Web API Misuse Detection in JavaScript Web Applications

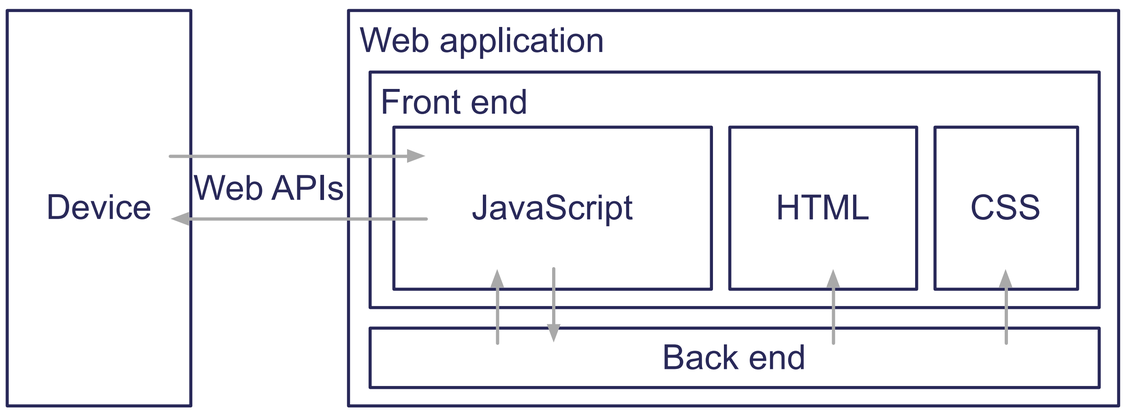
SungGyeong Bae
KAIST
imai0917@kaist.ac.kr

Hyunghun Cho
Samsung Electronics
hyunghun.cho@samsung.com

Sukyoung Ryu
KAIST
sryu.cs@kaist.ac.kr

Introduction

Web applications are prevalent thanks to the dynamic nature of JavaScript. However, because of the extreme freedom of JavaScript developers, it is very difficult to define and detect errors in JavaScript web applications. To mitigate the problem, various platform vendors provide Web APIs[1] in Web IDL for JavaScript developers. We present a technique to statically analyze Web APIs and JavaScript web applications that use Web APIs and to detect possible misuses in them.



JavaScript[2]

- Scripting language
- Prototype-based object-oriented language
- Functional language
- Dynamic language
- Implicit type conversion

Web IDL[3]

- Web Interface Description Language
- Web IDL specifies interfaces and types
- Web IDL has rich types (dictionary, enumeration, callback, …)

WebIDL Specification

```
[NoInterfaceObject] interface CalendarManager {
  void getCalendars(CalendarType type,
                    CalendarArraySuccessCallback successCallback,
                    optional ErrorCallback? errorCallback) raises(WebAPIException);
```

WebIDL Specification

```
[Callback=FunctionOnly, NoInterfaceObject]
interface CalendarArraySuccessCallback {
  void onsuccess(Calendar[] calendars);
};
```

WebIDL Specification

```
enum CalendarType { "EVENT", "TASK" };
```

WebIDL Specification

```
[NoInterfaceObject] interface Calendar {
  readonly attribute CalendarId id;
  readonly attribute DOMString name;
  CalendarItem get(CalendarItemId id) raises(W
```

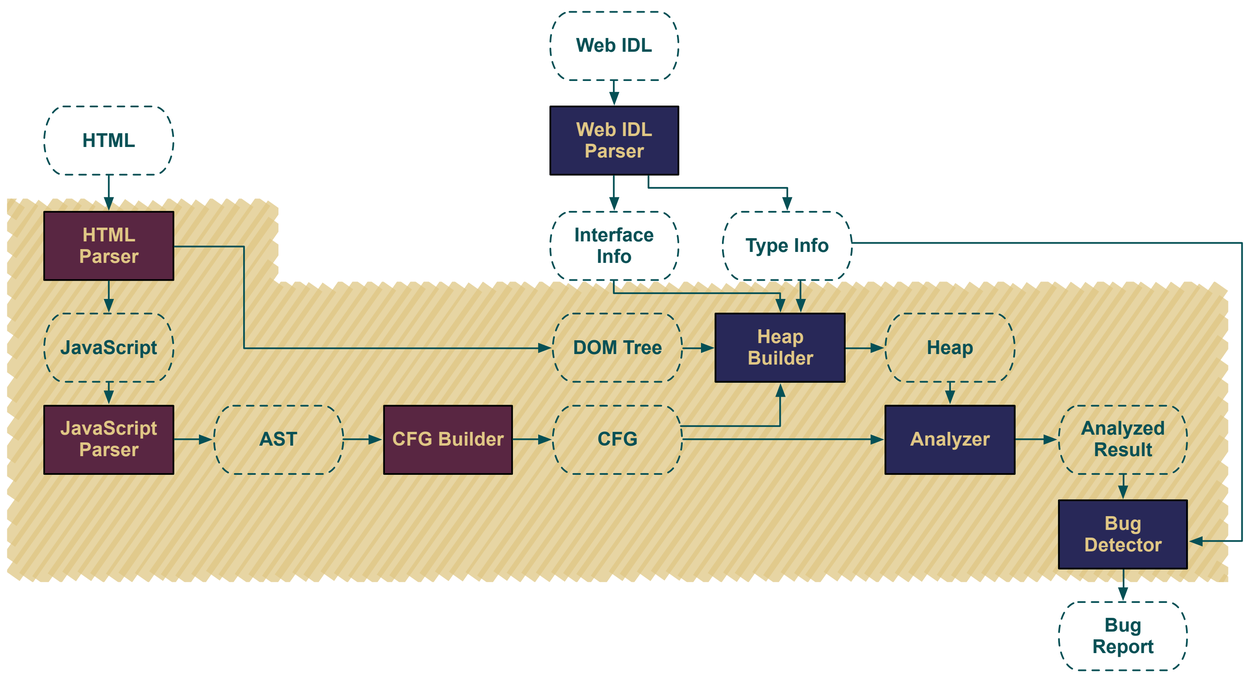
Web API Misuses

```
1 function CB(arg) { }
2 webapis.xxx;
```
Undefined APIs
```
2 webapis.calendar.getCalendars();
```
Wrong arity
```
2 webapis.calendar.getCalendars("EVEN", CB);
```
Wrong argument types
```
2 webapis.calendar.getCalendars("EVENT", CB);
```
Missing error callbacks
```
2 function sCB(calendars) { calendars[0].foo; }
3 webapis.calendar.getCalendars("EVENT", sCB, CB);
```
Wrong accesses to arguments in callback
```
2 function sCB(calendars) { calendars[0].id; }
3 webapis.calendar.getCalendars("EVENT", sCB, CB);
```
Missing try-catch statements
```
3 try {
4   webapis.calendar.getCalendars("EVENT", sCB, CB);
5 } catch (error) { }
```
```
1 var taskItem =
2   new webapis.CalendarTask({
3     boo: 0
4   });
```
Undefined properties of dictionaries
```
1 var taskItem =
2   new webapis.CalendarTask({
3     progress: 0
4   });
```

Methodology

Extend SAFE(Scalable Analysis Framework for ECMAScript)[4, 5]

- Web IDL parser to parse Web APIs
- Platform object that implements every interface defined in Web APIs
- Mock-up objects that model Web IDL types
- Callback function invocation that models Web API functions
- Web API misuse detection by using analysis results



Reference

[1] *Samsung Web API Guide*, http://img-developer.samsung.com/onlinedocs/samsung_webapi_guide/html/index.html
[2] *ECMAScript® Language Specification*, http://www.ecma-international.org/publications/standards/Ecma-262.htm
[3] *Web IDL*, http://www.w3.org/TR/WebIDL/
[4] Hongki Lee, et al., *SAFE: Formal Specification and Implementation of a Scalable Analysis Framework for ECMAScript*, In FOOL, 2012
[5] SAFE, http://safe.kaist.ac.kr