



Cross-Site Scripting (XSS) 취약점 탐지를 위한 분석

최 기환¹, 도 경구¹

¹한양대학교 컴퓨터공학과 프로그래밍언어 연구실

{khchoi, doh}@plasse.hanyang.ac.kr



Objectives

- 웹 어플리케이션이 크로스 사이트 스크립팅 취약점을 가지고 있는지 그렇지 않은지를 알아내기 위한 자동화 툴을 만드는 것이 목표이다. 이를 위해서 다음의 두 가지 분석 방법을 이용하여 자동화 툴을 개발하고 있다.

- 동적 분석(Dynamic Analysis) : 침투 시험기 개발
- 정적 분석(Static Analysis)

Introduction

웹 어플리케이션은 사용자들에게 좀 더 나은 서비스를 제공하기 위해 동적인 웹 페이지들을 제공한다. 그러나 이렇게 제공되는 동적인 웹 페이지들은 의도적으로 조작된 URL을 통해 페이지에 신뢰할 수 없는 페이로드(Payload)를 삽입하는 방법 등을 통해 공격자들의 공격 대상이 되는 위험성을 가지고 있다. 이를 이용한 대표적인 웹 어플리케이션 공격 방법이 크로스 사이트 스크립팅(Cross-Site Scripting, XSS)이다.

크로스 사이트 스크립팅 취약점은 사용자가 입력한 데이터를 어플리케이션에서 처리할 때 콘텐츠에 대한 검증이나 인코딩을 하지 않고 어플리케이션에서 받아들이거나, 웹 브라우저로 전송할 때 발생한다. 즉, 공격자의 악의적인 JavaScript, VBScript, ActiveX, HTML, 또는 Flash를 사용자의 취약한 동적 페이지 내로 삽입을 허용할 때 피해자의 브라우저에서 스크립트가 실행됨으로써 발생한다. 이로 인해 개인정보의 위협, 쿠키 가로채기 또는 조작, 잘못된 인증 요청, 시스템에 대한 악의적인 코드 등을 실행함으로써 피해자의 세션이 노출되거나 웹 사이트의 변조를 할 수 있는 등의 웹 보안 문제가 야기된다.

잘 알려진 크로스 사이트 스크립팅의 세 가지는 반사(Reflected or Non-persistent), 저장(Stored or Persistent), 그리고 문서 객체 모델(Document Object Model) 기반 크로스 사이트 스크립팅이다.

What happens to XSS?

- 다음의 URL 'http://www.vulnerable.site/welcome.html'이 아래의 내용을 포함하고 있다고 가정하자:

```
<HTML>
<TITLE>Welcome!</TITLE>
Hi
<SCRIPT>
  var a = document.URL;
  a = unescape(a);
  document.write(a.substring(a.indexOf("name=")+5,a.length));
</SCRIPT>
Welcome to our system!
</HTML>
```

클릭을 유도하는 다음과 같은 주소를 전자우편 등을 통해 사용자에게 보내면:

```
'http://www.vulnerable.site/welcome.html?name=
<script>alert(document.cookie)</script>'
```

- 사용자가 웹 어플리케이션으로 요청(request)을 보낸다.
- 웹 어플리케이션은 사용자가 보내온 입력을 동적 페이지에 포함해 보내게 된다.
- 스크립트가 포함된 동적 페이지는 사용자의 브라우저에서 그대로 실행되게 된다.

이렇게 스크립트가 적절한 필터 함수 등에 의해 걸러지지 않고 실행되기 때문에 XSS 취약점이 발생하는 것이다.

Attack Scenarios

- 반사된 XSS (Reflected or Non-persistent XSS)

클라이언트에서 제공한 데이터가 웹 어플리케이션의 응답 페이지에 바로 포함돼 실행되는 것을 악용하는 공격 방법이다. 사용자의 입력에 대해 마옴표와 같은 인용처리를 하지 않고 그대로 페이지에 포함시키는 웹 어플리케이션의 경우, 사용자의 입력에 스크립트가 포함돼 있으면, 그 스크립트가 동적으로 생성되는 페이지에 포함돼 클라이언트에서 실행된다.

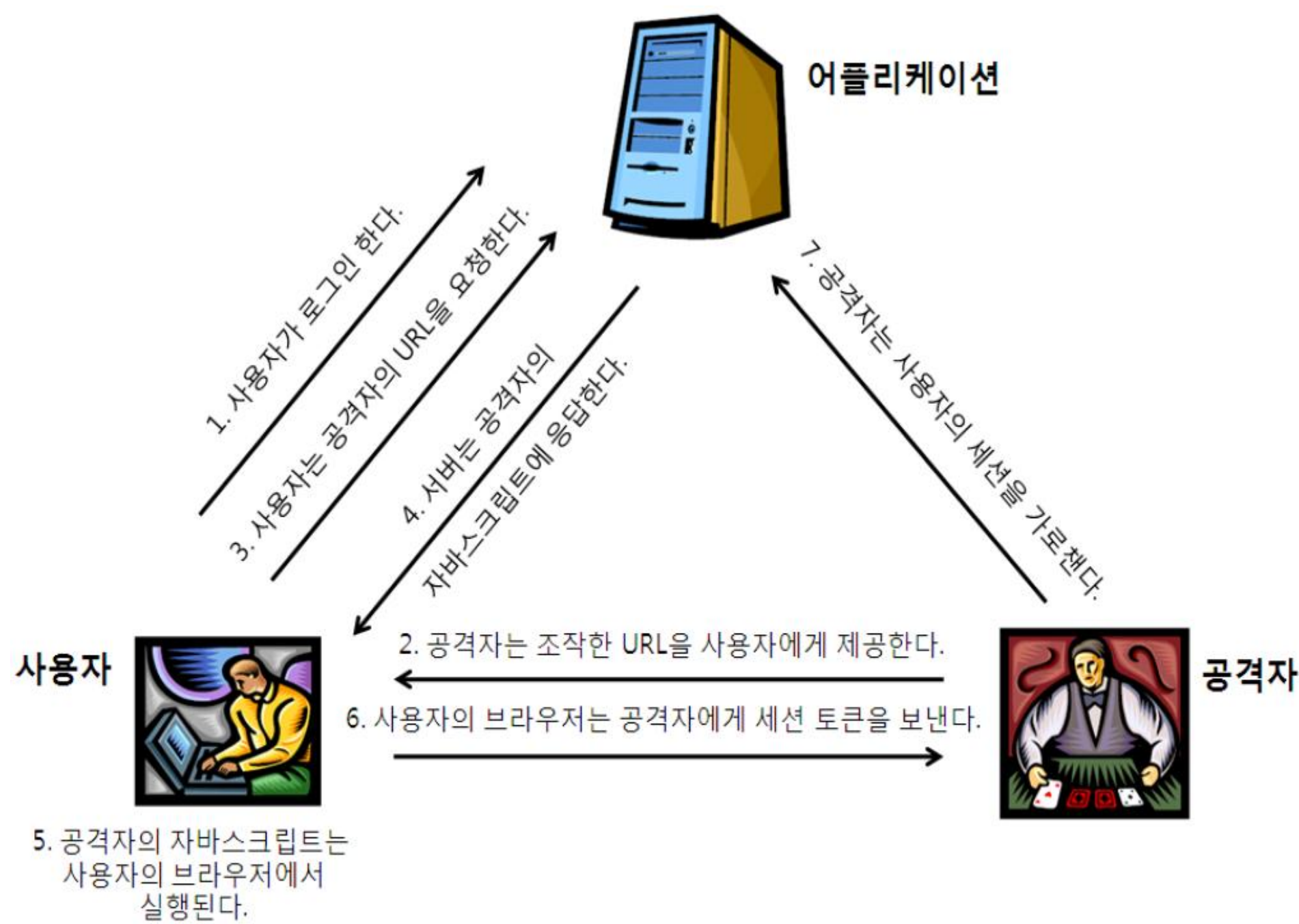


그림 1: 반사된 XSS 공격 시나리오

- 저장된 XSS (Stored or Persistent XSS)

공격자가 공격을 위해 가장 많이 살펴보는 곳이 같은 사이트를 방문하는 다른 사용자들에게 보이는 데이터를 입력하는 부분이다. 즉, 방명록, 게시판, 댓글 등에 글을 남기기 위해 사용자가 입력하는 부분이다. 공격자는 이러한 부분에 악의적인 스크립트를 저장하고, 후에 어떤 사용자가 해당 게시물을 클릭하는 순간 스크립트가 웹 페이지에서 로드되면서 실행되는 것이다.



그림 2: 저장된 XSS 공격 시나리오

- DOM 기반 XSS (Document Object Model based XSS)

조작된 URL을 통해 DOM 데이터에 접근할 때 사용할 수 있는 DOM 객체를 포함하는 자바스크립트가 존재하는 페이지가 있을 때, 사용자측 브라우저에서 DOM 객체를 통해서 공격 코드를 가지고 와 공격이 실행된다.

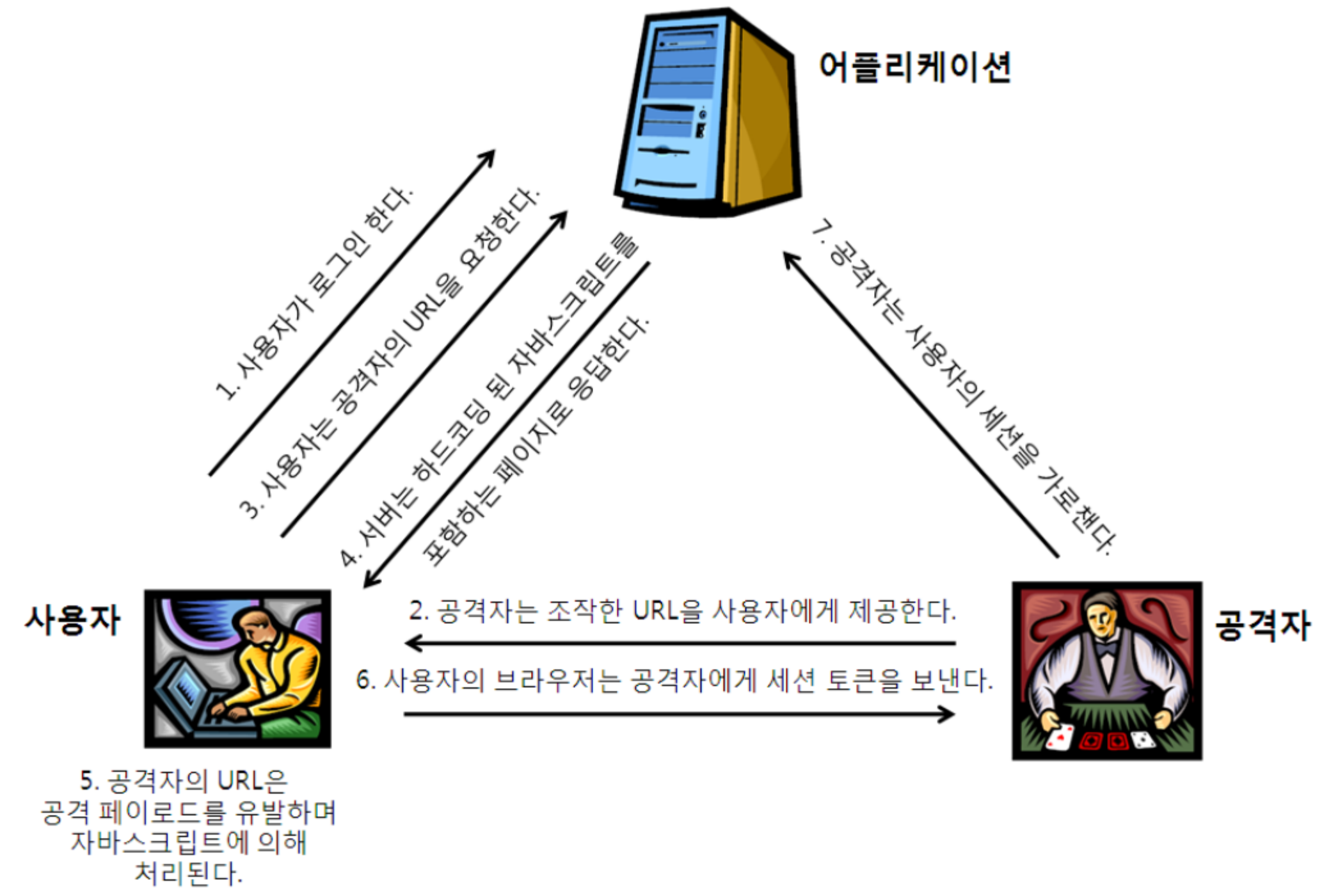
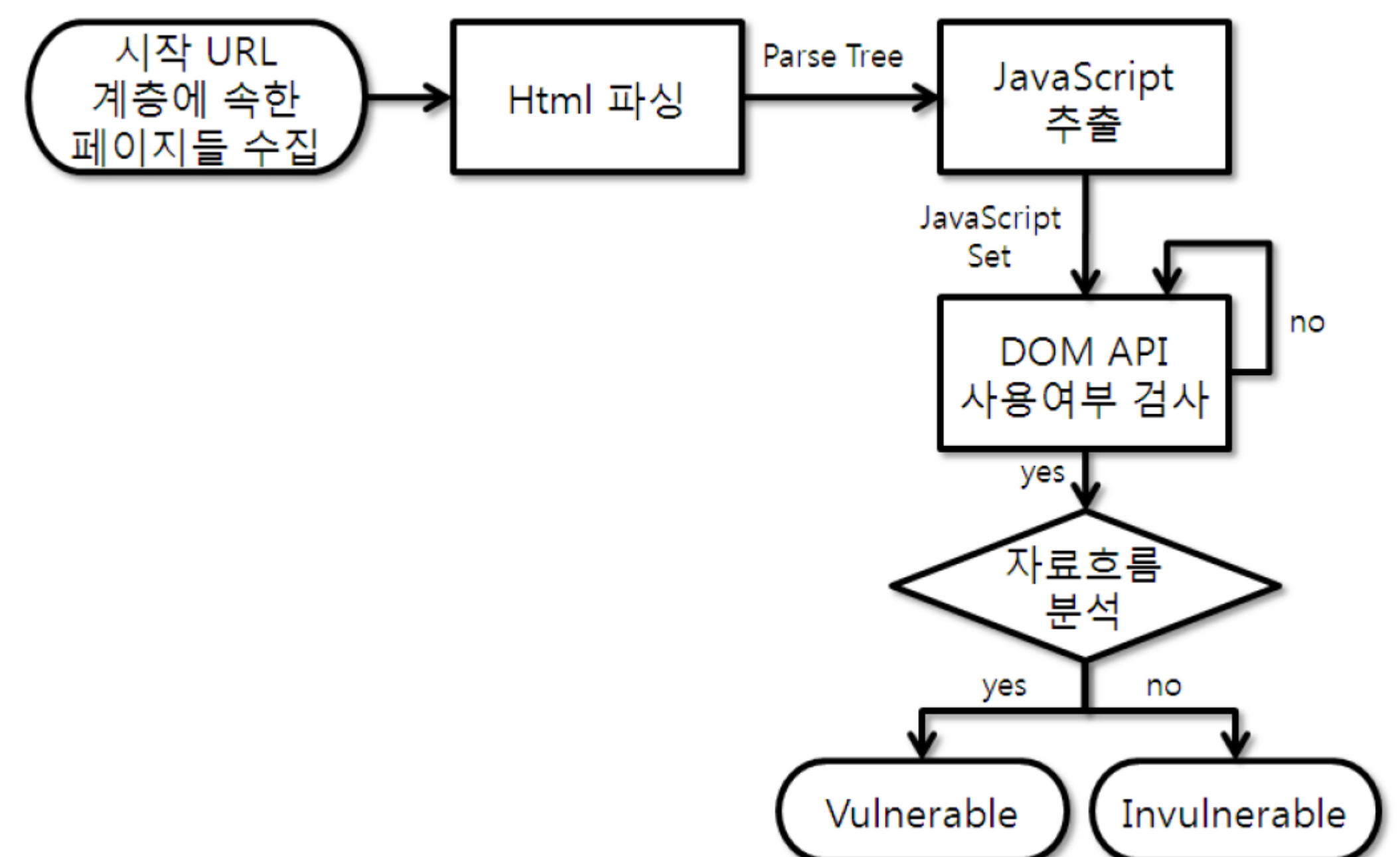


그림 3: DOM 기반 XSS 공격 시나리오

Detection Technologies

- 동적 분석(Dynamic Analysis)

동적 분석 방법은 일종의 침투 시험기로서 웹 어플리케이션이 어떻게 구성되어 있는지 알지 못하는 상태에서 수행하는 블랙박스 테스트(Blackbox Testing)이다.



- 정적 분석(Static Analysis)

정적 분석 방법은 스크립트 코드가 드러나 있지 않고 출력 명령문에 의해서 만들어져 삽입 되어 있는 경우 스크립트에 대한 정보를 알 수 없어 취약점을 가지고 있는 스크립트의 경우 문제가 될 수 있다.

Conclusions

- XSS 취약점을 탐지하는 자동화된 툴을 개발하고 있다. 이를 위해 두 가지 분석 방법을 사용하고 있다.

- 동적 분석(Dynamic Analysis)
 - * 침투 시험기 개발
- 정적 분석(Static Analysis)
 - * Abstract Parsing [Doh et al, SAS'09]

Further Study

- XSS 취약점 이외에 취약점을 발생시키는 다양한 공격 방법이 존재한다. 이러한 취약점들을 찾아내는 자동화된 탐지 툴을 계속 확장해나갈 것이다.
 - 세션 관리 공격, 접근 통제 공격, 코드 삽입 공격, 경로 탐색 공격, ...etc