환경 구축

결과

https://chromium-review.googlesource.com/c/v8/v8/+/6573553

상기 버그에 취약한 환경을 구축하고, PoC 백트레이스를 찍어봄.

1. **취약한 환경 세팅**

```
git clone https://chromium.googlesource.com/chromium/tools/depot_tools


echo 'export PATH="$HOME/depot_tools:$PATH"' >> ~/.bashrc
source ~/.bashrc


cd ~/
fetch v8

git fetch origin main
git checkout 36debe3a9abe6c78eb2e4fc470bf1c0226eb4924^
git diff 36debe3a9abe6c78eb2e4fc470bf1c0226eb4924^ 36debe3a9abe6c78

gclient sync --with_branch_heads
gclient runhooks

vpython3 tools/dev/v8gen.py x64.debug

gn gen out.gn/x64.patched --args='is_debug=true is_component_build=false

autoninja -C out.gn/x64.debug d8
```

2. **Regress 수정 버전 PoC**

```
//gn gen out.gn/x64.debug --args='is_debug=true is_component_build=false
//Flags: --allow-natives-syntax --turbolev

function bar(i) {
        if (i == 5) { ThrowSomething(); }
}

function foo(x) {
        let b = x + 2;
        try {
            for (let i = 0; i < 42; i++) {
            b >>= 0.0;
            bar(i);
        }
  } catch(e) {
return b + 42;
  }
}

function assertEq(ast, exp){

      if(ast == exp) {
            print("1");
      }

      else{
            print("0");
      }

}

%NeverOptimizeFunction(bar);
%PrepareFunctionForOptimization(foo);
assertEq(49, foo(5.3));
assertEq(49, foo(5.3));
```

```
%OptimizeFunctionOnNextCall(foo);
assertEq(49, foo(5.3));
```

3. **디버깅 실행**

```
out.gn/x64.debug/d8 --allow-natives-syntax --turbolev poc.js
```

4. **백트레이스 출력**

```
(gef)bt
```