
Checkpoint 2

p7zip

Arnav Nidumolu, Atharva Kale, Pascal von Fintel, Patrick
Negus

2023-04-06

Checkpoint 2

Contents:

- [Checkpoint 2](#)
 - [Contents:](#)
 - [Static Analysis](#)
 - [Dynamic Analysis](#)
 - * Generating a corpus
 - * Experimenting with fuzzing composition flags
 - * Parallel fuzzing
 - * Results

Static Analysis

Dynamic Analysis

```
Starting evaluation of codeql/cpp-queries/Security/CWE/CWE-704/WcharCharConversion.q1.
[42/47 eval 8ms] Evaluation done; writing results to codeql/cpp-queries/Security/CWE/CWE-676/DangerousUseOfCin.bqrs.
Starting evaluation of codeql/cpp-queries/Security/CWE/CWE-732/OpenCallMissingModeArgument.q1.
[43/47 eval 63ms] Evaluation done; writing results to codeql/cpp-queries/Security/CWE/CWE-704/WcharCharConversion.bqrs.
Starting evaluation of codeql/cpp-queries/Security/CWE/CWE-732/UnsafeDacLSecurityDescriptor.q1.
[44/47 eval 28ms] Evaluation done; writing results to codeql/cpp-queries/Security/CWE/CWE-732/OpenCallMissingModeArgument.bqrs.
Starting evaluation of codeql/cpp-queries/Summary/LinesOfCode.q1.
[45/47 eval 13ms] Evaluation done; writing results to codeql/cpp-queries/Security/CWE/CWE-732/UnsafeDacLSecurityDescriptor.bqrs.
Starting evaluation of codeql/cpp-queries/Summary/LinesOfUserCode.q1.
[46/47 eval 4ms] Evaluation done; writing results to codeql/cpp-queries/Summary/LinesOfCode.bqrs.
[47/47 eval 2.1s] Evaluation done; writing results to codeql/cpp-queries/Summary/LinesOfUserCode.bqrs.
Shutting down query evaluator.
Interpreting results.
Analysis produced the following diagnostic data:
```

Diagnostic	Summary
Successfully extracted files	44 results

```
Analysis produced the following metric data:
```

Metric	Value
Total lines of user written C/C++ code in the database	4129
Total lines of C/C++ code in the database	400756

```
→ codeql-playground git:(main) ✗ cat analysis.csv
→ codeql-playground git:(main) ✗
```

american fuzzy lop ++4.07a {main-afl-} (...Bundles/Alone2/_o/bin/7zz) [fast]

process timing		overall results	
run time	: 2 days, 13 hrs, 15 min, 48 sec	cycles done	: 149
last new find	: 0 days, 0 hrs, 12 min, 9 sec	corpus count	: 9166
last saved crash	: none seen yet	saved crashes	: 0
last saved hang	: 0 days, 0 hrs, 13 min, 43 sec	saved hangs	: 79
cycle progress		map coverage	
now processing	: 7353.33 (80.2%)	map density	: 0.91% / 5.65%
runs timed out	: 0 (0.00%)	count coverage	: 5.25 bits/tuple
stage progress		findings in depth	
now trying	: splice 1	favorable items	: 773 (8.43%)
stage execs	: 42/43 (97.67%)	new edges on	: 1488 (16.23%)
total execs	: 117M	total crashes	: 0 (0 saved)
exec speed	: 598.4/sec	total tmouts	: 298 (0 saved)
fuzzing strategy yields		item geometry	
bit flips	: disabled (default, enable with -D)	levels	: 33
byte flips	: disabled (default, enable with -D)	pending	: 140
arithmetics	: disabled (default, enable with -D)	pend fav	: 0
known ints	: disabled (default, enable with -D)	own finds	: 8281
dictionary	: n/a	imported	: 198
havoc/splice	: 5532/44.1M, 2749/73.6M	stability	: 81.95%
py/custom/rq	: unused, unused, unused, unused		
trim/eff	: disabled, disabled		

[cpu000:116%]

american fuzzy lop ++4.07a {variant-afl-tsan} (.../Alone2/_o/bin/7zz) [fast]

process timing		overall results	
run time : 2 days, 13 hrs, 11 min, 34 sec		cycles done : 1	
last new find : 0 days, 0 hrs, 16 min, 36 sec		corpus count : 7029	
last saved crash : none seen yet		saved crashes : 0	
last saved hang : 0 days, 0 hrs, 17 min, 57 sec		saved hangs : 91	
cycle progress		map coverage	
now processing : 1058.242 (15.1%)		map density : 6.77% / 24.98%	
runs timed out : 1 (0.01%)		count coverage : 5.52 bits/tuple	
stage progress		findings in depth	
now trying : splice 14		favored items : 672 (9.56%)	
stage execs : 9/12 (75.00%)		new edges on : 1251 (17.80%)	
total execs : 5.40M		total crashes : 0 (0 saved)	
exec speed : 37.03/sec (slow!)		total tmouts : 191 (0 saved)	
fuzzing strategy yields		item geometry	
bit flips : disabled (default, enable with -D)		levels : 8	
byte flips : disabled (default, enable with -D)		pending : 2705	
arithmetics : disabled (default, enable with -D)		pend fav : 0	
known ints : disabled (default, enable with -D)		own finds : 1435	
dictionary : n/a		imported : 4907	
havoc/splice : 494/775k, 941/2.11M		stability : 69.15%	
py/custom/rq : unused, unused, unused, unused			
trim/eff : 6.48%/2.46M, disabled			
		[cpu003: 150%]	

american fuzzy lop ++4.07a {variant-afl-msan} (.../Alone2/_o/bin/7zz) [fast]

process timing run time : 2 days, 13 hrs, 14 min, 31 sec last new find : 0 days, 0 hrs, 2 min, 53 sec last saved crash : 0 days, 1 hrs, 13 min, 31 sec last saved hang : 0 days, 0 hrs, 36 min, 4 sec		overall results cycles done : 1 corpus count : 7427 saved crashes : 978 saved hangs : 94
cycle progress now processing : 4889*0 (65.8%) runs timed out : 26 (0.35%)	map coverage map density : 1.05% / 5.58% count coverage : 5.37 bits/tuple	
stage progress now trying : trim 8/8 stage execs : 27/90 (30.00%) total execs : 4.77M exec speed : 30.83/sec (slow!)	findings in depth favored items : 722 (9.72%) new edges on : 1327 (17.87%) total crashes : 220k (978 saved) total tmouts : 206 (0 saved)	
fuzzing strategy yields bit flips : disabled (default, enable with -D) byte flips : disabled (default, enable with -D) arithmetics : disabled (default, enable with -D) known ints : disabled (default, enable with -D) dictionary : n/a havoc/splice : 465/343k, 1270/1.46M py/custom/rq : unused, unused, unused, unused trim/eff : 6.55%/2.91M, disabled		item geometry levels : 5 pending : 2456 pend fav : 1 own finds : 1268 imported : 5472 stability : 74.96%
		[cpu002:100%]

```
american fuzzy lop ++4.07a {variant-afl-asan} (.../Alone2/_o/bin/7zz) [fast]
process timing ----- overall results -----
    run time : 2 days, 13 hrs, 13 min, 13 sec          cycles done : 1
    last new find : 0 days, 0 hrs, 10 min, 37 sec      corpus count : 6737
last saved crash : none seen yet                      saved crashes : 0
last saved hang : 0 days, 1 hrs, 15 min, 58 sec       saved hangs : 46
cycle progress ----- map coverage -----
now processing : 5762*0 (85.5%)                       map density : 7.40% / 23.60%
runs timed out : 0 (0.00%)                            count coverage : 5.53 bits/tuple
stage progress ----- findings in depth -----
now trying : trim 512/512                             favored items : 667 (9.90%)
stage execs : 112/290 (38.62%)                        new edges on : 1292 (19.18%)
total execs : 4.64M                                    total crashes : 0 (0 saved)
exec speed : 5.56/sec (zzzz...)                       total tmouts : 108 (0 saved)
fuzzing strategy yields ----- item geometry -----
bit flips : disabled (default, enable with -D)        levels : 6
byte flips : disabled (default, enable with -D)        pending : 3259
arithmetics : disabled (default, enable with -D)       pend fav : 1
known ints : disabled (default, enable with -D)        own finds : 1235
dictionary : n/a                                         imported : 4815
havoc/splice : 422/552k, 813/2.08M                     stability : 69.98%
py/custom/rq : unused, unused, unused, unused
trim/eff : 8.06%/1.95M, disabled                      [cpu001:100%]
```

Static Analysis

We ran the codebase through the static analysis tool `cppcheck`, which tagged 1569 warnings and errors. One of the common errors flagged by `cppcheck` was `shiftTooManyBits`

390r-debugging-setup\p7zip\CPP\7zip\Archive\7z\7zIn.cpp				
261	shiftTooManyBits	758	error	Shifting 32-bit value by 32 bits is undefined behaviour
1546	shiftTooManyBits	758	error	Shifting 32-bit value by 32 bits is undefined behaviour
1547	shiftTooManyBits	758	error	Shifting 32-bit value by 32 bits is undefined behaviour
1598	shiftTooManyBits	758	error	Shifting 32-bit value by 62 bits is undefined behaviour

Unfortunately, when looking at the actual source code, almost all of these errors come from an innocuous function:

```
#define GetUi32(p) ( \
    ((const Byte *)(p))[0] | \
    ((UInt32)((const Byte *)(p))[1] << 8) | \
    ((UInt32)((const Byte *)(p))[2] << 16) | \
    ((UInt32)((const Byte *)(p))[3] << 24))
```

The rest, on closer inspection, are also falsely flagged as errors, such as this one:

[2594](#) shiftTooManyBits [758](#) error Shifting 32-bit value by 63 bits is undefined behaviour

```
2594 if (node.FileSize ≥ ((UInt64)1 << 63))
2595     return S_FALSE;
```

A more promising error seems to be a possible null pointer exception:

[390r-debugging-setup\p7zip\CPP\7zip\Archive\Zip\ZipHandlerOut.cpp](#)

[41](#) nullPointerRedundantCheck [476](#) warning Either the condition 'password' is redundant or there is possible null pointer dereference: s++.

[41](#) nullPointerArithmeticRedundantCheck [682](#) warning Either the condition 'password' is redundant or there is pointer arithmetic with NULL pointer.

```
37 static bool IsSimpleAsciiString(const wchar_t *s)
38 {
39     for (;;)
40     {
41         wchar_t c = *s++;
42         if (c == 0)
43             return true;
44         if (c < 0x20 || c > 0x7F)
45             return false;
46     }
47 }
```

This function is only called once, in the same file at line 415:


```

392     CMyComPtr<ICryptoGetTextPassword2> getTextPassword;
393     {
394         CMyComPtr<IArchiveUpdateCallback> udateCallBack2(callback);
395         udateCallBack2.QueryInterface(IID_ICryptoGetTextPassword2, &getTextPassword);
396     }
397     CCompressionMethodMode options;
398     (CBaseProps &)options = _props;
399     options._dataSizeReduce = largestSize;
400     options._dataSizeReduceDefined = largestSizeDefined;
401
402     options.PasswordIsDefined = false;
403     options.Password.Wipe_and_Empty();
404     if (getTextPassword)
405     {
406         CMyComBSTR_Wipe password;
407         Int32 passwordIsDefined;
408         RINOK(getTextPassword->CryptoGetTextPassword2(&passwordIsDefined, &password));
409         options.PasswordIsDefined = IntToBool(passwordIsDefined);
410         if (options.PasswordIsDefined)
411         {
412             if (!m_ForceAesMode)
413                 options.IsAesMode = thereAreAesUpdates;
414
415             if (!IsSimpleAsciiString(password))
416                 return E_INVALIDARG;

```

It looks like `password` gets populated in `CryptoGetTexPassword2`, looking at that function, and the subsequent call to `StringToBstr`, it unfortunately looks like the nullpointer is properly checked for.

```

97     STDMETHODIMP CUpdateCallback100Imp::CryptoGetTextPassword2(Int32 *passwordIsDefined, BSTR *password)
98     {
99         *password = NULL;
100         *passwordIsDefined = BoolToInt>PasswordIsDefined);
101         if (!PasswordIsDefined)
102             return S_OK;
103         return StringToBstr>Password, password);
104     }

```

```

77     inline HRESULT StringToBstr(LPCOLESTR src, BSTR *bstr)
78     {
79         *bstr = ::SysAllocString(src);
80         return (*bstr) ? S_OK : E_OUTOFMEMORY;
81     }

```