Problem 2

1. Project Name

Openjpeg

2. Source Code Version

Version 2.1.2
https://github.com/uclouvain/openjpeg/releases/tag/v2.1.2

3. PoC Downloadable from Internet?

Yes

4. CVE ID

CVE-2016-10506

5. Details Procedures that trigger the crash

a. How the project program is compiled.

The program is compiled using the installation method described by the GitHub source:

1. Go to the directory of the project program and type:
2. mkdir
3. cd build
4. cmake .. –DCMAKE_BUILD_TYPE=Release
5. make
6. sudo apt-get install liblcms2-dev libtiff-dev libpng-dev libz-dev

b. The exact running arguments

1. Go to project_program/build/bin,
2. Add in the input file “poc6.bmp”
3.type : “valgrind opj_compress -i in/poc6.bmp -o out/poc6.png”

Crash Description

The crash found occurs due to a jump based on an uninitialized variable:

```
==16692== Syscall param msg->desc.port.name points to uninitialised byte(s)
==16692==    at 0x1004A834A: mach_msg_trap (in /usr/lib/system/libsystem_kernel.dylib)
==16692==    by 0x1004A7796: mach_msg (in /usr/lib/system/libsystem_kernel.dylib)
==16692==    by 0x1004A1485: task_set_special_port (in /usr/lib/system/libsystem_kernel.dylib)
==16692==    by 0x10063D10E: _os_trace_create_debug_control_port (in /usr/lib/system/libsystem_trace.dylib)
==16692==    by 0x10063D458: _libtrace_init (in /usr/lib/system/libsystem_trace.dylib)
==16692==    by 0x1001A69DF: libSystem_initializer (in /usr/lib/libSystem.B.dylib)
==16692==    by 0x100036A1A: ImageLoaderMachO::doModInitFunctions(ImageLoader::LinkContext const&) (in /usr/lib/dyld)
==16692==    by 0x100036C1D: ImageLoaderMachO::doInitialization(ImageLoader::LinkContext const&) (in /usr/lib/dyld)
==16692==    by 0x1000324A9: ImageLoader::recursiveInitialization(ImageLoader::LinkContext const&, unsigned int, char
==16692==    by 0x100032440: ImageLoader::recursiveInitialization(ImageLoader::LinkContext const&, unsigned int, char
==16692==    by 0x100031523: ImageLoader::processInitializers(ImageLoader::LinkContext const&, unsigned int, ImageLoa
==16692==    by 0x1000315B8: ImageLoader::runInitializers(ImageLoader::LinkContext const&, ImageLoader::InitializerTi
==16692==  Address 0x1048a9cec is on thread 1's stack
==16692==  in frame #2, created by task_set_special_port (???:)
==16692==
```

The crash below was caused by an Out-of-Bounds Read issue found in function bmp24toimage of convertbmp.c. This means OpenJpeg reads the data past the end or before the beginning of the intended buffer. The root cause of this issue was an Integer Overflow issue, causing the values of the integers to be wrong. Hence, Openjpeg is unable to read the value and terminate the program. The issue is shown below:

```
==16692== Invalid read of size 1
==16692==    at 0x10000D5C4: bmptoimage (in ./opj_compress)
==16692==    by 0x1000028D5: main (in ./opj_compress)
==16692==  Address 0x100c8bce5 is 1 bytes after a block of size 4 alloc'd
==16692==    at 0x1000B6C7A: calloc (in /usr/local/Cellar/valgrind/3.13.0/lib/valgrind/vgpreload_memcheck-amd64-darwin.so)
==16692==    by 0x10000C978: bmptoimage (in ./opj_compress)
==16692==    by 0x1000028D5: main (in ./opj_compress)
==16692==
==16692== Invalid read of size 1
==16692==    at 0x10000D5CF: bmptoimage (in ./opj_compress)
==16692==    by 0x1000028D5: main (in ./opj_compress)
==16692==  Address 0x100c8bce4 is 0 bytes after a block of size 4 alloc'd
==16692==    at 0x1000B6C7A: calloc (in /usr/local/Cellar/valgrind/3.13.0/lib/valgrind/vgpreload_memcheck-amd64-darwin.so)
==16692==    by 0x10000C978: bmptoimage (in ./opj_compress)
==16692==    by 0x1000028D5: main (in ./opj_compress)
==16692==
==16692== Invalid read of size 1
==16692==    at 0x10000D5D8: bmptoimage (in ./opj_compress)
==16692==    by 0x1000028D5: main (in ./opj_compress)
==16692==  Address 0x100c8bce6 is 2 bytes after a block of size 4 alloc'd
==16692==    at 0x1000B6C7A: calloc (in /usr/local/Cellar/valgrind/3.13.0/lib/valgrind/vgpreload_memcheck-amd64-darwin.so)
==16692==    by 0x10000C978: bmptoimage (in ./opj_compress)
==16692==    by 0x1000028D5: main (in ./opj_compress)
==16692==
==16692==
==16692== Process terminating with default action of signal 11 (SIGSEGV)
==16692==  Access not within mapped region at address 0x101082000
==16692==    at 0x10000D5C4: bmptoimage (in ./opj_compress)
==16692==    by 0x1000028D5: main (in ./opj_compress)
==16692==  If you believe this happened as a result of a stack
==16692==  overflow in your program's main thread (unlikely but
==16692==  possible), you can try to increase the size of the
==16692==  main thread stack using the --main-stacksize= flag.
==16692==  The main thread stack size used in this run was 8388608.
```

Furthermore, there are also leaks in memory, with 48 bytes definitely lost in 1 block and 192 indirectly lost in 1 block:

```
==16692== HEAP SUMMARY:
==16692==     in use at exit: 12,884,924,357 bytes in 169 blocks
==16692==   total heap usage: 185 allocs, 16 frees, 12,884,930,501 bytes allocated
==16692==
==16692== LEAK SUMMARY:
==16692==    definitely lost: 48 bytes in 1 blocks
==16692==    indirectly lost: 192 bytes in 1 blocks
==16692==      possibly lost: 4,294,967,372 bytes in 4 blocks
==16692==    still reachable: 8,589,938,900 bytes in 10 blocks
==16692==         suppressed: 17,845 bytes in 153 blocks
==16692== Rerun with --leak-check=full to see details of leaked memory
==16692==
```

Brief explanation about bug fixed.

Overflow check operations were added to convertbmp.c file to ensure that the fault does not occur.