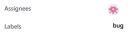
New issue Jump to bottom

## Stack Exhaustion (ecma\_proxy\_object\_get, ecma\_proxy\_object\_set) #3785





```
Inszetei commented on May 23, 2020

JerryScript revision
6 cd 389b

Build platform

Ubuntu 20.04 LTS (Linux 5.4.0-31-generic x86.64)

Build steps

python tools/build.py --profile-es2815-subset --lto-off --compile-flage-g \
--error-messages-on --debug --compile-flage-g --strip-off --logging-on \
--compile-flage-fsanitize-address

Test cases

Var v2 = {};

var v4 = new Proxy(Uint&Array,v2);

v4__proto__ = v4;

v4[1] = 2;

var v3 = new Proxy(parseFloat,v1);

v3__proto_ = v3;

var v6 = "aa" -constructor;

var v7 = parseFloat & v6;
```

akosthekiss commented on May 23, 2020

Member

Could you please try whether you get SO even if you set a limit for the stack? (Note: You can use --stack-limit option when building the engine to limit the maximum amount of stack that the engine can use.)

nszetei commented on May 23, 2020

Author

Could you please try whether you get SO even if you set a limit for the stack? (Note: You can use --stack-limit option when building the engine to limit the maximum amount of stack that the engine can use.)

Yes. Unlike #3783 here I got SO in both cases (e.g. with --stack-limit=10).



Inszetei commented on May 23, 2020

Just to have it grouped together, a PoC for ecma\_proxy\_object\_has:

function main() {
 var v1 = [13.37,13.37,13.37,13.37];
 var v4 = {isExtensible:Infinity};
 var v6 = new Proxy(MeaMenp,v4);
 v6.\_proto\_\_ = v1;
 v1.\_proto\_ = v6;
 with (v1) {
 value0f = 0;
 }
 }
 main();

A ambatyai self-assigned this on May 25, 2020

🟷 🐽 dbatyai added the 🛮 bug label on May 25, 2020

