Talos Vulnerability Report

TALOS-2020-1225

CGAL libcgal multiple code execution vulnerabilities in Nef polygon-parsing code

FEBRUARY 24, 2021

CVE NUMBER

CVE-2020-28601,CVE-2020-28602,CVE-2020-28603,CVE-2020-28604,CVE-2020-28605,CVE-2020-28606,CVE-2020-28607,CVE-2020-28608,CVE-2020-28609,CVE-2020-28610,CVE-2020-28611,CVE-2020-28612,CVE-2020-28613,CVE-2020-28614,CVE-2020-28615,CVE-2020-28616, CVE-2020-28617,CVE-2020-28618,CVE-2020-28619,CVE-2020-28620, CVE-2020-28621,CVE-2020-28622,CVE-2020-28623,CVE-2020-28624,CVE-2020-28625,CVE-2020-28626,CVE-2020-28627,CVE-2020-28628,CVE-2020-28629,CVE-2020-28630,CVE-2020-28631,CVE-2020-28632,CVE-2020-28633,CVE-2020-28634,CVE-2020-28635,CVE-2020-28636,CVE-2020-35628,CVE-2020-35629,CVE-2020-35630, CVE-2020-35631,CVE-2020-35632,CVE-2020-35633,CVE-2020-35634,CVE-2020-35635,CVE-2020-35636

Summary

Multiple code execution vulnerabilities exists in the Nef polygon-parsing functionality of CGAL libcgal CGAL-5.1.1. A specially crafted malformed file can lead to an out-of-bounds read and type confusion, which could lead to code execution. An attacker can provide malicious input to trigger any of these vulnerabilities.

Tested Versions

CGAL Project libcgal CGAL-5.1.1

Product URLs

https://github.com/CGAL/cgal

CVSSv3 Score

10.0 - CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

CWE

CWE-129 - Improper Validation of Array Index

Details

Libcgal is an open-source C++ library that provides geometric algorithms for fast and reliable data processing. It is used in an array of research projects and computational areas, and other open-source projects such as Openscad.

Out of the multitude of shapes CGAL is capable of handling, today we visit the Nef polygon, whose parsing code can be found at `CGAL/include/CGAL/Nef_2` (for 2-dimensional operations), `CGAL/include/CGAL/Nef_3` for (3-dimensional operations), or `CGAL/include/CGAL/Nef_S2` (for 2-dimensional operations on a Nef Polygon bound by a sphere). For the purposes of this advisory, we only discuss Nef_3 specifically, however we will also briefly cover issues within the other objects as well.
To start, an example `.nef3` file:

```
Selective Nef Complex                   // [1]
standard
vertices   8                            // [2]
halfedges  42                           // [3]
facets     18
volumes    2
shalfedges 84
shalfloops 2
sfaces     30
0 { 0 2, 0 5, 0 1, -2 | 0 0 5 1 } 1      // [4]
1 { 3 5, 6 11, 2 3, -2 | 5 0 5 1 } 1
2 { 6 8, 12 17, 4 5, -2 | 5 5 5 1 } 1
3 { 9 11, 18 23, 6 7, -2 | 0 5 0 1 } 1
4 { 12 14, 24 29, 8 9, -2 | 5 0 0 1 } 1
5 { 15 17, 30 35, 10 11, -2 | 5 5 0 1 } 1
6 { 18 20, 36 41, 12 13, -2 | 0 5 5 1 } 1
7 { 21 23, 42 47, 14 15, -2 | 0 0 3 1 } 0
0 { 3, 0, 0 0 | 1 0 0 1 } 1              // [5]
1 { 18, 0, 0 5 | 0 1 0 1 } 1
2 { 21, 0, 0 4 | 0 0 -1 1 } 1
3 { 0, 1, 0 6 | -1 0 0 1 } 1
4 { 6, 1, 0 7 | 0 1 0 1 } 1
5 { 13, 1, 0 10 | 0 0 -1 1 } 1
// [...]
40 { 28, 13, 0 82 | 1 0 0 1 } 0
41 { 30, 13, 0 79 | 0 0 -1 1 } 0
// [...]
```

After the magic bytes [1] we see a set of numbers corresponding to the amount of each given data type. Thus the line at [2] tells us there's 8 vertices, and the line at [3] tells us there's 42 halfedges and so forth. At [4] we see the start of the vertices, 8 entries in all, and at [5] we begin the halfedges. A set of vectors are initialized from these entries as such:

```
// "Nef_3/SNC_io_parser.h"
template <typename EW>
void SNC_io_parser<EW>::read()
{
   //[...]

  for(i=0; i<vn; ++i)  Vertex_of.push_back(this->sncp()->new_vertex_only());
  for(i=0; i<en; ++i)  Edge_of.push_back(this->sncp()->new_halfedge_only());
  for(i=0; i<fn; ++i)  Halffacet_of.push_back(this->sncp()->new_halffacet_only());
  for(i=0; i<cn; ++i)  Volume_of.push_back(this->sncp()->new_volume_only());
  for(i=0; i<sen; ++i) SEdge_of.push_back(this->sncp()->new_shalfedge_only());
  for(i=0; i<sln; ++i) SLoop_of.push_back(this->sncp()->new_shalfloop_only());
  for(i=0; i<sfn; ++i) SFace_of.push_back(this->sncp()->new_sface_only());
  //[...]
```

Let us now examine the parsing code for a given vertices (e.g. `5 { 15 17, 30 35, 10 11, -2 | 5 5 0 1 } 1`):

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_vertex(Vertex_handle vh) {

  bool OK = true;
  int index;
  //[...]

  in >> index;                  // [1]
  OK = OK && test_string("{");  // [2]
  vh->sncp() = this->sncp();

  in >> index;                  // [3]
  vh->svertices_begin() = (index >= 0 ? Edge_of[index] : this->svertices_end()); // [4]
  in >> index;
  vh->svertices_last()  = index >= 0 ? Edge_of[index] : this->svertices_end();
  OK = OK && test_string(",");
  in >> index;
  vh->shalfedges_begin() = index >= 0 ? SEdge_of[index] : this->shalfedges_end();
  in >> index;
  vh->shalfedges_last()  = index >= 0 ? SEdge_of[index] : this->shalfedges_end();
  OK = OK && test_string(",");
  in >> index;
  vh->sfaces_begin() = index >= 0 ? SFace_of[index] : this->sfaces_end();
  in >> index;
  vh->sfaces_last()  = index >= 0 ? SFace_of[index] : this->sfaces_end();
  OK = OK && test_string(",");
  in >> index;
  vh->shalfloop() = index >= 0 ? SLoop_of[index] : this->shalfloops_end();
  OK = OK && test_string("|");
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  in >> hx >> hy >> hz >> hw;
  vh->point() = Point_3(hx,hy,hz,hw);
#else
  vh->point() =
    Geometry_io<typename K::Kernel_tag, Kernel>::template read_point<Kernel, K>(in);
#endif
  OK = OK && test_string("}");
  in >> vh->mark();

  return OK;
}
```

At [1] we see the index of the entry being read into `int index`, and at [2] we see the left bracket being discarded. The first datapoint of our vertices is read in as an integer at [3], and then assuming it's `>= 0`, our `Vertex_handle vh->svertices_begin()` object member is assigned as the `Edge_of[index]` vector index. This pattern continues for every member of our `Vertex_handle` object, for every vertex object that is read in. Let us now examine what happens upon reading a given halfedge (e.g. `0 { 3, 0, 0 0 | 1 0 0 1 } 1`):

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_edge(Halfedge_handle eh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT hx,hy,hz,hw;
#endif
  in >> index;
  OK = OK && test_string("{");

  in >> index;
  eh->twin() = Edge_of[index];
  OK = OK && test_string(",");
  in >> index;
  eh->center_vertex() = Vertex_of[index];
  OK = OK && test_string(",");
  in >> index;
  if(index == 0) {
    in >> index;
    eh->out_sedge() = SEdge_of[index];
  } else {
    in >> index;
    eh->incident_sface() = SFace_of[index];
  }
  OK = OK && test_string("|");
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  in >> hx >> hy >> hz >> hw;
  eh->point() = Sphere_point(hx,hy,hz);
#else
  eh->point() =
    Geometry_io<typename K::Kernel_tag, Kernel>::template read_point<Kernel,K>(in);
#endif
  OK = OK && test_string("}");
  in >> eh->mark();

  return OK;
}
```

Without being repetitive, it suffices to say that the `read_edge` function follows the same exact code pattern, reading in indexes from our file and then assigning object members to vector items whose index we just read; this is the pattern for `read_facet`, `read_volume`, `read_sedge`, `read_sloop`, and `read_sface` as well. Also worth noting about this code pattern: there's no checking on the indexes between reading them and using them as a vector index. Thus, every object member can be assigned arbitrary memory instead of another given object. This quickly becomes a huge issue when we start dereferencing these objects in other parts of the code, resulting in type confusion and code execution.

CVE-2020-28601 - Nef_2/PM_io_parser.h PM_io_parser::read_vertex() Face_of[] OOB read

An oob read vulnerability exists in Nef_2/PM_io_parser.h PM_io_parser::read_vertex() Face_of[] OOB read:

```
template <typename PMDEC>
bool PM_io_parser<PMDEC>::read_vertex(Vertex_handle v)
{
  // precondition: nodes exist
  // syntax: index { mark, point, isolated object }
  int n; bool iso; int f; Mark m; Point p;
  if ( !(in >> n) ||
       !check_sep("{") ||
       !(in >> iso) ||
       !(in >> f) ||
       !check_sep(",") ||
       !(in >> m) ||
       !check_sep(",") ||
       !(in >> p) ||
       !check_sep("}") ) return false;

  if (iso) v->set_face(Face_of[f]);                // <--- oob read into `Face_of`
```

Crash Information

```
AddressSanitizer:DEADLYSIGNAL
=================================================================
==3292887==ERROR: AddressSanitizer: SEGV on unknown address 0x61900001b4c0 (pc 0x7f6eccdfed82 bp 0x7ffd85dbaef0 sp 0x7ffd85dba6a8 T0)
==3292887==The signal is caused by a READ memory access.
    #0 0x7f6eccdfed82  /build/glibc-ZN95T4/glibc-2.31/string/../sysdeps/x86_64/multiarch/memmove-vec-unaligned-erms.S:312
    #1 0x52f6d7 in __asan_memcpy (//boop/assorted_fuzzing/openscad/openscad-openscad-2020.12-RC2/nef3_fuzzdir/nef3_harness.bin+0x52f6d7)
    #2 0x7f6ed0202370 in bool CGAL::SNC_io_parser<CGAL::SNC_structure<CGAL::Cartesian<CGAL::Gmpq>, CGAL::SNC_indexed_items, bool>
>::read_sedge<CGAL::Cartesian<CGAL::Gmpq> >
(CGAL::internal::In_place_list_iterator<CGAL::SNC_in_place_list_shalfedge<CGAL::SNC_indexed_items::SHalfedge<CGAL::SNC_structure<CGAL::Carte
sian<CGAL::Gmpq>, CGAL::SNC_indexed_items, bool> > >,
std::allocator<CGAL::SNC_in_place_list_shalfedge<CGAL::SNC_indexed_items::SHalfedge<CGAL::SNC_structure<CGAL::Cartesian<CGAL::Gmpq>,
CGAL::SNC_indexed_items, bool> > > > >) /usr/local/include/CGAL/Nef_3/SNC_io_parser.h:1787:16
    #3 0x7f6ed01ea16c in void CGAL::SNC_io_parser<CGAL::SNC_structure<CGAL::Cartesian<CGAL::Gmpq>, CGAL::SNC_indexed_items, bool>
>::read_items<CGAL::Cartesian<CGAL::Gmpq> >(int) /usr/local/include/CGAL/Nef_3/SNC_io_parser.h:1469:10
    #4 0x7f6ed01e7e60 in CGAL::SNC_io_parser<CGAL::SNC_structure<CGAL::Cartesian<CGAL::Gmpq>, CGAL::SNC_indexed_items, bool> >::read()
/usr/local/include/CGAL/Nef_3/SNC_io_parser.h:1437:5
    #5 0x7f6ed01e2c91 in std::istream& CGAL::operator>><CGAL::Cartesian<CGAL::Gmpq>, CGAL::SNC_indexed_items, bool>(std::istream&,
CGAL::Nef_polyhedron_3<CGAL::Cartesian<CGAL::Gmpq>, CGAL::SNC_indexed_items, bool>&)
/usr/local/include/CGAL/IO/Nef_polyhedron_iostream_3.h:44:5
    #6 0x7f6ed01e22cc in import_nef3(std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> > const&, Location
const&) //boop/assorted_fuzzing/openscad/openscad-openscad-2020.12-RC2/openscad-openscad-2020.12-RC2/src/import_nef.cc:29:5
    #7 0x5648cc in LLVMFuzzerTestOneInput //boop/assorted_fuzzing/openscad/openscad-openscad-2020.12-RC2/./fuzz_nef3_harness.cpp:71:21
    #8 0x46a631 in fuzzer::Fuzzer::ExecuteCallback(unsigned char const*, unsigned long) (//boop/assorted_fuzzing/openscad/openscad-openscad-
2020.12-RC2/nef3_fuzzdir/nef3_harness.bin+0x46a631)
    #9 0x455da2 in fuzzer::RunOneTest(fuzzer::Fuzzer*, char const*, unsigned long) (//boop/assorted_fuzzing/openscad/openscad-openscad-
2020.12-RC2/nef3_fuzzdir/nef3_harness.bin+0x455da2)
    #10 0x45b856 in fuzzer::FuzzerDriver(int*, char***, int (*)(unsigned char const*, unsigned long))
(//boop/assorted_fuzzing/openscad/openscad-openscad-2020.12-RC2/nef3_fuzzdir/nef3_harness.bin+0x45b856)
    #11 0x484512 in main (//boop/assorted_fuzzing/openscad/openscad-openscad-2020.12-RC2/nef3_fuzzdir/nef3_harness.bin+0x484512)
    #12 0x7f6eccd670b2 in __libc_start_main /build/glibc-ZN95T4/glibc-2.31/csu/../csu/libc-start.c:308:16
    #13 0x43046d in _start (//boop/assorted_fuzzing/openscad/openscad-openscad-2020.12-RC2/nef3_fuzzdir/nef3_harness.bin+0x43046d)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV /build/glibc-ZN95T4/glibc-2.31/string/../sysdeps/x86_64/multiarch/memmove-vec-unaligned-erms.S:312
==3292887==ABORTING
```

CVE-2020-28602 - Nef_2/PM_io_parser.h PM_io_parser::read_vertex() Halfedge_of[] OOB read

An oob read vulnerability exists in Nef_2/PM_io_parser.h PM_io_parser::read_vertex() Halfedge_of[]:

```
template <typename PMDEC>
bool PM_io_parser<PMDEC>::read_vertex(Vertex_handle v)
{
  // precondition: nodes exist
  // syntax: index { mark, point, isolated object }
  int n; bool iso; int f; Mark m; Point p;
  if ( !(in >> n) ||
       !check_sep("{") ||
       !(in >> iso) ||
       !(in >> f) ||
       !check_sep(",") ||
       !(in >> m) ||
       !check_sep(",") ||
       !(in >> p) ||
       !check_sep("}") ) return false;

  if (iso) v->set_face(Face_of[f]);
  else     v->set_halfedge(Halfedge_of[f]);  // <--- oob read into `Halfedge_of`
```

CVE-2020-28603 - Nef_2/PM_io_parser.h PM_io_parser::read_hedge() e->set_prev() OOB read

An oob read vulnerability exists in Nef_2/PM_io_parser.h PM_io_parser::read_hedge() e->set_prev():

```
template <typename PMDEC>
bool PM_io_parser<PMDEC>::read_hedge(Halfedge_handle e)
{ // syntax: index { opposite, prev, next, vertex, face, mark }
  int n, eo, epr, ene, v, f; bool m;
  if ( !(in >> n) ||
       !check_sep("{") ||
       !(in >> eo) || !check_sep(",") ||
       !(in >> epr) || !check_sep(",") ||
       !(in >> ene) || !check_sep(",") ||
       !(in >> v) || !check_sep(",") ||
       !(in >> f) || !check_sep(",") ||
       !(in >> m) || !check_sep("}") )
    return false;
  CGAL_assertion_msg
     (eo >= 0 || (std::size_t) eo < en || epr >= 0 || (std::size_t) epr < en || ene >= 0 || (std::size_t) ene < en ||
      v >= 0 || (std::size_t) v < vn || f >= 0 || (std::size_t) f < fn ,
      "wrong index in read_hedge");  // assertion does not stop oobs since it's or'ed

  // precond: objects exist!
  CGAL_assertion(EI[e->opposite()]);
  e->set_prev(Halfedge_of[epr]);  // <- oob read
```

CVE-2020-28604 - Nef_2/PM_io_parser.h PM_io_parser::read_hedge() e->set_next() OOB read

An oob read vulnerability exists in Nef_2/PM_io_parser.h PM_io_parser::read_hedge() e->set_next():

```
template <typename PMDEC>
bool PM_io_parser<PMDEC>::read_hedge(Halfedge_handle e)
{ // syntax: index { opposite, prev, next, vertex, face, mark }
  int n, eo, epr, ene, v, f; bool m;
  if ( !(in >> n) ||
       !check_sep("{") ||
       !(in >> eo) || !check_sep(",") ||
       !(in >> epr) || !check_sep(",") ||
       !(in >> ene) || !check_sep(",") ||
       !(in >> v) || !check_sep(",") ||
       !(in >> f) || !check_sep(",") ||
       !(in >> m) || !check_sep("}") )
    return false;
  CGAL_assertion_msg
     (eo >= 0 || (std::size_t) eo < en || epr >= 0 || (std::size_t) epr < en || ene >= 0 || (std::size_t) ene < en ||
      v >= 0 || (std::size_t) v < vn || f >= 0 || (std::size_t) f < fn ,
      "wrong index in read_hedge");  // assertion does not stop oobs since it's or'ed

  // precond: objects exist!
  CGAL_assertion(EI[e->opposite()]);
  e->set_prev(Halfedge_of[epr]);
  e->set_next(Halfedge_of[ene]);  // <- oob read
```

CVE-2020-28605 - Nef_2/PM_io_parser.h PM_io_parser::read_hedge() e->set_vertex() OOB read

An oob read exists in Nef_2/PM_io_parser.h PM_io_parser::read_hedge() e->set_vertex():

```
template <typename PMDEC>
bool PM_io_parser<PMDEC>::read_hedge(Halfedge_handle e)
{ // syntax: index { opposite, prev, next, vertex, face, mark }
  int n, eo, epr, ene, v, f; bool m;
  if ( !(in >> n) ||
       !check_sep("{") ||
       !(in >> eo) || !check_sep(",") ||
       !(in >> epr) || !check_sep(",") ||
       !(in >> ene) || !check_sep(",") ||
       !(in >> v) || !check_sep(",") ||
       !(in >> f) || !check_sep(",") ||
       !(in >> m) || !check_sep("}") )
    return false;
  CGAL_assertion_msg
     (eo >= 0 || (std::size_t) eo < en || epr >= 0 || (std::size_t) epr < en || ene >= 0 || (std::size_t) ene < en ||
      v >= 0 || (std::size_t) v < vn || f >= 0 || (std::size_t) f < fn ,
      "wrong index in read_hedge");  // assertion does not stop oobs since it's or'ed

  // precond: objects exist!
  CGAL_assertion(EI[e->opposite()]);
  e->set_prev(Halfedge_of[epr]);
  e->set_next(Halfedge_of[ene]);
  e->set_vertex(Vertex_of[v]);  // <- oob read
```

CVE-2020-28606 - Nef_2/PM_io_parser.h PM_io_parser::read_hedge() e->set_face() OOB read

An oob read vulnerability exists in Nef_2/PM_io_parser.h PM_io_parser::read_hedge() e->set_face():

```
template <typename PMDEC>
bool PM_io_parser<PMDEC>::read_hedge(Halfedge_handle e)
{ // syntax: index { opposite, prev, next, vertex, face, mark }
  int n, eo, epr, ene, v, f; bool m;
  if ( !(in >> n) ||
       !check_sep("{") ||
       !(in >> eo) || !check_sep(",") ||
       !(in >> epr) || !check_sep(",") ||
       !(in >> ene) || !check_sep(",") ||
       !(in >> v) || !check_sep(",") ||
       !(in >> f) || !check_sep(",") ||
       !(in >> m) || !check_sep("}") )
    return false;
  CGAL_assertion_msg
     (eo >= 0 || (std::size_t) eo < en || epr >= 0 || (std::size_t) epr < en || ene >= 0 || (std::size_t) ene < en ||
      v >= 0 || (std::size_t) v < vn || f >= 0 || (std::size_t) f < fn ,
      "wrong index in read_hedge");  // assertion does not stop oobs since it's or'ed

  // precond: objects exist!
  CGAL_assertion(EI[e->opposite()]);
  e->set_prev(Halfedge_of[epr]);
  e->set_next(Halfedge_of[ene]);
  e->set_vertex(Vertex_of[v]);
  e->set_face(Face_of[f]); // <- oob read
```

CVE-2020-28607 - Nef_2/PM_io_parser.h PM_io_parser::read_face() set_halfedge() OOB read

An oob read vulnerability exists in Nef_2/PM_io_parser.h PM_io_parser::read_face() set_halfedge():

```
template <typename PMDEC>
bool PM_io_parser<PMDEC>::read_face(Face_handle f)
{ // syntax: index { halfedge, fclist, ivlist, mark }
  int n, ei, vi; Mark m;
  if ( !(in >> n) || !check_sep("{") ) return false;
  if ( !(in >> ei) || !check_sep(",") ) return false;
  if (ei >= 0) f->set_halfedge(Halfedge_of[ei]); // <- oob read
```

CVE-2020-28608 - Nef_2/PM_io_parser.h PM_io_parser::read_face() store_fc() OOB read

An oob read vulnerability exists in Nef_2/PM_io_parser.h PM_io_parser::read_face() store_fc():

```
template <typename PMDEC>
bool PM_io_parser<PMDEC>::read_face(Face_handle f)
{ // syntax: index { halfedge, fclist, ivlist, mark }
  int n, ei, vi; Mark m;
  if ( !(in >> n) || !check_sep("{") ) return false;
  if ( !(in >> ei) || !check_sep(",") ) return false;
  if (ei >= 0) f->set_halfedge(Halfedge_of[ei]);
  while (in >> ei) {
    CGAL_assertion_msg(ei >= 0 && (std::size_t) ei < en, "wrong index in face cycle list.");
    f->store_fc(Halfedge_of[ei]);   // <- oob read
  } in.clear();
```

CVE-2020-28609 - Nef_2/PM_io_parser.h PM_io_parser::read_face() store_iv() OOB read

An oob read vulnerability exists in Nef_2/PM_io_parser.h PM_io_parser::read_face() store_iv():

```
template <typename PMDEC>
bool PM_io_parser<PMDEC>::read_face(Face_handle f)
{ // syntax: index { halfedge, fclist, ivlist, mark }
  int n, ei, vi; Mark m;
  if ( !(in >> n) || !check_sep("{") ) return false;
  if ( !(in >> ei) || !check_sep(",") ) return false;
  if (ei >= 0) f->set_halfedge(Halfedge_of[ei]);
  while (in >> ei) {
    CGAL_assertion_msg(ei >= 0 && (std::size_t) ei < en, "wrong index in face cycle list.");
    f->store_fc(Halfedge_of[ei]);
  } in.clear();
  if (!check_sep(",")) { return false; }
  while (in >> vi) {
    CGAL_assertion_msg(vi >= 0 && (std::size_t) vi < vn, "wrong index in iso vertex list.");
    f->store_iv(Vertex_of[vi]);
  } in.clear();
  if (!check_sep(",") || !(in >> m) || !check_sep("}") )
    return false;
  mark(f) = m;  // <- oob read
  return true;
}
```

CVE-2020-28610 - Nef_S2/SM_io_parser.h SM_io_parser::read_vertex() set_face() OOB read

An oob read vulnerability exists in Nef_S2/SM_io_parser.h SM_io_parser::read_vertex() set_face():

```
template <typename Decorator_>
bool SM_io_parser<Decorator_>::read_vertex(SVertex_handle v)
{
  // precondition: nodes exist
  // syntax: index { isolated incident_object, mark, point}
  int n; bool iso; int f; Mark m; Sphere_point p;
  if ( !(in >> n) ||
       !check_sep("{") ||
       !(in >> iso) ||
       !(in >> f) ||
       !check_sep(",") ||
       !(in >> m) ||
       !check_sep(",") ||
       !(in >> p) ||
       !check_sep("}") ) return false;

  if (iso) set_face(v,SFace_of[f]);   // <- oob read
  else     set_first_out_edge(v,Edge_of[f]);
  v->mark() = m; v->point() = p;
  return true;
}
```

CVE-2020-28611 - Nef_S2/SM_io_parser.h SM_io_parser::read_vertex() set_first_out_edge() OOB read

An oob read vulnerability exists in Nef_S2/SM_io_parser.h SM_io_parser::read_vertex() set_first_out_edge():

```
template <typename Decorator_>
bool SM_io_parser<Decorator_>::read_vertex(SVertex_handle v)
{
  // precondition: nodes exist
  // syntax: index { isolated incident_object, mark, point}
  int n; bool iso; int f; Mark m; Sphere_point p;
  if ( !(in >> n) ||
       !check_sep("{") ||
       !(in >> iso) ||
       !(in >> f) ||
       !check_sep(",") ||
       !(in >> m) ||
       !check_sep(",") ||
       !(in >> p) ||
       !check_sep("}") ) return false;

  if (iso) set_face(v,SFace_of[f]);
  else     set_first_out_edge(v,Edge_of[f]);   // <- oob read
  v->mark() = m; v->point() = p;
  return true;
}
```

CVE-2020-28612 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->svertices_begin() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->svertices_begin():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_vertex(Vertex_handle vh) {

  bool OK = true;
  int index;
  #ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT hx, hy, hz, hw;
  #endif

  in >> index;
  OK = OK && test_string("{");
  vh->sncp() = this->sncp();

  in >> index;
  vh->svertices_begin() = (index >= 0 ? Edge_of[index] : this->svertices_end());  // <- oob read here
```

CVE-2020-28613 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->svertices_last() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->svertices_last():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_vertex(Vertex_handle vh) {

  bool OK = true;
  int index;
  #ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT hx, hy, hz, hw;
  #endif

  in >> index;
  OK = OK && test_string("{");
  vh->sncp() = this->sncp();

  in >> index;
  vh->svertices_begin() = (index >= 0 ? Edge_of[index] : this->svertices_end());
  in >> index;
  vh->svertices_last()  = index >= 0 ? Edge_of[index] : this->svertices_end();  // <- oob read here
```

CVE-2020-28614 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->shalfedges_begin() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->shalfedges_begin():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_vertex(Vertex_handle vh) {

  bool OK = true;
  int index;
  #ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT hx, hy, hz, hw;
  #endif

  in >> index;
  OK = OK && test_string("{");
  vh->sncp() = this->sncp();

  in >> index;
  vh->svertices_begin() = (index >= 0 ? Edge_of[index] : this->svertices_end());
  in >> index;
  vh->svertices_last()  = index >= 0 ? Edge_of[index] : this->svertices_end();
  OK = OK && test_string(",");
  in >> index;
  vh->shalfedges_begin() = index >= 0 ? SEdge_of[index] : this->shalfedges_end(); // <- oob read here
```

CVE-2020-28615 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->shalfedges_last() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->shalfedges_last():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_vertex(Vertex_handle vh) {

  bool OK = true;
  int index;
  #ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT hx, hy, hz, hw;
  #endif

  in >> index;
  OK = OK && test_string("{");
  vh->sncp() = this->sncp();

  in >> index;
  vh->svertices_begin() = (index >= 0 ? Edge_of[index] : this->svertices_end());
  in >> index;
  vh->svertices_last()  = index >= 0 ? Edge_of[index] : this->svertices_end();
  OK = OK && test_string(",");
  in >> index;
  vh->shalfedges_begin() = index >= 0 ? SEdge_of[index] : this->shalfedges_end();
  in >> index;
  vh->shalfedges_last()  = index >= 0 ? SEdge_of[index] : this->shalfedges_end(); // <- oob read here
```

CVE-2020-28616 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->sfaces_begin() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->sfaces_begin():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_vertex(Vertex_handle vh) {

  bool OK = true;
  int index;
  #ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT hx, hy, hz, hw;
  #endif

  in >> index;
  OK = OK && test_string("{");
  vh->sncp() = this->sncp();

  in >> index;
  vh->svertices_begin() = (index >= 0 ? Edge_of[index] : this->svertices_end());
  in >> index;
  vh->svertices_last()  = index >= 0 ? Edge_of[index] : this->svertices_end();
  OK = OK && test_string(",");
  in >> index;
  vh->shalfedges_begin() = index >= 0 ? SEdge_of[index] : this->shalfedges_end();
  in >> index;
  vh->shalfedges_last()  = index >= 0 ? SEdge_of[index] : this->shalfedges_end();
  OK = OK && test_string(",");
  in >> index;
  vh->sfaces_begin() = index >= 0 ? SFace_of[index] : this->sfaces_end(); // <- oob read here
```

CVE-2020-28617 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->sfaces_last() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->sfaces_last():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_vertex(Vertex_handle vh) {

  bool OK = true;
  int index;
  #ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT hx, hy, hz, hw;
  #endif

  in >> index;
  OK = OK && test_string("{");
  vh->sncp() = this->sncp();

  in >> index;
  vh->svertices_begin() = (index >= 0 ? Edge_of[index] : this->svertices_end());
  in >> index;
  vh->svertices_last()  = index >= 0 ? Edge_of[index] : this->svertices_end();
  OK = OK && test_string(",");
  in >> index;
  vh->shalfedges_begin() = index >= 0 ? SEdge_of[index] : this->shalfedges_end();
  in >> index;
  vh->shalfedges_last()  = index >= 0 ? SEdge_of[index] : this->shalfedges_end();
  OK = OK && test_string(",");
  in >> index;
  vh->sfaces_begin() = index >= 0 ? SFace_of[index] : this->sfaces_end();
  in >> index;
  vh->sfaces_last()  = index >= 0 ? SFace_of[index] : this->sfaces_end();  // <- oob read here
```

CVE-2020-28618 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->shalfloop() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_vertex() vh->shalfloop():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_vertex(Vertex_handle vh) {

  bool OK = true;
  int index;
  #ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT hx, hy, hz, hw;
  #endif

  in >> index;
  OK = OK && test_string("{");
  vh->sncp() = this->sncp();

  in >> index;
  vh->svertices_begin() = (index >= 0 ? Edge_of[index] : this->svertices_end());
  in >> index;
  vh->svertices_last()  = index >= 0 ? Edge_of[index] : this->svertices_end();
  OK = OK && test_string(",");
  in >> index;
  vh->shalfedges_begin() = index >= 0 ? SEdge_of[index] : this->shalfedges_end();
  in >> index;
  vh->shalfedges_last()  = index >= 0 ? SEdge_of[index] : this->shalfedges_end();
  OK = OK && test_string(",");
  in >> index;
  vh->sfaces_begin() = index >= 0 ? SFace_of[index] : this->sfaces_end();
  in >> index;
  vh->sfaces_last()  = index >= 0 ? SFace_of[index] : this->sfaces_end();
  OK = OK && test_string(",");
  in >> index;
  vh->shalfloop() = index >= 0 ? SLoop_of[index] : this->shalfloops_end(); // <- oob read here
```

CVE-2020-28619 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_edge() eh->twin() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_edge() eh->twin():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_edge(Halfedge_handle eh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT hx,hy,hz,hw;
#endif
  in >> index;
  OK = OK && test_string("{");

  in >> index;
  eh->twin() = Edge_of[index]; // <- oob read here
```

CVE-2020-28620 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_edge() eh->center_vertex() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_edge() eh->center_vertex():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_edge(Halfedge_handle eh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT hx,hy,hz,hw;
#endif
  in >> index;
  OK = OK && test_string("{");

  in >> index;
  eh->twin() = Edge_of[index];
  OK = OK && test_string(",");
  in >> index;
  eh->center_vertex() = Vertex_of[index]; // <- oob read here
```

CVE-2020-28621 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_edge() eh->out_sedge() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_edge() eh->out_sedge():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_edge(Halfedge_handle eh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT hx,hy,hz,hw;
#endif
  in >> index;
  OK = OK && test_string("{");

  in >> index;
  eh->twin() = Edge_of[index];
  OK = OK && test_string(",");
  in >> index;
  eh->center_vertex() = Vertex_of[index];
  OK = OK && test_string(",");
  in >> index;
  if(index == 0) {
    in >> index;
    eh->out_sedge() = SEdge_of[index]; // <- oob read here
```

CVE-2020-28622 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_edge() eh->incident_sface() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_edge() eh->incident_sface():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_edge(Halfedge_handle eh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT hx,hy,hz,hw;
#endif
  in >> index;
  OK = OK && test_string("{");

  in >> index;
  eh->twin() = Edge_of[index];
  OK = OK && test_string(",");
  in >> index;
  eh->center_vertex() = Vertex_of[index];
  OK = OK && test_string(",");
  in >> index;
  if(index == 0) {
    in >> index;
    eh->out_sedge() = SEdge_of[index];
  } else {
    in >> index;
    eh->incident_sface() = SFace_of[index]; // <- oob read here
  }
```

CVE-2020-28623 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_facet() fh->twin() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_facet() fh->twin():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_facet(Halffacet_handle fh) {

  bool OK = true;
  int index;
  char cc;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  fh->twin() = Halffacet_of[index]; // <- oob read here
```

CVE-2020-28624 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_facet() fh->boundary_entry_objects SEdge_of OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_facet() fh->boundary_entry_objects SEdge_of:

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_facet(Halffacet_handle fh) {

  bool OK = true;
  int index;
  char cc;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  fh->twin() = Halffacet_of[index];
  OK = OK && test_string(",");

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    fh->boundary_entry_objects().push_back(make_object(SEdge_of[index])); // <- oob read here
    in >> cc;
  }
```

CVE-2020-28625 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_facet() fh->boundary_entry_objects SLoop_of OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_facet() fh->boundary_entry_objects SLoop_of:

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_facet(Halffacet_handle fh) {

  bool OK = true;
  int index;
  char cc;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  fh->twin() = Halffacet_of[index];
  OK = OK && test_string(",");

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    fh->boundary_entry_objects().push_back(make_object(SEdge_of[index]));
    in >> cc;
  }

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    fh->boundary_entry_objects().push_back(make_object(SLoop_of[index])); // <- oob read here
    in >> cc;
  }
```

CVE-2020-28626 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_facet() fh->incident_volume() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_facet() fh->incident_volume():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_facet(Halffacet_handle fh) {

  bool OK = true;
  int index;
  char cc;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  fh->twin() = Halffacet_of[index];
  OK = OK && test_string(",");

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    fh->boundary_entry_objects().push_back(make_object(SEdge_of[index]));
    in >> cc;
  }

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    fh->boundary_entry_objects().push_back(make_object(SLoop_of[index]));
    in >> cc;
  }

  in >> index;
  fh->incident_volume() = Volume_of[index+addInfiBox]; // <- oob read
```

CVE-2020-28627 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_volume() ch->shell_entry_objects() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_volume() ch->shell_entry_objects():

```
template <typename EW>
bool SNC_io_parser<EW>::
read_volume(Volume_handle ch) {

  bool OK = true;
  int index;
  char cc;

  in >> index;
  OK = OK && test_string("{");

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    ch->shell_entry_objects().push_back(make_object(SFace_of[index])); // oob read here
    in >> cc;
  }
```

CVE-2020-28628 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_volume() seh->twin() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_volume() seh->twin():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_sedge(SHalfedge_handle seh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  seh->twin() = SEdge_of[index]; // <- oob read here
```

CVE-2020-28629 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->sprev() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->sprev():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_sedge(SHalfedge_handle seh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  seh->twin() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->sprev() = SEdge_of[index]; // <- oob read here
```

CVE-2020-28630 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->snext() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->snext():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_sedge(SHalfedge_handle seh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  seh->twin() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->sprev() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->snext() = SEdge_of[index]; // <- oob read here
```

CVE-2020-28631 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->source() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->source():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_sedge(SHalfedge_handle seh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  seh->twin() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->sprev() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->snext() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->source() = Edge_of[index]; // <- oob read here
```

CVE-2020-28632 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->incident_sface() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->incident_sface():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_sedge(SHalfedge_handle seh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  seh->twin() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->sprev() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->snext() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->source() = Edge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->incident_sface() = SFace_of[index]; // <- oob read here
```

CVE-2020-28633 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->prev() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->prev():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_sedge(SHalfedge_handle seh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  seh->twin() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->sprev() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->snext() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->source() = Edge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->incident_sface() = SFace_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->prev() = SEdge_of[index]; // <- oob read here
```

CVE-2020-28634 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->next() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->next():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_sedge(SHalfedge_handle seh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  seh->twin() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->sprev() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->snext() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->source() = Edge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->incident_sface() = SFace_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->prev() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->next() = SEdge_of[index]; // <- oob read here
```

CVE-2020-28635 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->facet() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sedge() seh->facet():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_sedge(SHalfedge_handle seh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  seh->twin() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->sprev() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->snext() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->source() = Edge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->incident_sface() = SFace_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->prev() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->next() = SEdge_of[index];
  OK = OK && test_string(",");
  in >> index;
  seh->facet() = Halffacet_of[index]; // <- oob read here
```

CVE-2020-28636 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sloop() slh->twin() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sloop() slh->twin():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_sloop(SHalfloop_handle slh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  slh->twin() = SLoop_of[index]; // <- oob read here
```

CVE-2020-35628 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sloop() slh->incident_sface() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sloop() slh->incident_sface():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_sloop(SHalfloop_handle slh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  slh->twin() = SLoop_of[index];
  OK = OK && test_string(",");
  in >> index;
  slh->incident_sface() = SFace_of[index]; // <- oob read here
```

CVE-2020-35629 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sloop() slh->facet() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sloop() slh->facet():

```
template <typename EW>
template <typename K>
bool SNC_io_parser<EW>::
read_sloop(SHalfloop_handle slh) {

  bool OK = true;
  int index;
#ifdef CGAL_NEF_NATURAL_COORDINATE_INPUT
  typename K::RT a,b,c,d;
#endif

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  slh->twin() = SLoop_of[index];
  OK = OK && test_string(",");
  in >> index;
  slh->incident_sface() = SFace_of[index];
  OK = OK && test_string(",");
  in >> index;
  slh->facet() = Halffacet_of[index]; // <- oob read here
```

CVE-2020-35630 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() sfh->center_vertex() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() sfh->center_vertex():

```
template <typename EW>
bool SNC_io_parser<EW>::
read_sface(SFace_handle sfh) {

  bool OK = true;
  int index;
  char cc;

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  sfh->center_vertex() = Vertex_of[index]; // <- oob read here
```

CVE-2020-35631 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() SD.link_as_face_cycle() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() SD.link_as_face_cycle():

```
template <typename EW>
bool SNC_io_parser<EW>::
read_sface(SFace_handle sfh) {

  bool OK = true;
  int index;
  char cc;

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  sfh->center_vertex() = Vertex_of[index];
  OK = OK && test_string(",");

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    //    sfh->boundary_entry_objects().push_back(SEdge_of[index]);
    SM_decorator SD(&*sfh->center_vertex());
    SD.link_as_face_cycle(SEdge_of[index],sfh); // <- oob read here
    in >> cc;
  }
```

CVE-2020-35632 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() sfh->boundary_entry_objects Edge_of OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() sfh->boundary_entry_objects Edge_of:

```
template <typename EW>
bool SNC_io_parser<EW>::
read_sface(SFace_handle sfh) {

  bool OK = true;
  int index;
  char cc;

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  sfh->center_vertex() = Vertex_of[index];
  OK = OK && test_string(",");

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    //    sfh->boundary_entry_objects().push_back(SEdge_of[index]);
    SM_decorator SD(&*sfh->center_vertex());
    SD.link_as_face_cycle(SEdge_of[index],sfh);
    in >> cc;
  }

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    sfh->boundary_entry_objects().push_back(make_object(Edge_of[index])); // <- oob read here
    this->sncp()->store_sm_boundary_item(Edge_of[index], --(sfh->sface_cycles_end()));
    in >> cc;
  }
```

CVE-2020-35633 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() store_sm_boundary_item() Edge_of OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() store_sm_boundary_item() Edge_of:

```
template <typename EW>
bool SNC_io_parser<EW>::
read_sface(SFace_handle sfh) {

  bool OK = true;
  int index;
  char cc;

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  sfh->center_vertex() = Vertex_of[index];
  OK = OK && test_string(",");

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    //    sfh->boundary_entry_objects().push_back(SEdge_of[index]);
    SM_decorator SD(&*sfh->center_vertex());
    SD.link_as_face_cycle(SEdge_of[index],sfh);
    in >> cc;
  }

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    sfh->boundary_entry_objects().push_back(make_object(Edge_of[index]));
    this->sncp()->store_sm_boundary_item(Edge_of[index], --(sfh->sface_cycles_end())); // <- oob read here
    in >> cc;
  }
```

CVE-2020-35634 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() sfh->boundary_entry_objects Sloop_of OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() sfh->boundary_entry_objects Sloop_of:

```
template <typename EW>
bool SNC_io_parser<EW>::
read_sface(SFace_handle sfh) {

  bool OK = true;
  int index;
  char cc;

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  sfh->center_vertex() = Vertex_of[index];
  OK = OK && test_string(",");

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    //    sfh->boundary_entry_objects().push_back(SEdge_of[index]);
    SM_decorator SD(&*sfh->center_vertex());
    SD.link_as_face_cycle(SEdge_of[index],sfh);
    in >> cc;
  }

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    sfh->boundary_entry_objects().push_back(make_object(Edge_of[index]));
    this->sncp()->store_sm_boundary_item(Edge_of[index], --(sfh->sface_cycles_end()));
    in >> cc;
  }

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    sfh->boundary_entry_objects().push_back(make_object(SLoop_of[index])); // <- oob read here
```

CVE-2020-35635 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() store_sm_boundary_item() Sloop_of OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() store_sm_boundary_item() Sloop_of:

```
template <typename EW>
bool SNC_io_parser<EW>::
read_sface(SFace_handle sfh) {

  bool OK = true;
  int index;
  char cc;

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  sfh->center_vertex() = Vertex_of[index];
  OK = OK && test_string(",");

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    //    sfh->boundary_entry_objects().push_back(SEdge_of[index]);
    SM_decorator SD(&*sfh->center_vertex());
    SD.link_as_face_cycle(SEdge_of[index],sfh);
    in >> cc;
  }

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    sfh->boundary_entry_objects().push_back(make_object(Edge_of[index]));
    this->sncp()->store_sm_boundary_item(Edge_of[index], --(sfh->sface_cycles_end()));
    in >> cc;
  }

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    sfh->boundary_entry_objects().push_back(make_object(SLoop_of[index]));
    this->sncp()->store_sm_boundary_item(SLoop_of[index], --(sfh->sface_cycles_end())); // <- oob read here
```

CVE-2020-35636 - Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() sfh->volume() OOB read

An oob read vulnerability exists in Nef_S2/SNC_io_parser.h SNC_io_parser::read_sface() sfh->volume():

```
template <typename EW>
bool SNC_io_parser<EW>::
read_sface(SFace_handle sfh) {

  bool OK = true;
  int index;
  char cc;

  in >> index;
  OK = OK && test_string("{");

  in >> index;
  sfh->center_vertex() = Vertex_of[index];
  OK = OK && test_string(",");

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    //    sfh->boundary_entry_objects().push_back(SEdge_of[index]);
    SM_decorator SD(&*sfh->center_vertex());
    SD.link_as_face_cycle(SEdge_of[index],sfh);
    in >> cc;
  }

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    sfh->boundary_entry_objects().push_back(make_object(Edge_of[index]));
    this->sncp()->store_sm_boundary_item(Edge_of[index], --(sfh->sface_cycles_end()));
    in >> cc;
  }

  in >> cc;
  while(isdigit(cc)) {
    in.putback(cc);
    in >> index;
    sfh->boundary_entry_objects().push_back(make_object(SLoop_of[index]));
    this->sncp()->store_sm_boundary_item(SLoop_of[index], --(sfh->sface_cycles_end()));
    in >> cc;
  }

  in >> index;
  sfh->volume() = Volume_of[index+addInfiBox]; // <- oob read here
```

Timeline

2021-01-12 - Vendor Disclosure

2021-02-23 - Vendor Patched

2021-02-24 - Public Release

CREDIT

Discovered by Lilith >_> of Cisco Talos.
