Null Pointer Dereference Bug of p7zip

# Bug Overview

A *null pointer dereference* bug affects the current and many old versions of p7zip. It is because the lack of check for the array variable folders.PackPositions after a loop of initialization. Details information is depicted in **Table 1**. 7-zip is a file archiver and p7zip is its linux version. For more information of 7-zip, see <http://www.7-zip.org/>

**Table 1** A null pointer dereference bug of p7zip

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| program | version | Platform | Cause&Effect | CWE |
| 7za，7z，7zr | 16.02, 15.14.1, 15.14,  15.09,  9.38.1,  9.38 | 32bit and 64bit Linux | A lack of null pointer check for the variable folders.PackPositions in function: “CInArchive::ReadAndDecodePackedStreams” in CPP/7zip/Archive/7z/7zIn.cpp may cause a crash of p7zip when decoding malformed 7zip files. | CWE-476,  Null pointer dereference |

# Bug Details

## Proof of Concept(PoC)

### How to Trigger It

Here is an example to trigger this bug in Ubuntu14.04i386 LTS.

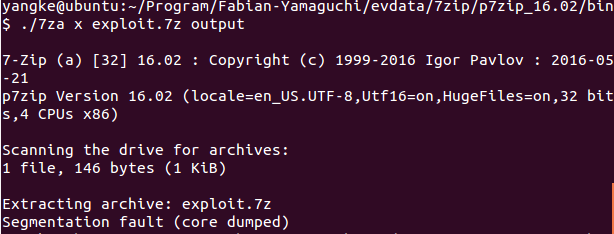
cd p7zip

cp makefile.linux\_x32 makefile

make(make 7z or make 7zr) # program bin/7za(bin/7z or bin/7zr) will be created.

cd bin/

./7za x exploit.7z output #exploit.7z is a malformed input file.

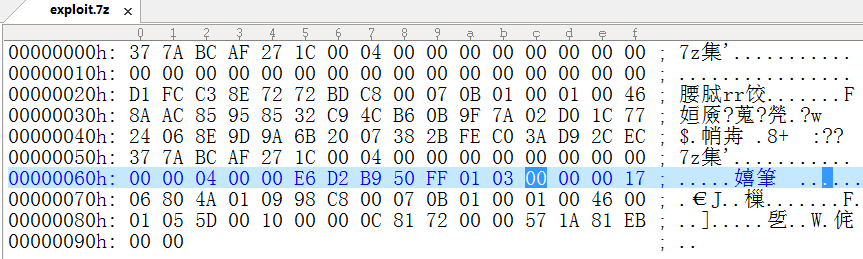


**Figure 1** A Crash caused by command ./7za x exploit.7z output

**Figure 1** shows the crash scene of the extraction. And the following file exploit.7z is the malformed input.



**Figure 2** shows the content of this file. The highlighted one byte is the key to trigger the crash. It, together with the following three bytes, constructs an information type field in 7zip format. And it is located at the third position of 40-Byte logic chunk. When the left most bit of this byte is zero it causes the dereference of null pointer folders.PackPositions in function CInArchive::ReadAndDecodePackedStreams. This triggers a *null pointer dereference* error(folders.PackPositions is null).



**Figure 2** the content of exploit.7z

### Crash Trace Analysis of 7za

p7zip uses option “-s” to tell gcc to remove the symbol table. Use the following commands to remove this and optimization option in Makefile and recompile p7zip. Note that make clean will remove the bin directory and recreate it.

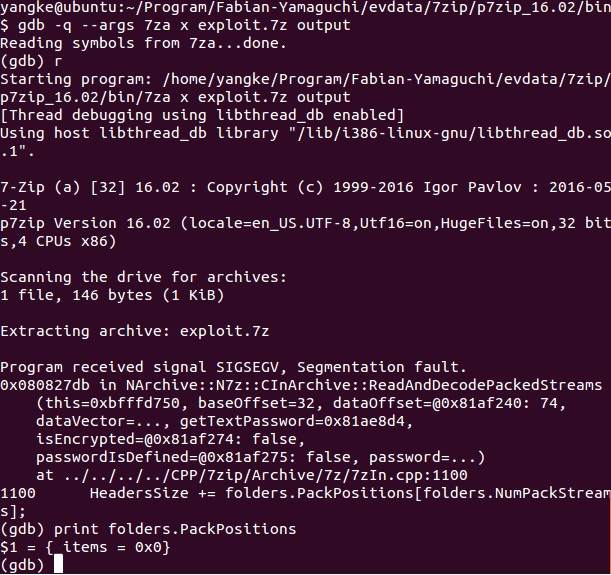
sed -i 's/=\-O \-s/=/g' `grep "OPTFLAGS=\-O \-s" . -rl`

sed -i 's/=\-O2 \-msse2 \-s/=\-msse2/g' `grep "\-O2 \-msse2 \-s" . -rl`

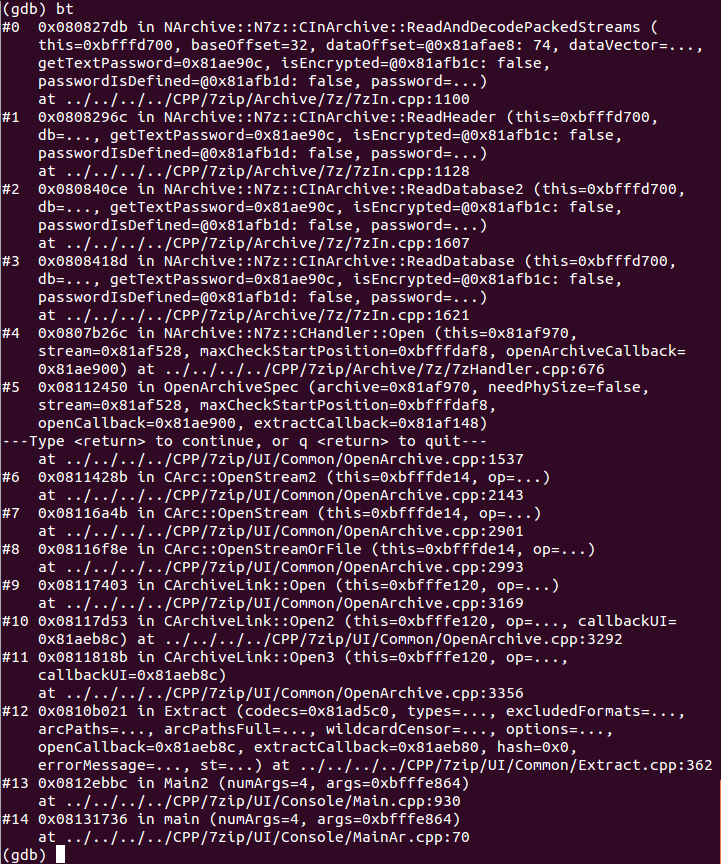
make clean && make

**Figure 3** shows the value of folders.PackPositions at crash point. It’s a null pointer and it causes the crash. **Figure 4** shows the crash stack.

The variable folders is initialized by calling ReadStreamInfo at line 1057 in file CPP/Archive/7z/7zIn.cpp(see **Figure 5**). It is located at the same function: CInArchive::ReadAndDecodePackedStreams. This function is called twice in this example, the crash happened during the second call.



**Figure 3** Use gdb to debug program 7za.

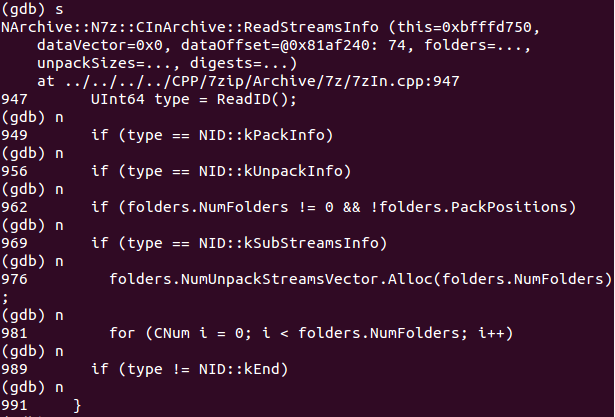


**Figure 4** Thecrash stack of 7za.

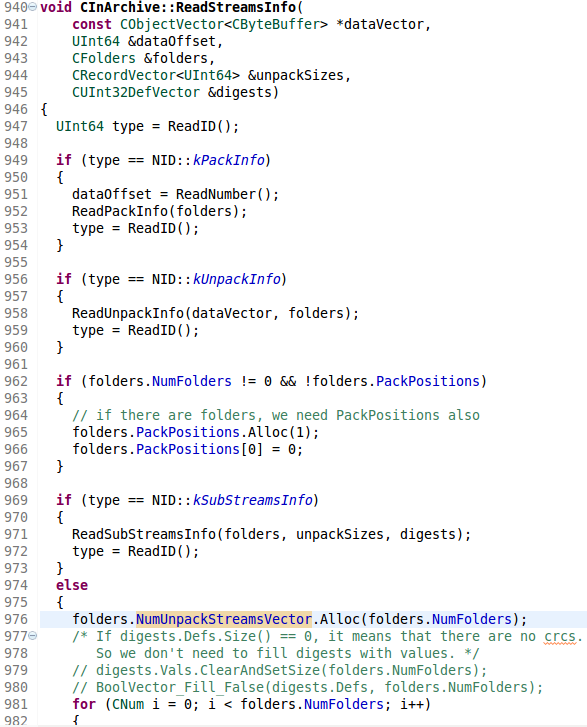
## 

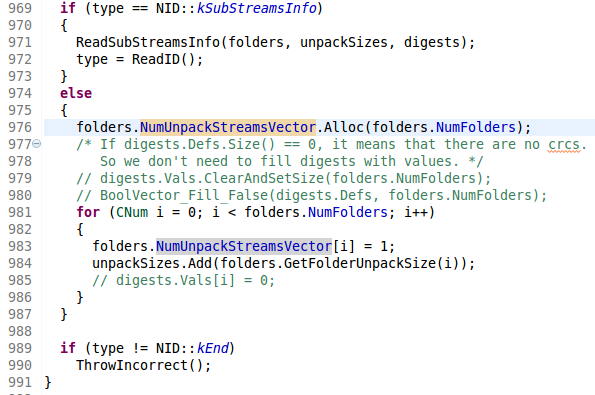
**Figure 5** The initialization of folders by calling ReadStreamInfo

By examining the ReadStreamInfo, we find that the returned variable type of function ReadID() in line 947 mismatches all the if condition, thus causing the missing initialization of folders.PackPositions(see **Figure 6** and **Figure 7**).



**Figure 6** The return variable type causes the missing of the initialization of folders.PackPositions.



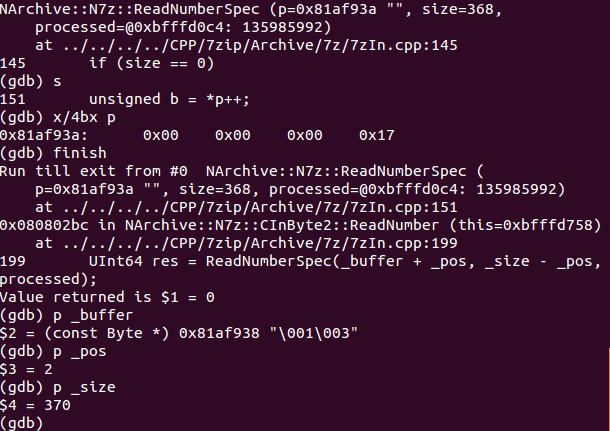


**Figure 7** The source code of function: ReadStreamInfo.

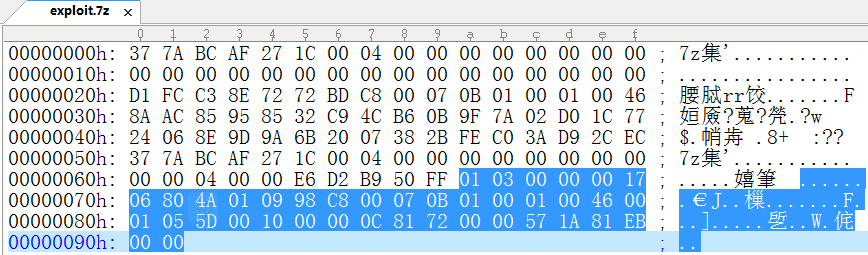
As shows in **Figure 8** and **Figure 9**, the final return value of ReadStreamInfo is controlled by the third byte in \_buffer(The type of pointer p in **Figure 8** is Byte \*). If the highest bit of this byte is not 1 then this Byte variable will be returned to the variable type in ReadStreamInfo. After further debugging, we find that the content of \_buffer(40 Bytes) corresponds to the file content starting from byte position **6ah** to **91h** highlighted in **Figure 10**. In this example, the third byte of this chunk is 0x00.



**Figure 8** The source code of function: ReadStreamInfo

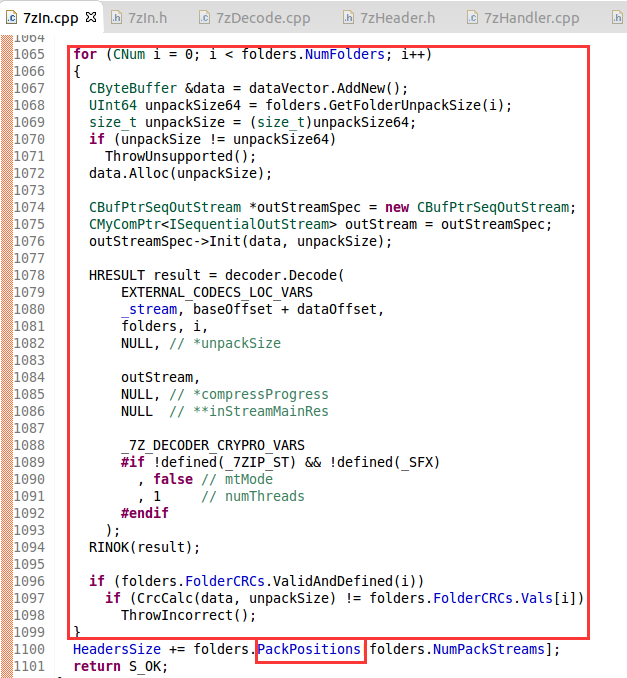


**Figure 9** The value of \_buffer, \_pos and \_size in ReadNumberSpec()



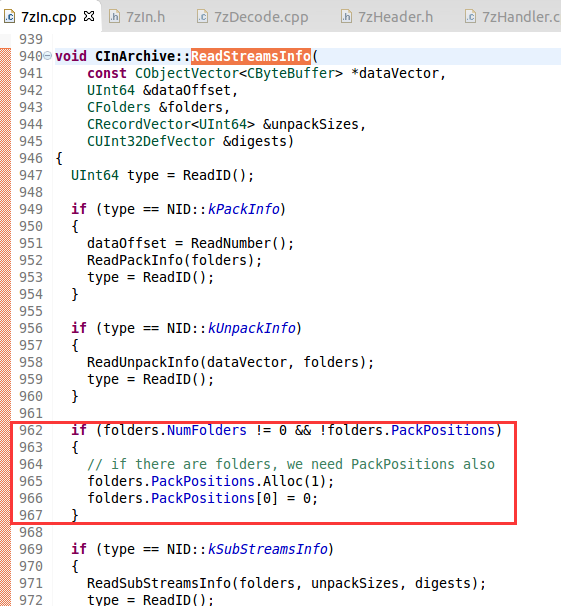
**Figure 10** The last 40 bytes in the file which corresponds to the content of \_buffer.

By reviewing this trace, it’s easy to find that the problem is located at function CInArchive::ReadAndDecodePackedStreams. It lacks a null pointer check for folders.PackPositions when folders.size() is zero(This means the loop times of for in line 1065 of file 7zIn.cpp is zero).



**Figure 11** CInArchive::ReadAndDecodePackedStreams ignored the null pointer under the condition of zero loop time.

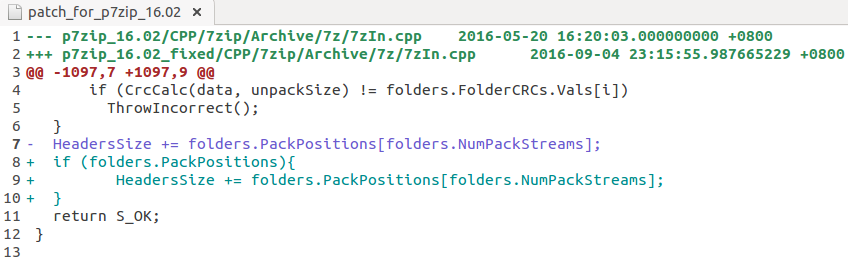
It should be pointed out that function ReadStreamsInfo has tried to handle the exception scenario when folder.NumFolders!=0 and folders.PackPositions is null. But the condition doesn’t holds in the crash example we discuss here(because folder.NumFolders==0). So it bypasses this exception check.



**Figure 12** The handling of null folders.PackPositions in function ReadStreamInfo().

# Possible Patch

Add the null pointer check for variable folders.PackPositions in function CInArchive::ReadAndDecodePackedStreams(see **Figure 12**) .



**Figure 12** A patch for p7zip\_16.02

We find that p7zip of version 16.02、15.14.1、15.14、15.09、9.38.1、9.38 have the same problem. Earlier version of p7zip doesn’t have this bug. The release time of p7zip-9.38 at sourceforge is 2015-12-12， so this is a relatively new bug of p7zip.

Here we give the patch file for p7zip16.02 as an example. cd to the extracted p7zip\_16.02 directory, copy this patch in it and use patch –p1 <patch\_for\_p7zip to patch it.

