Security Implications of Error.prototype.stack

Michael Ficarra • 68th Meeting of TC39

Our defensive secret checker

```
const reduceRight = Function.prototype.call.bind([].reduceRight);

const genSecretChecker = ((correct, incorrect) => {
  let secrets = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9];
  return guesses => {
   reduceRight(secrets, (memo, secret, idx) =>
     () => { guesses[idx] === secret ? memo() : incorrect(); }
   , correct)();
  };
}).bind(null);
```

Benign usage

```
const secretChecker = genSecretChecker(
  () => { console.log("pass"); },
  () => { console.log("fail"); }
);

secretChecker([
  0, 1, 2, "wrong", 4,
  5, 6, 7, 8, 9,
]);
```

```
fail
```

Adversary observes stack trace

```
const secretChecker = genSecretChecker(
  () => { console.log("pass"); },
  () => {
    console.log("fail");
    console.log(new Error().stack);
  }
);

secretChecker([
  0, 1, 2, "wrong", 4,
  5, 6, 7, 8, 9,
]);
```

```
fail
Error
    at genSecretChecker  (repl:5:17)
    at repl:5:56
    at repl:5:47
    at repl:5:47
    at repl:5:47
    at repl:6:16
    at repl:1:1
    at Script.runInThisContext  (vm.js:90:20)
```

frames = # leading correct guesses

```
const secretChecker = genSecretChecker(
  () => { console.log("pass"); },
  () => {
    console.log("fail");
    console.log(new Error().stack);
  }
);

secretChecker([
  0, 1, 2, "wrong", 4,
  5, 6, 7, 8, 9,
]);
```

```
fail
Error
    at genSecretChecker  (repl:5:17)
    at repl:5:56
    at repl:5:47
    at repl:5:47
    at repl:5:47
    at repl:6:16
    at repl:1:1
    at Script.runInThisContext  (vm.js:90:20)
```

frames = # leading correct guesses

```
secretChecker([
  0, 1, 2, 3, 4, 5, 6,
  "wrong", 8, 9,
]);
```

```
fail
Error
    at genSecretChecker  (repl:5:17)
    at repl:5:56
    at repl:5:47
    at repl:5:47
    at repl:5:47
    at repl:5:47
    at repl:5:47
    at repl:5:47
    at repl:5:47
    at repl:6:16
    at repl:1:1
    at Script.runInThisContext  (vm.js:90:20)
```

frames = # leading correct guesses

```
secretChecker([
  0, "wrong", 2, 3, 4,
  5, 6, 7, 8, 9,
]);
```

```
fail
Error
    at genSecretChecker  (repl:5:17)
    at repl:5:56
    at repl:5:47
    at repl:6:16
    at repl:1:1
    at Script.runInThisContext  (vm.js:90:20)
```

What can we do about it?

1. Formalise stack frames in the spec
2. Provide API for accessing structured representation of stack trace (System.getStack?)
3. Spec Error.prototype.stack getter in terms of rendering of System.getStack
4. Provide API for eliding frames from the System.getStack output for a given function (Error.censor?)
5. Those frames also don't appear in Error.prototype.stack output

Could also have non-security uses

- program level (as opposed to devtools level) library blackboxing
- more complete virtualisation
- ???

Open questions

1. what is the scope of the censorship?
2. is censorship a privileged API?
3. does censorship open a new side channel?

Error Stacks Proposal

https://github.com/tc39/proposal-error-stacks