# Problem 2

## 1. Project Name

ImageMagick

## 2. Source Code Version

Version 7.0.6.0
https://github.com/ImageMagick/ImageMagick/releases/tag/7.0.6-0

## 3. PoC Downloadable from Internet?

Yes

## 4. CVE ID

CVE-2017-14684

## 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. make
3. sudo make install
4. magick --version

### b. The exact running arguments

1. Add in the input file “poc4.svg”
2. Type: “valgrind convert in/poc4.svg out/poc4”

# Crash Description

The picture below shows the memory of the jump where, for example, the error occurred at 0x1001C6954 caused by the root address 0x1005A650C.

```
==9275== Syscall param msg->desc.port.name points to uninitialised byte(s)
==9275==    at 0x10084534A: mach_msg_trap (in /usr/lib/system/libsystem_kernel.dylib)
==9275==    by 0x100844796: mach_msg (in /usr/lib/system/libsystem_kernel.dylib)
==9275==    by 0x10083E485: task_set_special_port (in /usr/lib/system/libsystem_kernel.dylib)
==9275==    by 0x1009DA10E: _os_trace_create_debug_control_port (in /usr/lib/system/libsystem_trace.dylib)
==9275==    by 0x1009DA458: _libtrace_init (in /usr/lib/system/libsystem_trace.dylib)
==9275==    by 0x1005439DF: libSystem_initializer (in /usr/lib/libSystem.B.dylib)
==9275==    by 0x100018A1A: ImageLoaderMachO::doModInitFunctions(ImageLoader::LinkContext const&) (in /usr/
==9275==    by 0x100018C1D: ImageLoaderMachO::doInitialization(ImageLoader::LinkContext const&) (in /usr/li
==9275==    by 0x1000144A9: ImageLoader::recursiveInitialization(ImageLoader::LinkContext const&, unsigned
==9275==    by 0x100014440: ImageLoader::recursiveInitialization(ImageLoader::LinkContext const&, unsigned
==9275==    by 0x100013523: ImageLoader::processInitializers(ImageLoader::LinkContext const&, unsigned int,
==9275==    by 0x1000135B8: ImageLoader::runInitializers(ImageLoader::LinkContext const&, ImageLoader::Init
==9275==  Address 0x10488bd7c is on thread 1's stack
==9275==  in frame #2, created by task_set_special_port (???:)
==9275==
==9275== Conditional jump or move depends on uninitialised value(s)
==9275==    at 0x1005A650C: generate_block (in /usr/lib/system/libcorecrypto.dylib)
==9275==    by 0x1005A61F9: drbg_update (in /usr/lib/system/libcorecrypto.dylib)
==9275==    by 0x1005A5B3C: nistctr_init (in /usr/lib/system/libcorecrypto.dylib)
==9275==    by 0x1005A544D: init (in /usr/lib/system/libcorecrypto.dylib)
==9275==    by 0x100721383: arc4_init (in /usr/lib/system/libsystem_c.dylib)
==9275==    by 0x10072122E: arc4random (in /usr/lib/system/libsystem_c.dylib)
==9275==    by 0x10072166D: arc4random_uniform (in /usr/lib/system/libsystem_c.dylib)
==9275==    by 0x10073C227: find_temp_path (in /usr/lib/system/libsystem_c.dylib)
==9275==    by 0x10073C4CF: mkstemp (in /usr/lib/system/libsystem_c.dylib)
==9275==    by 0x1001BB9C1: GenerateEntropicChaos (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x1001BB477: AcquireRandomInfo (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x1001C6954: AcquireUniqueFileResource (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
```

At the same time, the use of uninitialized value also appeared at the address 0x100000D90 caused by the root address 0x10073C22A:

```
==9275== Use of uninitialised value of size 8
==9275==    at 0x10073C22A: find_temp_path (in /usr/lib/system/libsystem_c.dylib)
==9275==    by 0x10073C4CF: mkstemp (in /usr/lib/system/libsystem_c.dylib)
==9275==    by 0x1001BB9C1: GenerateEntropicChaos (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x1001BB4E2: AcquireRandomInfo (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x1001C6954: AcquireUniqueFileResource (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x1001F1998: AcquireUniqueFilename (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x1000E56CE: InvokeDelegate (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x1000DCC9B: ReadImage (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x1000DDA36: ReadImages (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x100433F12: ConvertImageCommand (in /usr/local/lib/libMagickWand-7.Q16HDRI.0.dylib)
==9275==    by 0x10048E782: MagickCommandGenesis (in /usr/local/lib/libMagickWand-7.Q16HDRI.0.dylib)
==9275==    by 0x100000D90: main (in /usr/local/bin/convert)
==9275==
==9275== Conditional jump or move depends on uninitialised value(s)
==9275==    at 0x100721676: arc4random_uniform (in /usr/lib/system/libsystem_c.dylib)
==9275==    by 0x10073C227: find_temp_path (in /usr/lib/system/libsystem_c.dylib)
==9275==    by 0x10073C4CF: mkstemp (in /usr/lib/system/libsystem_c.dylib)
==9275==    by 0x1001C6A04: AcquireUniqueFileResource (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x1001F1998: AcquireUniqueFilename (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x1000E56CE: InvokeDelegate (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x1000DCC9B: ReadImage (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x1000DDA36: ReadImages (in /usr/local/lib/libMagickCore-7.Q16HDRI.2.dylib)
==9275==    by 0x100433F12: ConvertImageCommand (in /usr/local/lib/libMagickWand-7.Q16HDRI.0.dylib)
==9275==    by 0x10048E782: MagickCommandGenesis (in /usr/local/lib/libMagickWand-7.Q16HDRI.0.dylib)
==9275==    by 0x100000D90: main (in /usr/local/bin/convert)
==9275==
```

The crash occurs due to heap-buffer-overflow in the GetNextToken(). The buffer for GetNextToken() was allocated in the heap portion of the memory and was allocated using a routine, malloc(). As shown below, there are leaks in memory, with 288 bytes definitely lost in 4 blocks and 4,096 indirectly lost in 2 blocks:

```
==9276== HEAP SUMMARY:
==9276==     in use at exit: 597,694 bytes in 2,038 blocks
==9276==   total heap usage: 5,176 allocs, 3,138 frees, 1,617,004 bytes allocated
==9276==
==9276== LEAK SUMMARY:
==9276==    definitely lost: 288 bytes in 4 blocks
==9276==    indirectly lost: 4,096 bytes in 2 blocks
==9276==      possibly lost: 72 bytes in 3 blocks
==9276==    still reachable: 569,067 bytes in 1,866 blocks
==9276==         suppressed: 24,171 bytes in 163 blocks
==9276== Rerun with --leak-check=full to see details of leaked memory
==9276==
```

# Brief explanation about bug fixed.

The bug is being fixed by removing the dangling pointers and checking for buffer overflow.