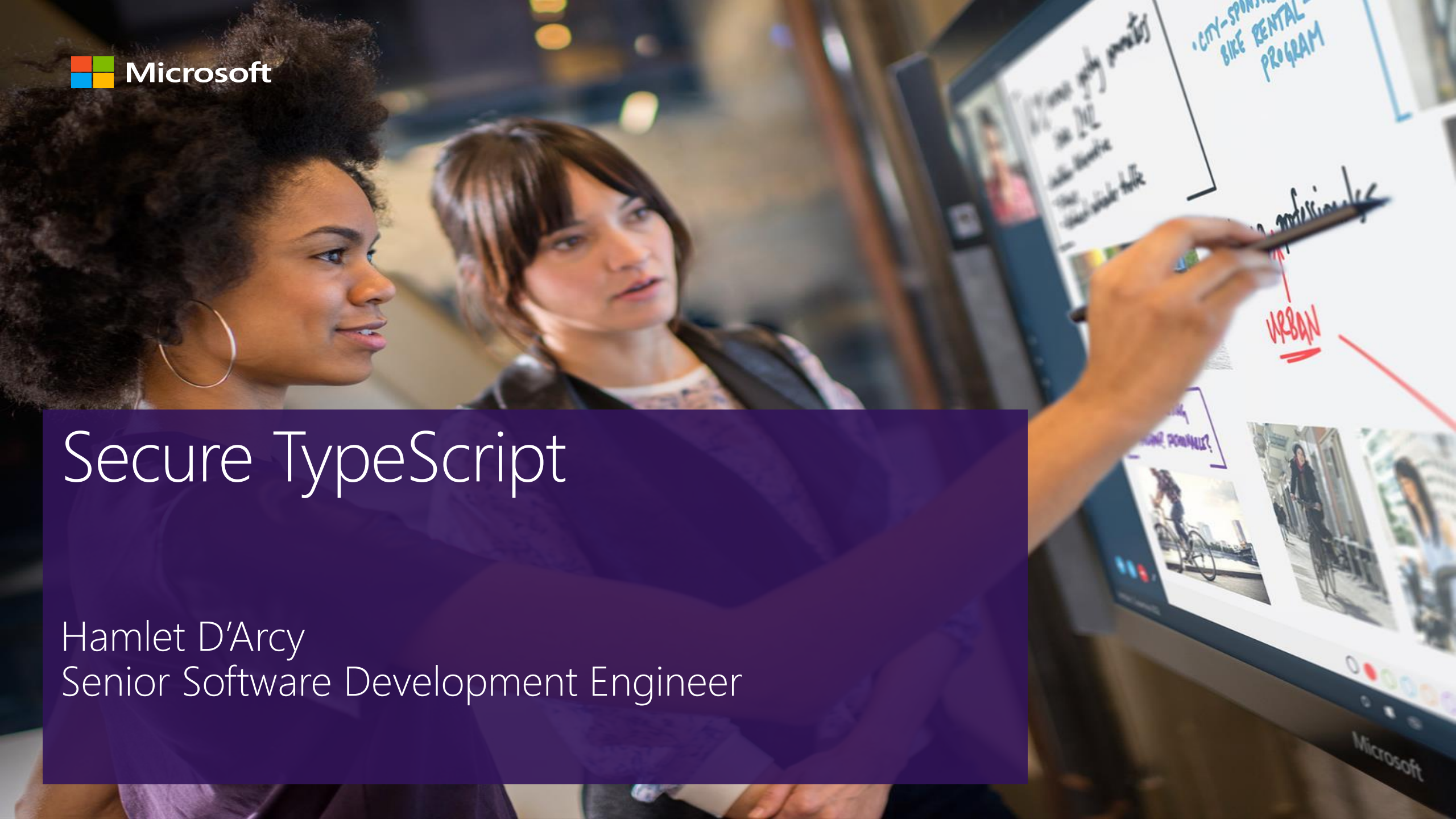


Secure TypeScript

Hamlet D'Arcy
Senior Software Development Engineer

```
export module actions {
    class AsyncAction {

        private delegate;
        constructor(delegate) {
            this.delegate = delegate;
        }

        public invoke(): number {
            return setTimeout(this.delegate, 0);
        }
    }
}
```

```
export module actions {
    class AsyncAction {

        private delegate;
        constructor(delegate) {
            this.delegate = delegate;
        }

        public invoke(): number {
            return setTimeout(this.delegate, 0);
        }
    }
}
```

```
>> src/preso.ts[4, 1]: missing 'use strict'
```

```
export module actions {
    class AsyncAction {

        private delegate;
        constructor(delegate) {
            this.delegate = delegate;
        }

        public invoke(): number {
            return setTimeout(this.delegate, 0);
        }
    }
}
```

```
>> src/preso.ts[4, 1]: missing 'use strict'
```

Microsoft SDL Violation

CWE 398 - Indicator of Poor Code Quality

CWE 710 - Coding Standards Violation

https://cwe.mitre.org/index.html

Common Weakness Enumeration

A Community-Developed Dictionary of Software Weakness Types

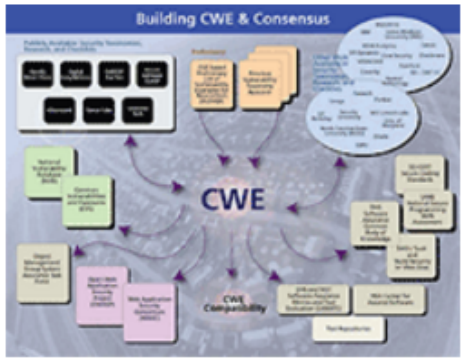
CWE List
- Full Dictionary View
- Development View
- Research View
- Fault Pattern View
- Reports
- Mapping & Navigation

About
- Sources
- Process
- Documents
- FAQs

Community
- Use & Citations
- SwA On-Ramp
- Discussion List
- Discussion Archives
- Contact Us

Scoring
- Prioritization
- CWSS
- CWRAF
- CWE/SANS Top 25

Compatibility
- Requirements
- Coverage Claims Representation
- Compatible Products



Enlarge

CWE™ International in scope and free for public use, CWE provides a unified, measurable set of software weaknesses that is enabling more effective discussion, description, selection, and use of software security tools and services that can find these weaknesses in source code and operational systems as well as better understanding and management of software weaknesses related to architecture and design.

CWE in the Enterprise

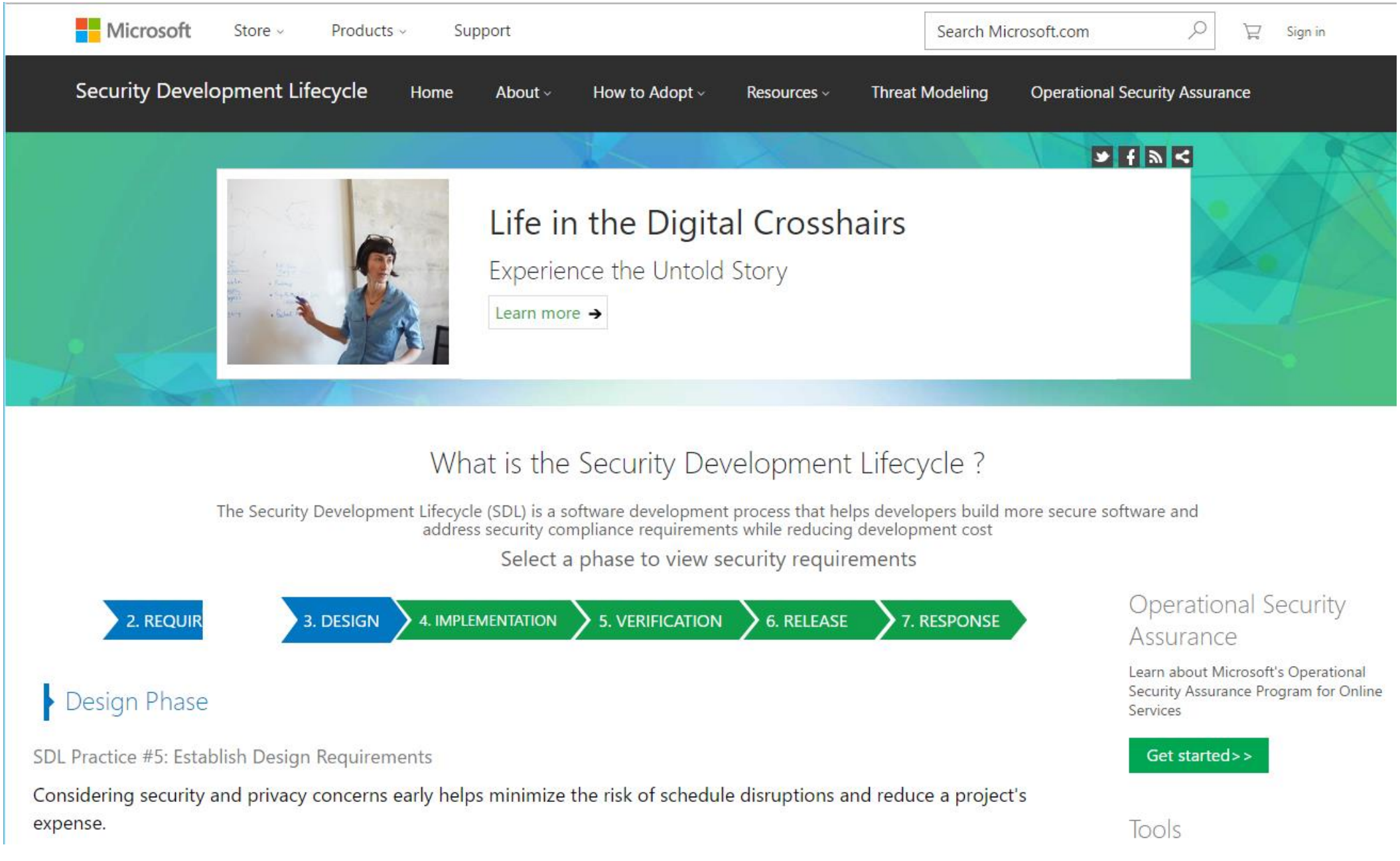
- Software Assurance
- Application Security
- Supply Chain Risk Management
- System Assessment
- Training
- Code Analysis
- Remediation & Mitigation
- NVD (National Vulnerability Database)
- *Recommendation ITU-T X.1524 CWE*, ITU-T CYBEX Series

Related Efforts

Vulnerabilities (CVE)

Weakness Scoring System (CWSS)

https://www.microsoft.com/en-us/sdl/



```
"use strict";

export module actions {
    class AsyncAction {

        private delegate;
        constructor(delegate) {
            this.delegate = delegate;
        }

        public invoke(): number {
            return setTimeout(this.delegate, 0);
        }
    }
}
```

```
"use strict";

export module actions {
    class AsyncAction {

        private delegate;
        constructor(delegate) {
            this.delegate = delegate;
        }

        public invoke(): number {
            return setTimeout(this.delegate, 0);
        }
    }
}
```

```
>> src/preso.ts[14, 20]: Forbidden setTimeout string parameter: this.delegate
```

```
"use strict";

export module actions {
    class AsyncAction {

        private delegate;
        constructor(delegate) {
            this.delegate = delegate;
        }

        public invoke(): number {
            return setTimeout(this.delegate, 0);
        }
    }
}
```

```
>> src/preso.ts[14, 20]: Forbidden setTimeout string parameter: this.delegate
```

Microsoft SDL Violation
CWE 95 - Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection')
CWE 676 - Use of Potentially Dangerous Function
CWE 242 - Use of Inherently Dangerous Function
CWE 116 - Improper Encoding or Escaping of Output"

```
"use strict";

export module actions {
    class AsyncAction {

        private delegate;
        constructor(delegate) {
            this.delegate = delegate;
        }

        public invoke(): number {
            return setTimeout(this.delegate, 0);
        }
    }
}
```

```
>> src/preso.ts[14, 20]: Forbidden setTimeout string parameter: x
>> src/preso.ts[8, 29]: expected parameter: 'delegate' to have a typedef
```

```
"use strict";

export module actions {
    class AsyncAction {

        private delegate: () => void;
        constructor(delegate: () => void) {
            this.delegate = delegate;
        }

        public invoke(): number {
            return setTimeout(this.delegate, 0);
        }
    }
}
```

```
let count = 0;
function wrapFunction(delegate) {
    count++;
    return new Function(delegate);
}

wrapFunction(() => {})();
wrapFunction(() => {})();
console.log(count);
```

```
let count = 0;
function wrapFunction(delegate) {
    count++;
    return new Function(delegate);
}

wrapFunction(() => {})();
wrapFunction(() => {})();
console.log(count);
```

```
>> src/preso.ts[5, 12]: forbidden: Function constructor with string arguments
```

Microsoft SDL Violation
CWE 95 - Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection')
CWE 676 - Use of Potentially Dangerous Function
CWE 242 - Use of Inherently Dangerous Function
CWE 116 - Improper Encoding or Escaping of Output

```
import React = require('react');

function renderTwitterStream() {
    return <iframe src={'/static/twitter.html'} />
}
```

```
import React = require('react');

function renderTwitterStream() {
    return <iframe src={'/static/twitter.html'} />
}
```

```
>> src/preso.tsx[4, 12]: An iframe element requires a sandbox attribute
```

CWE 915 - Improperly Controlled Modification of Dynamically-Determined Object Attributes

```
import React = require('react');

function renderTwitterStream() {
    return <iframe sandbox='allow-forms allow-scripts'
                   src={'/static/twitter.html'} />
}
```

```
import React = require('react');

function renderTwitterStream() {
    return <iframe sandbox='allow-scripts allow-same-origin'
                   src={'/static/twitter.html'} />
}
```

```
import React = require('react');

function renderTwitterStream() {
    return <iframe sandbox='allow-scripts allow-same-origin'
                   src={'/static/twitter.html'} />
}
```

```
>> src/preso.tsx[4, 28]: An iframe element defines a sandbox with both
allow-scripts and allow-same-origin
```

- When the embedded document has the same origin as the main page, it is strongly discouraged to use both `allow-scripts` and `allow-same-origin` at the same time, as that allows the embedded document to programmatically remove the `sandbox` attribute. Although it is accepted, this case is no more secure than not using the `sandbox` attribute.

Other Security Rules…

- no-eval
- no-document-domain
- no-document-write
- no-inner-html
- no-http-string
- react-no-dangerous-html

```
/**
 * Common Bugs and Correctness. The following rules should be turned on because they
 * find common bug patterns in the code or enforce type safety.
 */

"jquery-deferred-must-complete"
"mocha-no-side-effect-code"
"no-backbone-get-set-outside-model"
"no-bitwise"
"no-conditional-assignment"
"no-constant-condition"
"no-stateless-class"
"no-unnecessary-bind"
"no-unnecessary-override"
"no-unsafe-finally"
"promise-must-complete"
"react-this-binding-issue"
"react-unused-props-and-state"
"triple-equals"
"valid-typeof"
```

```
/**
 * Code Clarity. The following rules should be turned on because they
 * make the code generally more clear to the reader.
 */
```

```
…
"chai-prefer-contains-to-index-of"
"chai-vague-errors”
"max-func-body-length"
"no-function-expression"
"no-unnecessary-field-initialization"
"no-unsupported-browser-code"
"prefer-const"
…
```

Rulesets

- tslint-microsoft-contrib - https://github.com/Microsoft/tslint-microsoft-contrib
- tslint - https://github.com/palantir/tslint
- tslint-react - https://github.com/palantir/tslint-react
- tslint-eslint-rules - https://github.com/buzinas/tslint-eslint-rules
- codelyzer (Angular Rules)- https://github.com/mgechev/codelyzer
- vrsource-tslint-rules - https://github.com/vrsource/vrsource-tslint-rules

Resources

- recommended_ruleset.js and recommended tslint.json
- grunt validate-config – forces all rules to be defined
- tslint-warnings.csv – All CWE mappings for all rules
- Visual Studio Code – TSLint Plugin
- Microsoft Careers – 46 Open Position in Switzerland

```
[].constructor.constructor
```

```
[].constructor.constructor('alert(1)')()
```