README.md

Fuzz-libpparam - CVE-2020-28723

Building clang++

Replace atomic for clang++

```
"cstdatomic" . // for GCC
./xlist.hpp:#include <cstdatomic> ===> ./xlist.hpp:#include <atomic>
```

Memory Leak

```
INFO: Seed: 4138993635
INFO: Loaded 1 modules (372 guards): [0x566420, 0x5669f0),
INFO: -max_len is not provided, using 64
INFO: A corpus is not provided, starting from an empty corpus
#0      READ units: 1
<device>eth1</device>
<ipv4>192.168.0.1</ipv4>
<ipv6>3fee::1</ipv6>
<rx_packets>57347</rx_packets>
<tx_packets>48936</tx_packets>
<device>eth1</device>
<ipv4>192.168.0.1</ipv4>
<ipv6>3fee::1</ipv6>
<rx_packets>57347</rx_packets>
<tx_packets>48936</tx_packets>
<device>eth1</device>
<ipv4>192.168.0.1</ipv4>
<ipv6>3fee::1</ipv6>
<rx_packets>57347</rx_packets>
<tx_packets>48936</tx_packets>

=================================================================
==11759==ERROR: LeakSanitizer: detected memory leaks

Direct leak of 104 byte(s) in 1 object(s) allocated from:
    #0 0x4f0ab8 in operator new[](unsigned long) /home/fuzz/codes/libfuzzer/src/llvm/projects/compiler-rt/lib/asan/asan_new_delete.cc
    #1 0x7f6ded5ab8f8 in pparam::IPParam::split(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, char
    #2 0x7f6ded5b1a32 in pparam::IPv6Param::setAddress(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char>
    #3 0x7f6ded5b0faa in pparam::IPv6Param::set(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > consti
    #4 0x7f6ded5b05db in pparam::IPv6Param::operator=(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >
    #5 0x4f6196 in pparam::IPv6Param::operator=(char const*) (/home/fuzz/codes/libfuzzer//nic+0x4f6196)
    #6 0x4f3e3b in hello(int, char**) (/home/fuzz/codes/libfuzzer//nic+0x4f3e3b)
    #7 0x4f64aa in LLVMFuzzerTestOneInput (/home/fuzz/codes/libfuzzer//nic+0x4f64aa)
    #8 0x50abe4 in fuzzer::Fuzzer::ExecuteCallback(unsigned char const*, unsigned long) /home/fuzz/codes/libfuzzer/libFuzzer/Fuzzer/.
    #9 0x50ae0e in fuzzer::Fuzzer::RunOne(unsigned char const*, unsigned long) /home/fuzz/codes/libfuzzer/libFuzzer/Fuzzer/./FuzzerLo
    #10 0x50aa41 in fuzzer::Fuzzer::RunOne(std::vector<unsigned char, std::allocator<unsigned char> > const&) /home/fuzz/codes/libfuz
    #11 0x50aa41 in fuzzer::Fuzzer::ShuffleAndMinimize(std::vector<std::vector<unsigned char, std::allocator<unsigned char> >, std::a
    #12 0x50447f in fuzzer::FuzzerDriver(int*, char***, int (*)(unsigned char const*, unsigned long)) /home/fuzz/codes/libfuzzer/libF
    #13 0x501020 in main /home/fuzz/codes/libfuzzer/libFuzzer/Fuzzer/./FuzzerMain.cpp:20:10
    #14 0x7f6dec50e09a in __libc_start_main /build/glibc-B9XfQf/glibc-2.28/csu/../csu/libc-start.c:308:16

Direct leak of 104 byte(s) in 1 object(s) allocated from:
    #0 0x4f0ab8 in operator new[](unsigned long) /home/fuzz/codes/libfuzzer/src/llvm/projects/compiler-rt/lib/asan/asan_new_delete.cc
    #1 0x7f6ded5ab8f8 in pparam::IPParam::split(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >, char
    #2 0x7f6ded5b1a32 in pparam::IPv6Param::setAddress(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char>
    #3 0x7f6ded5b0faa in pparam::IPv6Param::set(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > consti
    #4 0x7f6ded5b05db in pparam::IPv6Param::operator=(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >
    #5 0x4f6196 in pparam::IPv6Param::operator=(char const*) (/home/fuzz/codes/libfuzzer//nic+0x4f6196)
    #6 0x4f3e3b in hello(int, char**) (/home/fuzz/codes/libfuzzer//nic+0x4f3e3b)
    #7 0x4f64aa in LLVMFuzzerTestOneInput (/home/fuzz/codes/libfuzzer//nic+0x4f64aa)
    #8 0x50abe4 in fuzzer::Fuzzer::ExecuteCallback(unsigned char const*, unsigned long) /home/fuzz/codes/libfuzzer/libFuzzer/Fuzzer/.
    #9 0x50a9fc in fuzzer::Fuzzer::ShuffleAndMinimize(std::vector<std::vector<unsigned char, std::allocator<unsigned char> >, std::al
    #10 0x50447f in fuzzer::FuzzerDriver(int*, char***, int (*)(unsigned char const*, unsigned long)) /home/fuzz/codes/libfuzzer/libF
    #11 0x501020 in main /home/fuzz/codes/libfuzzer/libFuzzer/Fuzzer/./FuzzerMain.cpp:20:10
    #12 0x7f6dec50e09a in __libc_start_main /build/glibc-B9XfQf/glibc-2.28/csu/../csu/libc-start.c:308:16

SUMMARY: AddressSanitizer: 208 byte(s) leaked in 2 allocation(s).

INFO: a leak has been found in the initial corpus.
```

```
INFO: to ignore leaks on libFuzzer side use -detect_leaks=0.

MS: 0 ; base unit: 0000000000000000000000000000000000000000
0xa,
\x0a
artifact_prefix='./'; Test unit written to ./leak-adc83b19e793491b1c6ea0fd8b46cd9f32e592fc
Base64: Cg==
```

```
void IPv6Param::setAddress(const string& iIP) throw (Exception)
{
        //validate
        string allowedIPv6 = "1234567890ABCDEFabcdef:";
        if (validateString(iIP, allowedIPv6)) {
                int partCount = 0;
                string *sparts = split(iIP, ':', partCount); // allocated heap
                //check box count limits

    ... SKIP

                        //fill remained empty boexes with 0
                        for (int i = emptyBox;
                                i < emptyBox + (8 - partCount) + 1; i++)
                                parts[i] = 0;
                        //copy
                        for (int i = 0; i < 8; ++i)
                                address[i] = parts[i];
                }
        } else
                throw Exception("IP is not valid", TracePoint("sparam"));
}
```

We see `string *sparts = split(iIP, ':', partCount);` allocated `*sparts` and never `free`,

Thanks, Ramin

Releases

No releases published

Packages

No packages published

Languages

C++ 96.1% CMake 2.1% Other 1.8%