ERC 6th Workshop

안전한 JavaScript 프로그램 개발을 위해

Sukyoung Ryu

PLRG @ KAIST

June 26, 2011

KAIST Computer Science

PLRG @ KAIST 뉴스

- 바심
  - 정재성 배성경 박준영 김지응 : 바심 프로그래밍 언어와 새로운 중학교 컴퓨터 교재 개발
- Coq
  - 김지응 : Coq을 이용한 객체지향 언어의 타입 안전성 증명
  - 배성경 : Coq을 이용한 이분 그래프의 표현과 증명
  - 박지원 : 유클리드 기하학에서의 기계적인 증명 방법 개발과 Coq을 이용한 구현
- Biomedical Image Processing
  - 한명희 김윤승 : 메디컬 이미징 소프트웨어의 품질 향상을 위한 연구

KAIST Computer Science

PLRG @ KAIST 뉴스

- Fortress
  - "Type-checking Modular Multiple Dispatch with Parametric Polymorphism and Multiple Inheritance," OOPSLA 2011
  - 나현익: Inheritance with Subtyping
- 신뢰받지 않은 개발자와 서비스 업체, 사용자 모두가 안전한 JavaScript 웹 어플리케이션을 개발/사용하게 하는 시스템 개발
  - 류석영 : 안전한 JavaScript 프로그램 개발을 위해
  - 박창희 이홍기 : JavaScript 프로그램에서 with의 사용에 대한 연구
  - 강성훈: Desugaring을 이용한 JavaScript 모듈 시스템
  - 최재준: JavaScript 부분언어에 대한 조사 연구

KAIST Computer Science

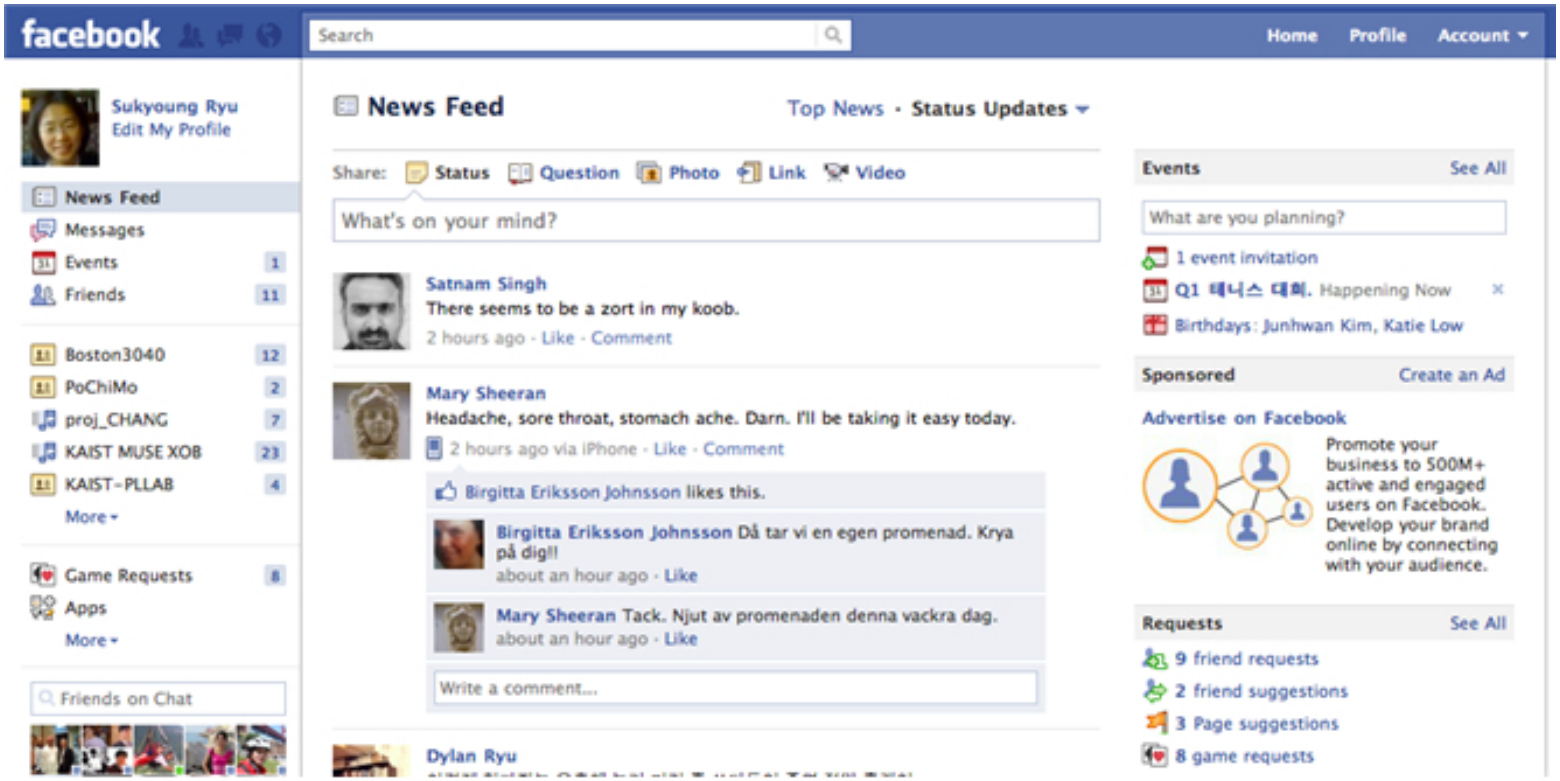
웹 어플리케이션 보안문제

- eBay, Amazon.com 등의 인터넷 기반 기업체부터 스마트폰까지 다양한 분야
- iGoogle, Blogger, Naver 등의 호스트 웹사이트에서 신뢰성이 검증되지 않은 외부 코드를 모아서 웹 페이지 생성
- 해커의 75%가 웹 어플리케이션 보안 취약성 공격
- Facebook, Twitter 등 95% 이상의 웹 어플리케이션이 보안에 취약
- 85%의 미국 산업체가 자료 누출 사고 경험

KAIST Computer Science

웹 어플리케이션 보안문제

- JavaScript 프로그래밍 언어를 사용한 Mashup 서비스



KAIST Computer Science

JavaScript

- 전형적인 스크립팅 언어
- 1995년에 Netscape의 Brendan Eich에 의해 개발
- ECMAScript로 표준화
- Java 스타일의 문법, 객체 기반 imperative 언어
- 함수를 값으로 사용
- 동적 타입 시스템
- “Any type can be converted to any other reasonable type”

KAIST Computer Science

JavaScript 문제점

- 모듈 시스템이 없음
- 단순한 타입만을 제공
- 타입 변환이 직관적이지 않음

  "A scripting language should never throw an exception [the script should just continue]" (Rob Pike, Google)
- 개발 및 분석 도구가 매우 미약함
- DOM(Document Object Model) 인터페이스를 통해 웹 페이지 내용을 동적으로 수정
- 웹 브라우저마다 DOM 구현이 호환성이 없음

KAIST Computer Science

JavaScript 웹 어플리케이션 보안문제

- JavaScript는 가장 널리 사용되는 웹 어플리케이션 개발 언어
- 보안이나 개인 정보 유출의 주 공격대상
- JavaScript 부분 언어
  - ADsafe: Yahoo!
  - FBJS: Facebook
  - Caja: Google

KAIST Computer Science

연구현황: 정적 검사

- 독일 Freiburg(Peter Thiemann): JavaScript 일부에 대해 동적 변환을 고려한 타입 시스템 개발
- 덴마크 Aarhus(Anders Møller): JavaScript 일부에 대한 타입 시스템을 요약 해석 기법을 사용하여 개발
- 미국 Stanford(John C. Mitchell): JavaScript 일부에 대한 의미구조를 엄밀하게 정의
- 미국 Brown(Shriram Krishnamurthi): JavaScript 대부분에 대한 의미구조를 엄밀하게 정의

연구현황: 동적 검사

- DoCoMo 연구소 미국 지사(Dachuan Yu): 브라우저를 수정하여 실제 프로그램 실행 중 발견되는 보안 문제 분석
- 미국 버클리(Dawn Song): 브라우저를 수정하여 객체 접근 권한을 동적으로 확인하는 방법

KAIST Computer Science

연구현황: 정적 & 동적

- 미국 Microsoft(Benjamin Livshits): 관점 지향(aspect oriented) 프로그래밍
- 미국 UCSD(Ranjit Jhala): 여러 단계의 정보 흐름(information flow) 분석
- 스웨덴 Chalmers(David Sands): JavaScript의 기본적인 메소드를 변형하여 보안 정책 검사

연구현황: 실험 연구

- 미국 William and Mary(Haining Wang): 실제로 보안 취약성이 있는 JavaScript 웹 어플리케이션의 분류
- 미국 UCSD(Ranjit Jhala): 대표적인 보안 취약 패턴 네 가지에 대해서 실제로 얼마나 발생하는지 실험
- 미국 퍼듀(Jan Vitek): 대부분의 JavaScript 연구에서 가정해온 성질들에 대한 실험적 연구
- 미국 퍼듀(Jan Vitek): JavaScript의 "eval"이 얼마나 많이, 어떻게 사용되는지에 대한 실험

KAIST Computer Science

연구현황: 테스팅

- DoCoMo 연구소 미국 지사(Dachuan Yu): 실행 중에 테스트의 입력 값을 자동으로 생성
- 덴마크 Aarhus 대학(Anders Møller): JavaScript 웹 어플리케이션을 자동으로 테스트하는 프레임워크
- 미국 퍼듀(Jan Vitek): JavaScript 벤치마크를 자동으로 생성

KAIST Computer Science

기존 연구의 문제점

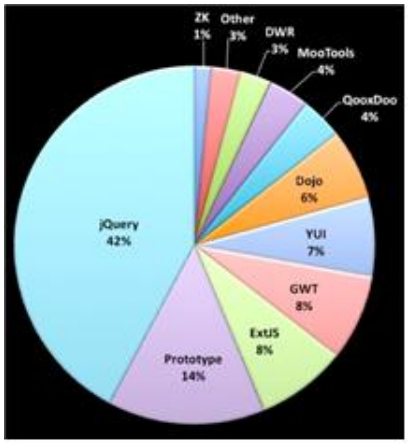
- 모듈 시스템 부재
  - > 작은 스크립트 개발을 위해 만들어진 언어
  - > ECMAScript Harmony
  - > 모듈 패턴
- 표준 라이브러리 부재

  jQuery, Prototype, ExtJS,

  GWT, YUI, Dojo, MooTools,

  MochiKit, Google Closure,

  SmartClient, Jindo

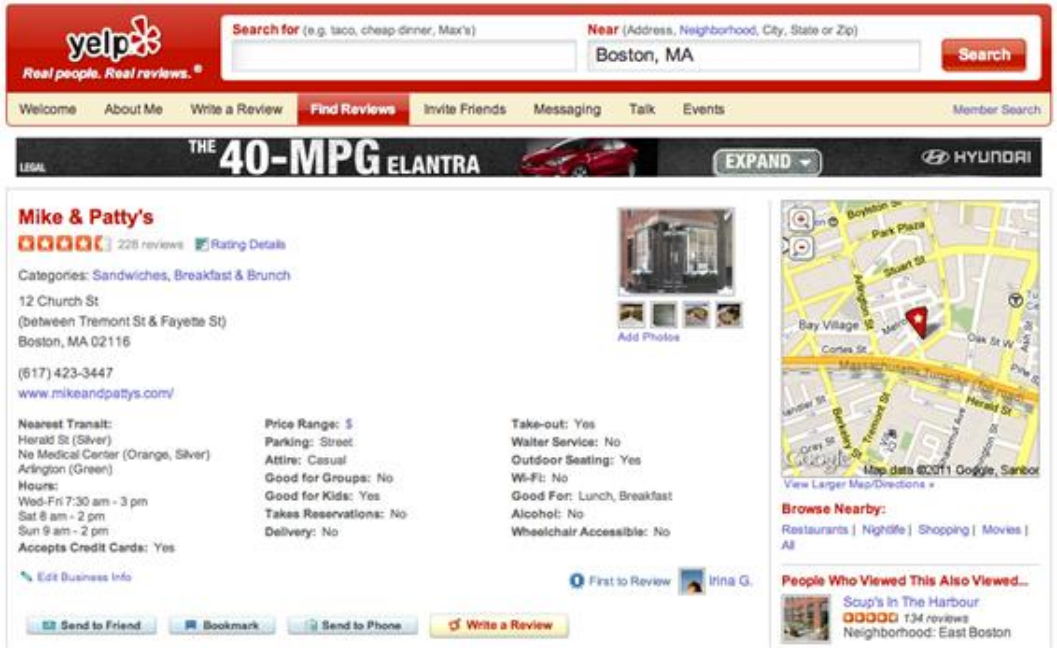


기존 연구의 문제점

- JavaScript 프로그램 개발 및 분석 지원을 위한 연구 미비
- 엄밀하게 정의되고 검증된 정적 성질 부재
  - ECMAScript 언어 스펙
  - Stanford(John C. Mitchell): ECMAScript 일부분을 엄밀하게 정의
  - Brown(Shriram Krishnamurthi): λ_{JS}

기존 연구의 문제점

- 다양한 외부 코드와 호스트 사이의 보안 정보 관리를 위한 세밀한 격리(isolation) 방법 부족
  - 브라우저 기반 샌드박싱(sandboxing): 동일 근원 정책(same origin policy)



KAIST Computer Science

기존 연구의 문제점

- JavaScript 웹 어플리케이션 실행 시의 동적 성질에 대한 분석 미약
  - 피상적인 사용 패턴 분석
  - 어떤 위험 패턴이 어떤 경우에 주로 사용되었는지 분석 필요
  - 더 안전한 프로그래밍 방법으로 대체될 수 있는지 연구 필요
  - 국내 웹사이트를 대상으로 한 웹 어플리케이션 보안 실태 연구 필요

KAIST Computer Science

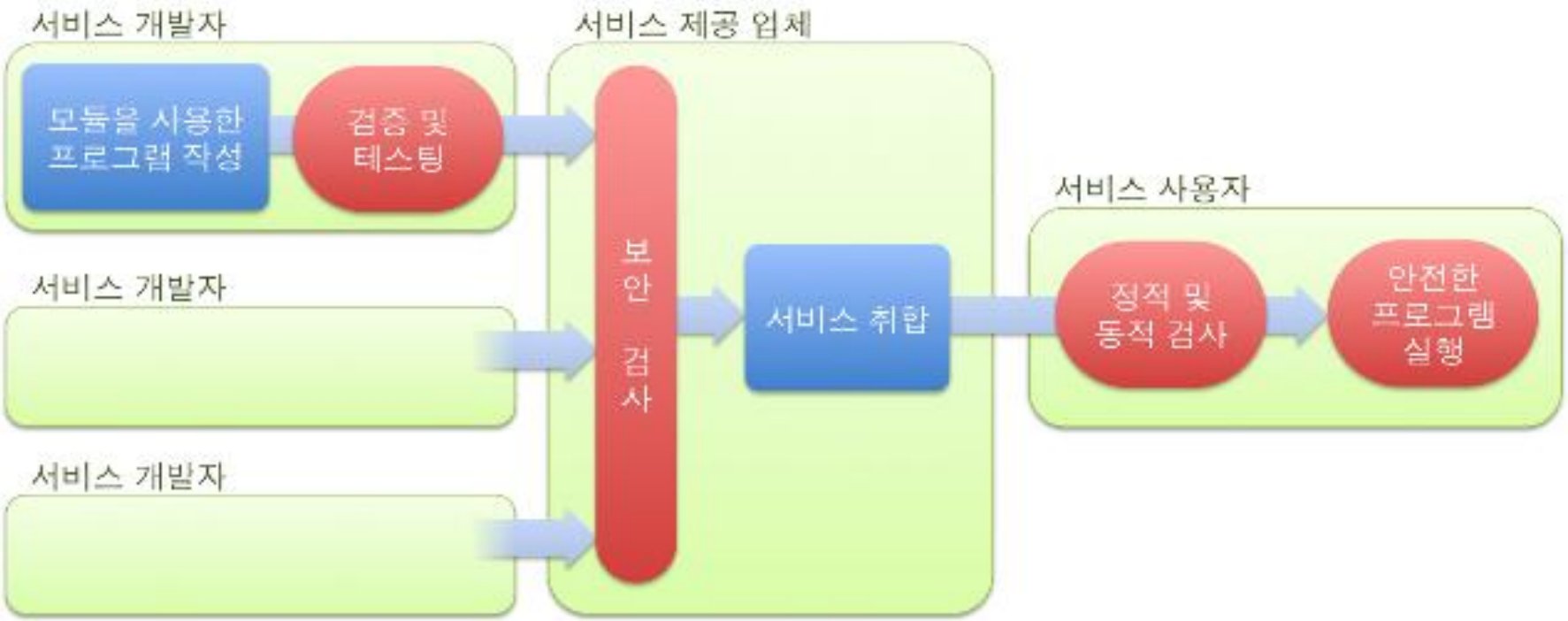
연구 계획

- 기존의 JavaScript 언어로 안전하게 변환시킬 수 있는 모듈 시스템 구축
- 외부의 JavaScript 웹 어플리케이션과 다른 코드 및 자원의 격리를 위한 언어 기반 샌드박싱 기법 개발
- 대표적인 웹사이트에 포함된 JavaScript 웹 어플리케이션의 동적 실행 양상 분석 및 진단

모듈 시스템

- 기존의 JavaScript 언어로 기계적으로 변환할 수 있는 모듈 시스템
- 기존 어플리케이션과의 호환성 문제 해결
- 모듈별 점진적 개발 방식의 장점을 JavaScript 웹 어플리케이션에도 적용
- 표준 라이브러리 부재의 문제를 모듈 시스템의 이름 공간(name space) 규칙으로 해결

프로그래밍 언어 기반 샌드박싱 기법



KAIST Computer Science

웹 어플리케이션의 동적 양상 분석

- 다양한 웹 어플리케이션의 실행 패턴을 분석하여 보안 취약성 진단
- 빈번하게 발생하는 보안 취약 패턴 인식
- 빈번하게 발생하는 보안 취약 패턴을 대체할 안전한 프로그래밍 패턴 개발
- 안전한 프로그래밍 패턴으로의 체계적인 변환 방법 개발

KAIST Computer Science

연구 진행 상황

- Desugaring을 이용한 JavaScript 모듈 시스템
- JavaScript 부분언어에 대한 조사 연구
- JavaScript 프로그램에서 with의 사용에 대한 연구

Sukyoung Ryu

sryu@cs.kaist.ac.kr
http://plrg.kaist.ac.kr