sf);
96              w = TJSCALED(width, sf);
```

```
97              h = TJSCALED(height, sf);
98
99              if (w >= 97 && h >= 75)
100             {
101                 tjregion cr = {32, 16, 65, 59};
102                 tj3SetCroppingRegion(handle, cr);
103             }
104             else
105                 tj3SetCroppingRegion(handle, TJUNCROPPED);
106         }
107
108         if ((dstBuf = malloc(w * h * tjPixelSize[tjpf] * sampleSize)) == NULL)
109             goto bailout;
110
111         if (precision == 8)
112         {
113             if (tj3Decompress8(handle, data, size, (unsigned char *)dstBuf, 0,
114                                tjpf) == 0)
115             {
116                 /* Touch all of the output pixels in order to catch uninitialized reads
117                    when using MemorySanitizer. */
118                 for (i = 0; i < w * h * tjPixelSize[tjpf]; i++)
119                     sum += ((unsigned char *)dstBuf)[i];
120             }
121             else
122                 goto bailout;
123         }
124         else if (precision == 12)
125         {
126             if (tj3Decompress12(handle, data, size, (short *)dstBuf, 0, tjpf) == 0)
127             {
128                 /* Touch all of the output pixels in order to catch uninitialized reads
129                    when using MemorySanitizer. */
130                 for (i = 0; i < w * h * tjPixelSize[tjpf]; i++)
131                     sum += ((short *)dstBuf)[i];
132             }
133             else
134                 goto bailout;
135         }
136         else
137         {
138             if (tj3Decompress16(handle, data, size, (unsigned short *)dstBuf, 0,
139                                 tjpf) == 0)
140             {
141                 /* Touch all of the output pixels in order to catch uninitialized reads
142                    when using MemorySanitizer. */
143                 for (i = 0; i < w * h * tjPixelSize[tjpf]; i++)
144                     sum += ((unsigned short *)dstBuf)[i];
145             }
146             else
147                 goto bailout;
148         }
```

```
149
150         free(dstBuf);
151         dstBuf = NULL;
152
153     bailout:
154         free(dstBuf);
155         tj3Destroy(handle);
156     }
157
158     void do_test(unsigned char *data, size_t size)
159     {
160         tjhandle handle = NULL;
161         void *dstBuf = NULL;
162         int width = 0, height = 0, precision, sampleSize, pfi;
163         /* TJPF_RGB-TJPF_BGR share the same code paths, as do TJPF_RGBX-TJPF_XRGB and
164            TJPF_RGBA-TJPF_ARGB.  Thus, the pixel formats below should be the minimum
165            necessary to achieve full coverage. */
166         enum TJPF pixelFormats[4] =
167             {TJPF_RGB, TJPF_BGRX, TJPF_GRAY, TJPF_CMYK};
168
169         const char *pixelFormatStr[] =
170             {"TJPF_RGB", "TJPF_BGRX", "TJPF_GRAY", "TJPF_CMYK"};
171
172         /*
173           IMPORTANT NOTE: data[0] = pixel format, data[1],data[2] = TJPARAM_*
174         */
175         pfi = data[0] % 4;
176
177         //printf("pfi: %s\n", pixelFormatStr[pfi]);
178
179         unsigned short tjparams = 0;
180         tjparams = data[1] << 8 | data[2];
181
182         data += 3;
183         size -= 3;
184
185         if ((handle = tj3Init(TJINIT_DECOMPRESS)) == NULL)
186             goto bailout;
187
188         if (tj3DecompressHeader(handle, data, size) < 0)
189         {
190             goto bailout;
191         }
192
193         width = tj3Get(handle, TJPARAM_JPEGWIDTH);
194         height = tj3Get(handle, TJPARAM_JPEGHEIGHT);
195         precision = tj3Get(handle, TJPARAM_PRECISION);
196         sampleSize = (precision > 8 ? 2 : 1);
197
198         /* Ignore 0-pixel images and images larger than 1 Megapixel, as Google's
199            OSS-Fuzz target for libjpeg-turbo did.  Casting width to (uint64_t)
200            prevents integer overflow if width * height > INT_MAX. */
```

```
201         if (width < 1 || height < 1 || (uint64_t)width * height > 1048576)
202         {
203             goto bailout;
204         }
205 
206         tj3Set(handle, TJPARAM_SCANLIMIT, 500);
207 
208         int w = width, h = height;
209         int pf = pixelFormats[pfi], i;
210         int64_t sum = 0;
211 
212         for (int i=0; i<testParamLen; i++)
213         {
214             if (tjparams & (1 << i)) {
215                 printf("Setting %s to 1\n", testParams[i].name);
216                 tj3Set(handle, testParams[i].index, 1);
217             }
218         }
219 
220         if (!tj3Get(handle, TJPARAM_LOSSLESS))
221         {
222             tjscalingfactor sf = {1, 2};
223             tj3SetScalingFactor(handle, sf);
224             w = TJSCALED(width, sf);
225             h = TJSCALED(height, sf);
226 
227             if (w >= 97 && h >= 75)
228             {
229                 tjregion cr = {32, 16, 65, 59};
230                 tj3SetCroppingRegion(handle, cr);
231             }
232             else
233                 tj3SetCroppingRegion(handle, TJUNCROPPED);
234         }
235 
236         if ((dstBuf = malloc(w * h * tjPixelSize[pf] * sampleSize)) == NULL)
237             goto bailout;
238 
239         if (precision == 8)
240         {
241             if (tj3Decompress8(handle, data, size, (unsigned char *)dstBuf, 0,
242                                pf) == 0)
243             {
244                 /* Touch all of the output pixels in order to catch uninitialized reads
245                    when using MemorySanitizer. */
246                 for (i = 0; i < w * h * tjPixelSize[pf]; i++)
247                     sum += ((unsigned char *)dstBuf)[i];
248             }
249             else
250                 goto bailout;
251         }
252         else if (precision == 12)
```

```
253         {
254             if (tj3Decompress12(handle, data, size, (short *)dstBuf, 0, pf) == 0)
255             {
256                 /* Touch all of the output pixels in order to catch uninitialized reads
257                    when using MemorySanitizer. */
258                 for (i = 0; i < w * h * tjPixelSize[pf]; i++)
259                     sum += ((short *)dstBuf)[i];
260             }
261             else
262                 goto bailout;
263         }
264         else
265         {
266             if (tj3Decompress16(handle, data, size, (unsigned short *)dstBuf, 0,
267                                 pf) == 0)
268             {
269                 /* Touch all of the output pixels in order to catch uninitialized reads
270                    when using MemorySanitizer. */
271                 for (i = 0; i < w * h * tjPixelSize[pf]; i++)
272                     sum += ((unsigned short *)dstBuf)[i];
273             }
274             else
275                 goto bailout;
276         }
277 
278         free(dstBuf);
279         dstBuf = NULL;
280 
281     bailout:
282         free(dstBuf);
283         tj3Destroy(handle);
284     }
285 
286     void make_testimage(const char *path)
287     {
288         tjhandle handle = NULL;
289         void *dstBuf = NULL;
290         int width = 0, height = 0, precision, sampleSize, pfi;
291         /* TJPF_RGB-TJPF_BGR share the same code paths, as do TJPF_RGBX-TJPF_XRGB and
292            TJPF_RGBA-TJPF_ARGB.  Thus, the pixel formats below should be the minimum
293            necessary to achieve full coverage. */
294         enum TJPF pixelFormats[4] =
295             {TJPF_RGB, TJPF_BGRX, TJPF_GRAY, TJPF_CMYK};
296 
297         /*
298             tj3Set(handle, TJPARAM_BOTTOMUP, pfi == 0);
299             tj3Set(handle, TJPARAM_FASTUPSAMPLE, pfi == 0);
300             tj3Set(handle, TJPARAM_FASTDCT, pfi == 0);
301         */
302         unsigned short tjparams = (1 << 2) | (1 << 0) | (1 << 3);
303 
304         unsigned char data[3] = {0};
```

```
305         data[0] = TJPF_RGB;
306         data[1] = (char)tjparams >> 8;
307         data[2] = (char)tjparams;
308 
309         FILE *fp = fopen("testcase", "wb");
310         if (fp)
311         {
312             fwrite(data, 1, sizeof(data), fp);
313 
314             FILE *fp2 = fopen(path, "rb");
315             if (fp2)
316             {
317                 unsigned char imgdata[200 * 1024];
318                 int nread = fread(imgdata, 1, sizeof(imgdata), fp2);
319                 fclose(fp2);
320 
321                 int nwritten = fwrite(imgdata, 1, nread, fp);
322                 printf("Written bytes: %d\n", 5 + nwritten);
323             }
324             else
325             {
326                 printf("Error opening file!");
327             }
328 
329             fclose(fp);
330 
331             unsigned char imgdata[200 * 1024];
332             fp = fopen("testcase", "rb");
333             if (fp)
334             {
335                 int nread = fread(imgdata, 1, sizeof(imgdata), fp);
336                 fclose(fp);
337 
338                 printf("All done. Testing it now...\n");
339 
340                 do_test(imgdata, nread);
341             }
342         }
343     }
344 
345     //#define MAKE_TESTIMAGE 1
346     //#define LOAD_TESTFILE 1
347     //#define TEST_WITH_PARAM 1
348 
349     int main(int argc, char **argv)
350     {
351     #ifdef MAKE_TESTIMAGE
352         make_testimage(argv[1]);
353         return 0;
354     #elif LOAD_TESTFILE
355         FILE *fp = fopen(argv[1], "rb");
356         if (fp)
```

```
357         {
358             unsigned char imgdata[200 * 1024];
359             int nread = fread(imgdata, 1, sizeof(imgdata), fp);
360             fclose(fp);
361 
362             if (nread < 4)
363             {
364                 printf("not enough data for test!");
365             }
366 
367             do_test(imgdata, nread);
368         }
369         else
370         {
371             printf("Error opening file\n");
372         }
373 
374         return 0;
375     #elif TEST_WITH_PARAM
376         FILE *fp = fopen(argv[1], "rb");
377         if (!fp) {
378             printf("Error opening file\n");
379             return 1;
380         }
381 
382         unsigned char data[200*1024] = {0};
383         size_t size = fread(data, 1, sizeof(data), fp);
384         fclose(fp);
385 
386         do_test_with_params(data, size, TJPF_RGB, 1, 0, 0);
387         return 0;
388     #endif
389 
390         ssize_t len;        /* how much input did we read? */
391         unsigned char *buf; /* test case buffer pointer    */
392 
393         __AFL_INIT();
394         buf = __AFL_FUZZ_TESTCASE_BUF; // this must be assigned before __AFL_LOOP!
395 
396         while (__AFL_LOOP(UINT_MAX))
397         {
398             len = __AFL_FUZZ_TESTCASE_LEN;
399             if (len < MINIMAL_TESTCASE_LEN)
400                 continue;
401 
402             unsigned char *buf_new = buf;
403 
404             do_test(buf_new, len);
405         }
406 
407         return 0;
408     }
```

Listing A.4: tj3Decompress() Listing

Crashing test inputs (first 3 bytes contain pixel format, and decompress parameters) are provided below. For all of the 3 inputs, the following parameters are set:

- *TJPARAM_FASTUPSAMPLE=1*
- *TJPARAM_BOTTOMUP=1*
- *TJPARAM_FASTDCT=1*

The test files can be loaded by setting `#define LOAD_TESTFILE 1` in the test harness before compilation.

For the first test case, the pixel format is set to 0 (which equals `TJPF_RGB`):

```
00000000: 0000 0dff d8ff e000 104a 4649 3030 3030  .........JFI0000
00000010: 3030 3030 0100 00ff db00 4300 3030 3030  0000......C.0000
00000020: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000030: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000040: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000050: 3030 3030 3030 3030 3030 3030 ffdb 0043  000000000000...C
00000060: 0109 0909 3030 3030 3030 3030 3030 3030  ....000000000000
00000070: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000080: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000090: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
000000a0: 32ff c300 110c 0001 0001 0301 2200 0211  2...........".."...
000000b0: 0103 1101 ffc4 001f 0000 0105 0101 0101  ................
000000c0: 0101 0000 0000 0000 0000 0102 0000 0010  ................
000000d0: 0708 090a 0bff c400 b510 0002 0103 0302  ................
000000e0: 0403 0505 0404 0000 017d 2102 0300 0411  .........}!.....
000000f0: 0512 2131 4106 1351 6007 2271 1432 8191  ..!1A..Q`."q.2..
00000100: a108 2342 b1c0 ff52 d100 0033 4072 8209  ..#B...R...3@r..
00000110: 0a63 1718 191a 2526 2728 262a 3435 3637  .c....%&'(&*4567
00000120: 38b5 b6b7 b8b9 9ec2 c3c4 c5c6 c7c8 c9ca  8...............
00000130: d2d3 d4d5 d6d7 d8d9 0000 03e8 7176 7778  ............qvwx
00000140: 7879 7a83 8485 8687 8889 8a92 9394 9596  xyz.............
00000150: 9798 999a a2a3 a4a5 a6a7 a2a9 aab2 c1b4  ................
00000160: b5b6 b79a b9ba c2c3 c4c5 c6c7 c8c9 cad2  ................
00000170: d3d4 d500 80ff ffda e1e2 e3e4 e5e6 e7e8  ................
00000180: e9ea f1f2 f3f4 f5f6 f7f8 f9fa ffc4 001f  ................
00000190: 0100 0301 0101 0101 0101 0101 0000 0000  ................
000001a0: 0010 0102 0304 0506 0708 090a 0bff dc00  ................
000001b0: b511 0002 0102 0404 0304 0705 0404 0001  ................
000001c0: 0277 0001 0203 1104 ffff ff7f 1241 5107  .w...........AQ.
000001d0: 6171 1322 3281 07f6 4291 a1b1 c109 2333  aq."2...B.....#3
000001e0: 5200 0100 80d1 0a16 2434 e125 f117 1819  R.......$4.%....
000001f0: 1a26 2728 292a 3573 6364 6566 6b68 696a  .&'()*5scdefkhij
00000200: 7374 7576 7778 797a 8384 8586 8788 6566  stuvwxyz......ef
00000210: 6768 696a 7373 7587 7778 7991 8283 8485  ghijssu.wxy.....
```

```
00000220: 8687 8889 9f92 9394 9596 979c 999a a2bd  ................
00000230: a4ee a6a7 a8a9 aab2 b3b4 393a 4344 4546  ..........9:CDEF
00000240: 4748 4900 0000 4056 5758 594a 6364 6566  GHI...@VWXYJcdef
00000250: 6768 696a 7374 7576 7779 7a83 8485 86f6  ghijstuvwyz.....
00000260: f7f8 f9fa ffda 000c 0301 0002 1103 0001  ................
00000270: 0000 03e8                                ....
```

Listing A.5: TJPF_RGB Testase

For the second test case, the pixel format is set to the pixel format is set to 0 (which equals `TJPF_BGRX`).

```
00000000: 0100 0dff d8ff e000 104a 4649 3030 3030  .........JFI0000
00000010: 3030 3030 0100 00ff db00 4300 3030 3030  0000......C.0000
00000020: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000030: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000040: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000050: 3030 3030 3030 3030 3030 3030 ffdb 0043  000000000000...C
00000060: 0109 0909 3030 3030 3030 3030 3030 3030  ....000000000000
00000070: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000080: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000090: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
000000a0: 32ff c300 110c 0001 0001 0301 2200 0211  2...........".."...
000000b0: 0103 1101 ffc4 001f 0000 0105 0101 0101  ................
000000c0: 0101 0000 0000 0000 0000 0102 0000 0010  ................
000000d0: 0708 090a 0bff c400 b510 0002 0103 0302  ................
000000e0: 0403 0505 0404 0000 017d 0102 0300 0411  .........}......
000000f0: 0512 2131 4106 1351 6007 2271 1432 8191  ..!1A..Q`."q.2..
00000100: a108 2342 b1c0 ff52 d100 0033 4072 8209  ..#B...R...3@r..
00000110: 0a16 1718 191a 2526 3128 262a 341c 3637  ......%&1(&*4.67
00000120: 38b5 b6b7 b8b9 9ec2 c3c4 c5c6 c7c8 c9ca  8...............
00000130: d2d3 d4d5 d6d7 d8d9 0000 03e8 7176 7778  ............qvwx
00000140: 7879 7a83 8485 8687 8889 8a92 9394 9596  xyz.............
00000150: 9798 999a a2a3 a4a5 a6a7 a2a9 aab2 c1b4  ................
00000160: b5b6 b79a b9ba c2c3 c4c5 c6c7 c8c9 cad2  ................
00000170: d3d4 d500 80ff ffda e1e2 e3e4 e5e6 e7e8  ................
00000180: e9ea f1f2 f3f4 f5f6 f7f8 f9fa ffc4 001f  ................
00000190: 0100 0301 0101 0101 0101 0101 0000 0000  ................
000001a0: 0010 0102 0304 0506 0708 090a 0bff dc00  ................
000001b0: b511 0002 0102 0404 0304 0705 0404 0001  ................
000001c0: 0277 0001 0203 1104 ffff ff7f 1241 5107  .w...........AQ.
000001d0: 6171 1322 3281 07f6 4291 a1b1 c109 2333  aq."2...B.....#3
000001e0: 5200 0100 80d1 0a16 2434 e125 f117 1819  R.......$4.%....
000001f0: 1a26 2728 292a 3573 6364 6566 6b68 696a  .&'()*5scdefkhij
00000200: 7374 7576 7778 797a 8384 8586 8788 6566  stuvwxyz......ef
00000210: 6768 696a 7373 7587 7778 7991 8283 8485  ghijssu.wxy.....
00000220: 8687 8889 9f92 9394 9596 979c 999a a2bd  ................
00000230: a4ee a6a7 a8a9 aab2 b3b4 393a 4344 4546  ..........9:CDEF
00000240: 4748 4900 0000 4056 5758 594a 6364 6566  GHI...@VWXYJcdef
00000250: 6768 696a 7374 7576 7779 7a83 8485 86f6  ghijstuvwyz.....
```

```
00000260: f7f8 f9fa ffda 000c 0301 0002 1103 0001  ................
00000270: 0000 03e8                                ....
```

Listing A.6: TJPF_BGRX Testcase

For the third test case, the pixel format is again set to `TJPF_RGB`:

```
00000000: 0000 0dff d8ff e000 104a 4649 4600 0101  .........JFIF...
00000010: 0e20 0100 0100 00ff db00 4300 0806 4a02  . ........C...J.
00000020: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000030: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000040: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000050: 3030 3030 3030 3030 2e33 3432 ffdb 0043  00000000.342...C
00000060: 0109 0909 0c0b 0c18 3030 3030 3030 3030  ........00000000
00000070: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000080: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000090: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
000000a0: 32ff c300 110c 0001 0040 0301 2200 0211  2........@..".."
000000b0: 0103 1101 ffc4 001f 0000 0105 0101 0101  ................
000000c0: 0101 0000 0000 0000 000f 0102 0000 0010  ................
000000d0: 0708 090a 0bff c400 b510 0002 0103 0302  ................
000000e0: 0403 0505 0404 0000 017d 0102 0300 0411  .........}......
000000f0: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000100: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000110: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000120: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000130: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000140: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000150: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000160: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000170: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000180: 3030 3030 3030 3030 f7f8 f9fa ffc4 001f  00000000........
00000190: 0100 0301 0101 0101 0101 0101 0000 0000  ................
000001a0: 0010 0102 0304 0506 0708 090a 0bff c400  ................
000001b0: b511 0002 0102 0404 0304 0705 0404 0001  ................
000001c0: 0277 0001 0203 1104 3030 3030 3030 3030  .w......00000000
000001d0: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
000001e0: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
000001f0: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000200: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000210: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000220: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000230: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000240: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000250: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
00000260: f7f8 f9fa ffda 000c 0301 0002 1103 1101  ................
00000270: 0001 0203 1104 0521 3106 1241 5107 6171  .......!1..AQ.aq
00000280: 133b 3281 0814 4291 a1b1 c109 2333 5200  .;2...B.....#3R.
00000290: 0000 03e8                                ....
```

Listing A.7: TJPF_RGB Testcase