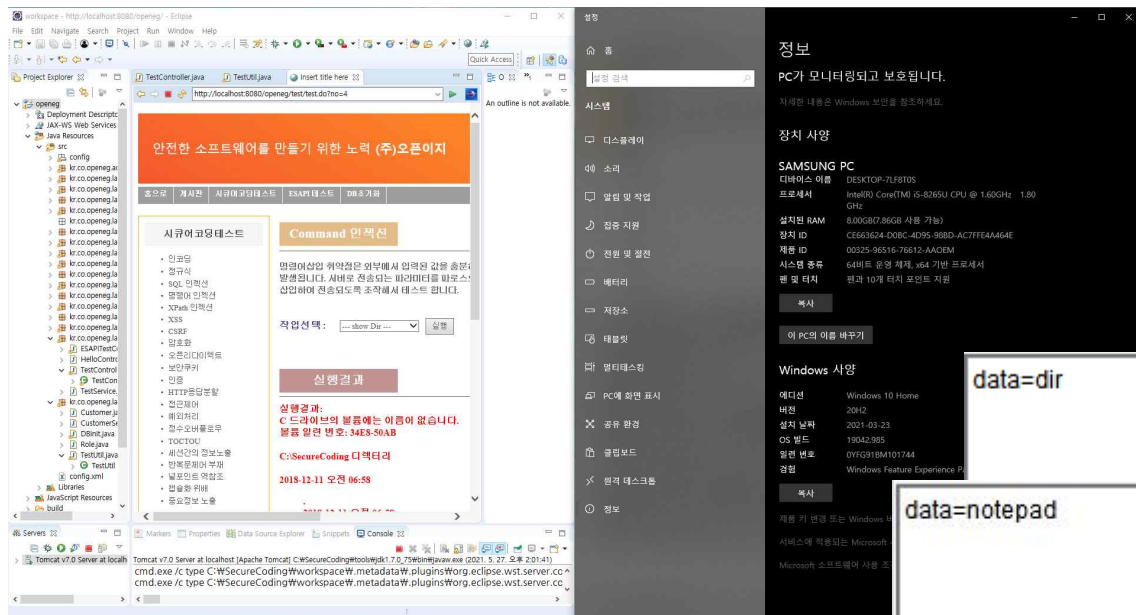
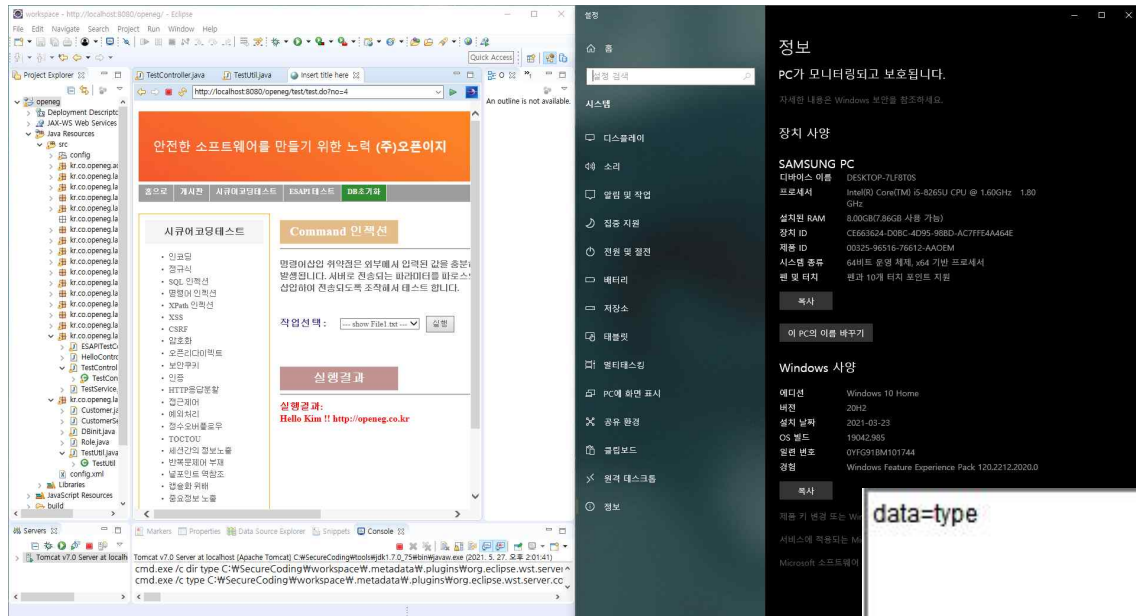


1. 운영체제 명령어 삽입



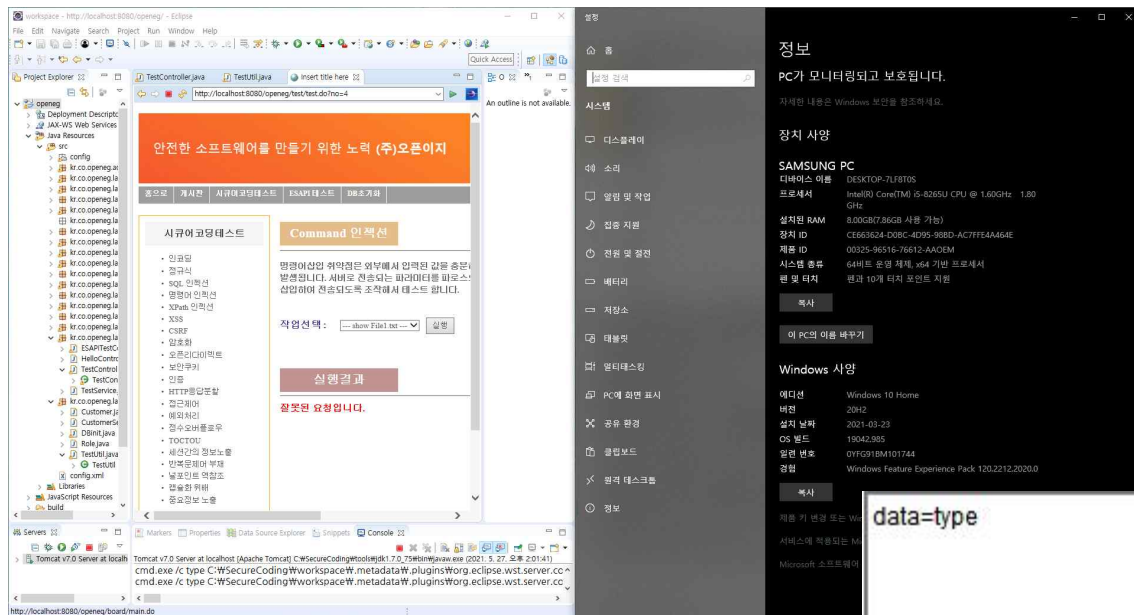
(코드수정)

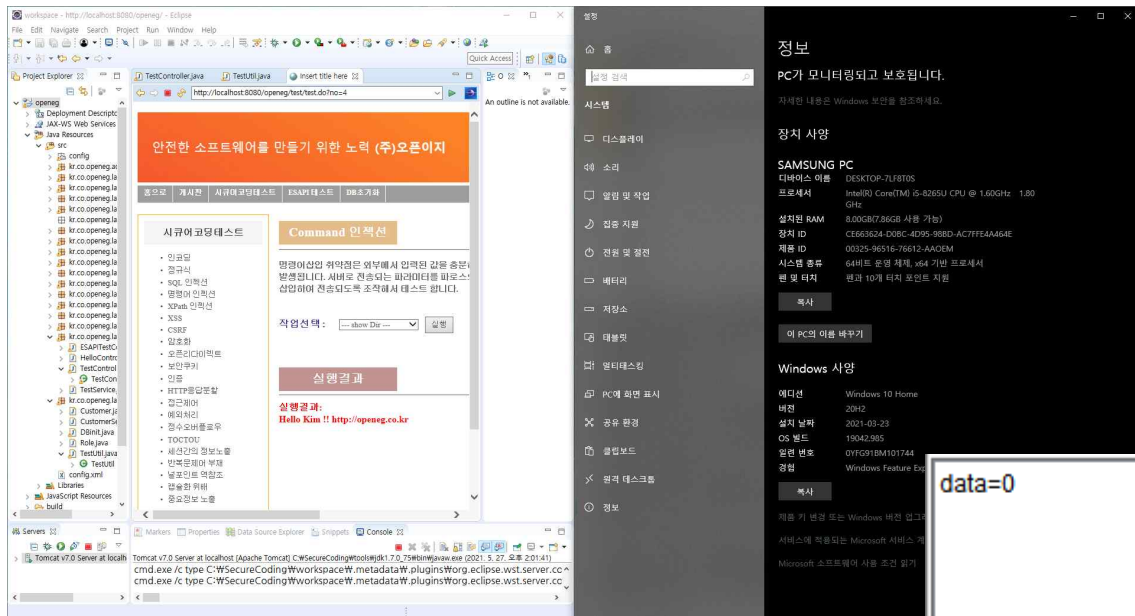
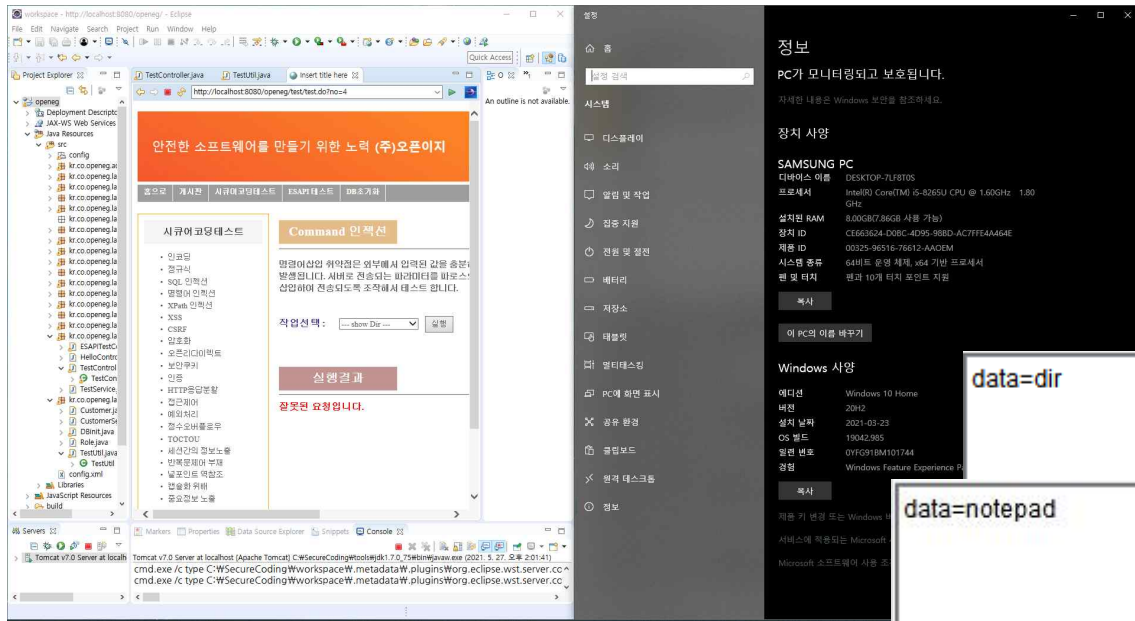
```
String[] allowCommand= {"type", "dir"};

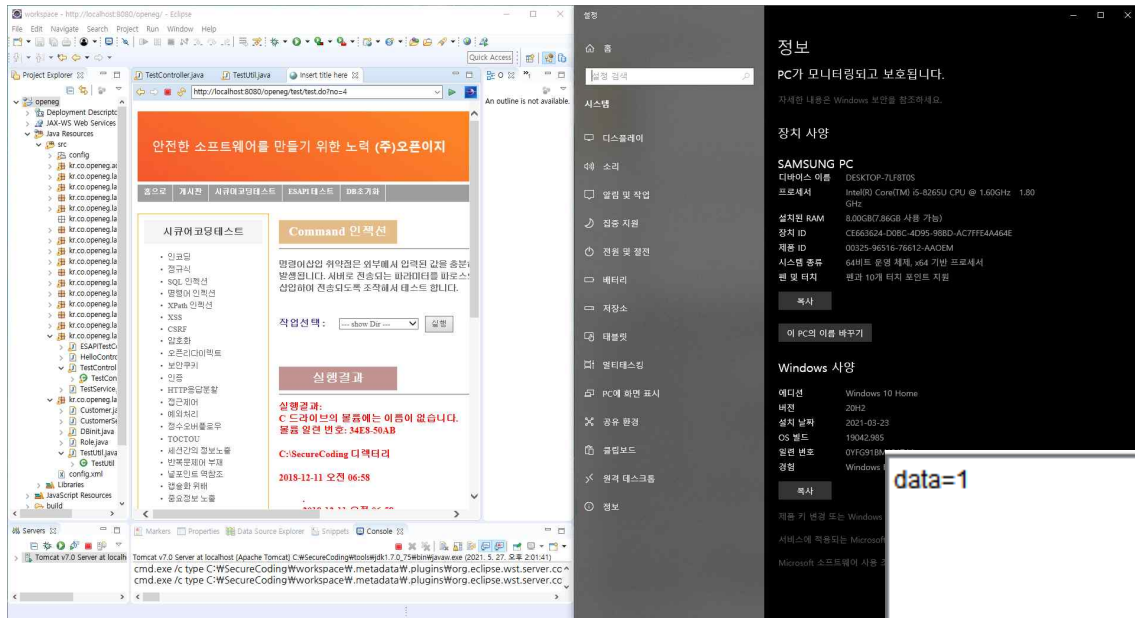
int index=TestUtil.getInt(data);

if(index < 0 || index > 1) {
    buffer.append("잘못된 요청입니다.");
    return buffer.toString();
}
else {
    data=allowCommand[index];
}

if ( data != null && data.equals("type")) {
    data=data+" "+
        request.getSession().getServletContext().getRealPath("/") +
        "file1.txt";
    System.out.println(data);
}
```

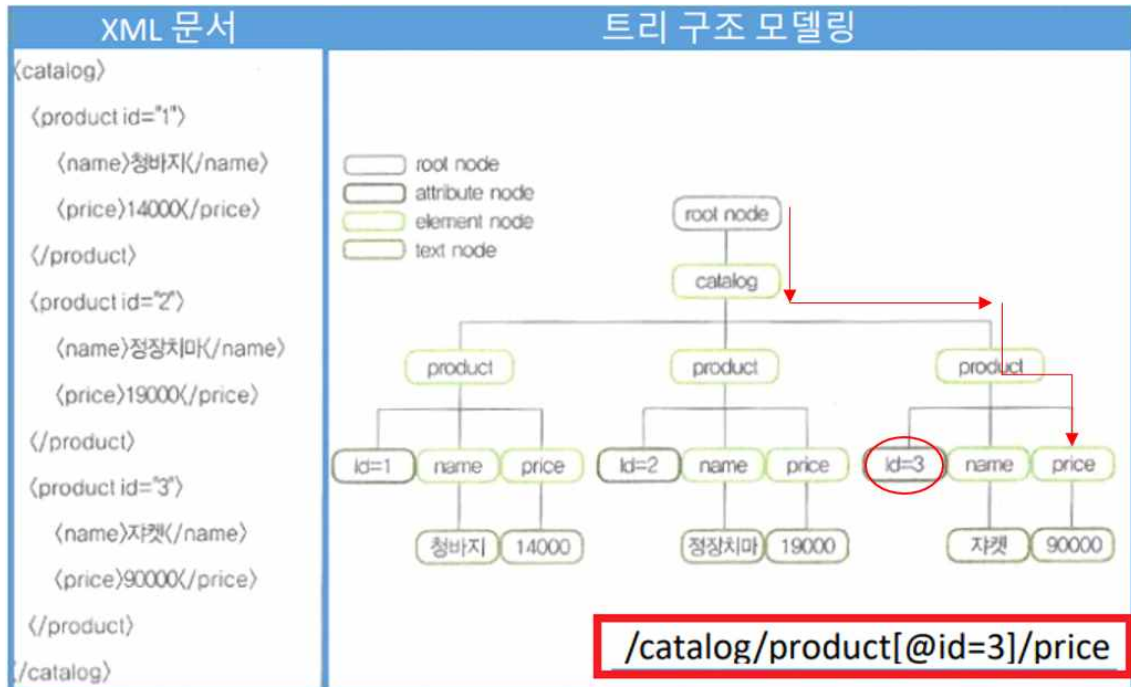






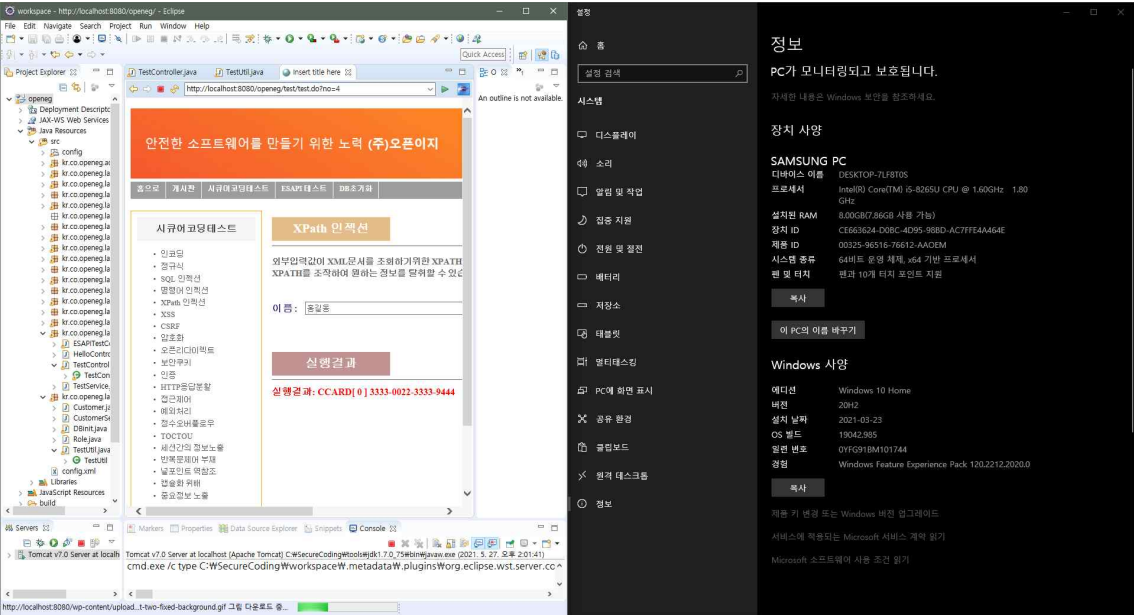
2. XPath 삽입

- XPath(XML Path Language): XML문서에 저장된 데이터를 애플리케이션에서 검색하거나 읽기 위해 사용하는 표현 방식



- XPath 삽입 취약점 발생 원인: 사용자 입력을 받아, 입력 값에 대한 검증 없이 동적으로 XPath 쿼리를 생성하면 공격자가 해당 쿼리문의 의미를 수정할 수 있음

(1) XPath 삽입 공격



XPath 인젝션

외부입력값이 XML문서를 조회하기위한 XPATH 생성에 사용되는 경우, 공격자는 '='[@ 와 같은 문자를 이용하여 XPATH를 조작하여 원하는 정보를 탈취할 수 있습니다.

이름:

실행결과

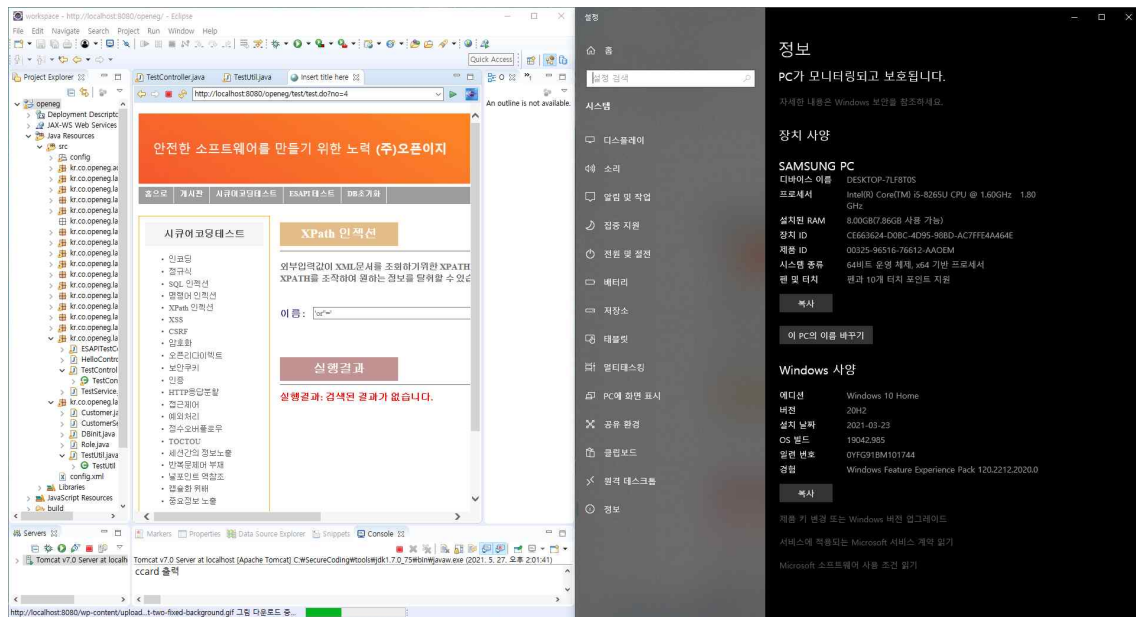
실행결과: CCARD[0] 3111-0022-3333-9444
CCARD[1] 3333-0022-3333-9444
CCARD[2] 1115-2266-7733-4144
CCARD[3] 3331-5553-3333-8884

(2) XPath 삽입 방어

(코드 수정)

```
public String XPathFilter(String input)
{
    return input.replaceAll("[',WW[]", "");
}

// String expression = "/addresses/address[@name='"+name+"']/ccard";
String expression = "/addresses/address[@name='"+XPathFilter(name)+"']/ccard";
```



- XPathFilter(): 입력된 값에 대해서 replaceAll이라는 함수를 사용하여 쿼리문에서 사용되어지면 안되는 특수문자들을 공백으로 바꾸어 주는 필터링 작업을 수행한다.

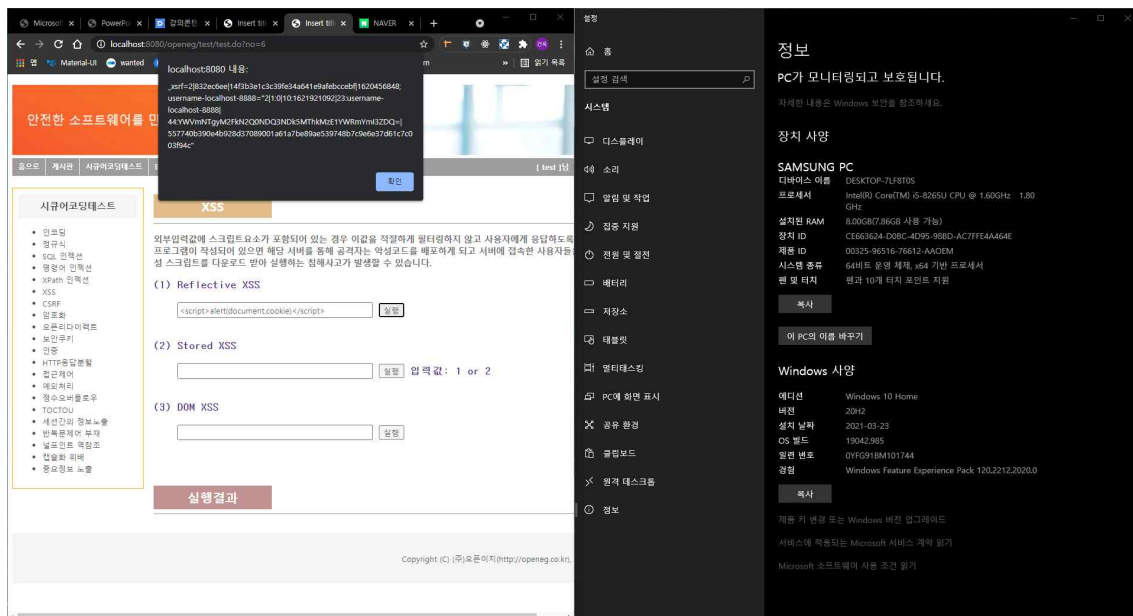
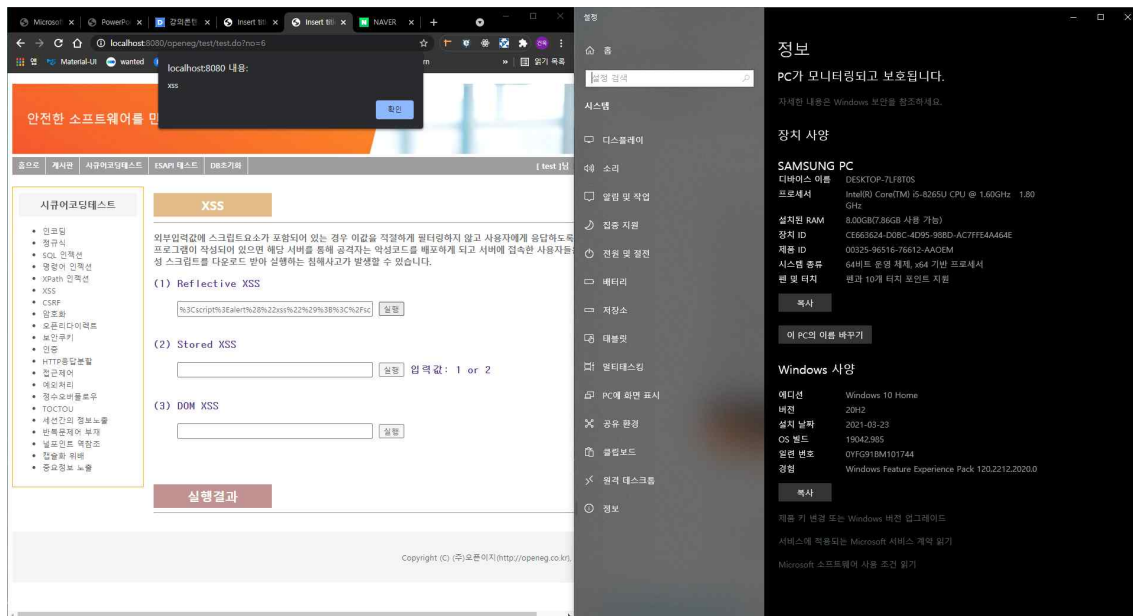
3. 크로스 사이트 스크립팅(XSS) 공격 실습

- 크로스 사이트 스크립팅(XSS) 취약점: 외부 입력 값이 충분한 검증 없이 동적으로 생성되는 응답 페이지에 사용되는 경우
- Reflective XSS(반사공격): 공격자가 악성 스크립트가 포함된 URL을 클라이언트에 노출한 후 클릭을 유도하고 악성 행위를 수행한다.
- Stored XSS: 악성 스크립트를 데이터베이스에 저장하여 모든 사용자들이 해당 스크립트를 실행하게 함으로써 악성 행위를 수행한다.
- XSS 취약점의 발생 원인: 사용자의 입력 값이나 데이터베이스에서 검색한 결과값을 검증하지 않고 응답의 일부로 사용하기 때문에 발생한다.

(XSS 공격 차단)

1. 입출력 값에 필터링 적용
2. 오픈소스 라이브러리 활용 XSS Filter(lucy xss filter)

(1) Reflective XSS 공격



(2) Reflective XSS 방어

(코드수정)

```
// Reflective XSS 테스트
@RequestMapping(value="/test/xss_test.do", method = RequestMethod.POST)
@ResponseBody
public String testXss(HttpServletRequest request) {
    StringBuffer buffer=new StringBuffer();
    String data=request.getParameter("data");

    //added
    try {
        data = URLDecoder.decode(data, "UTF-8");
        System.out.println("data:" +data);
    }
    catch(IOException e)
    {
        System.out.println(e);
    }

    XssFilter filter = XssFilter.getInstance("lucy-xss-superset.xml");
    buffer.append(filter.doFilter(data));
    return buffer.toString();

    //buffer.append(data);
    //return buffer.toString();
}
```

The screenshot displays a web browser window with a security tool interface on the left and a Windows system information window on the right.

Security Tool Interface:

- Header:** 안전한 소프트웨어를 만들기 위한 노력 (주)오픈이치
- Navigation:** 홈으로, 개시판, 취약요소검출테스트, ESAPI 테스트, DB조기화
- Left Panel:** 취약요소검출테스트
 - 인프라
 - 공공망
 - SQL 인젝션
 - 명령어 인젝션
 - XXPath 인젝션
 - XSS
 - CSRF
 - 암호화
 - 오류처리
 - 보안공기
 - 인증
 - HTTP응답분할
 - 접근제어
 - 데이터베이스
 - 경수요버블프루
 - TOCTOU
 - 계산기의 정보노출
 - 반복문제어 부재
 - 널포인트 역참조
 - 합출력 위해
 - 종료점의 노출
- Main Content:**
 - XSS**
 - 외부입력값에 스크립트요소가 포함되어 있는 경우 이값을 적절하게 필터링하지 않고 사용자에게 응답하는 프로그램이 작성되어 있으면 해당 서버를 통해 공격자는 악성코드를 배포하게 되고 서버에 접속한 사용자 스크립트를 다운로드 받아 실행하는 침해사고가 발생할 수 있습니다.
 - (1) Reflective XSS**
 - Input:
 - Button: 실행
 - (2) Stored XSS**
 - Input:
 - Button: 실행
 - Label: 입력값: 1 or 2
 - (3) DOM XSS**
 - Input:
 - Button: 실행
 - 실행결과**
 - Output:

```
<script>alert('xss')</script>
```
- Footer:** Copyright (C) (주)오픈이치(http://openeg.co.kr)

Windows System Information Window:

- 정보**
- PC가 모니터링되고 보호됩니다.**
- 장치 사양**
- SAMSUNG PC**
- 디바이스 이름:** DESKTOP-7L58T0S
- 프로세서:** Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
- 설치된 RAM:** 8.00GB(7.86GB 사용 가능)
- 장치 ID:** CEG63624-D0BC-4D95-958D-AC7FFE4A46AE
- 제품 ID:** 00325-96516-76612-AAOEM
- 시스템 종류:** 64비트 운영 체제, x64 기반 프로세서
- 빈 및 타지:** 빈과 10개 타지 포인트 지원
- Windows 사양**
- 에디션:** Windows 10 Home
- 버전:** 20H2
- 설치 날짜:** 2021-03-23
- OS 빌드:** 19042.585
- 원본 번호:** 0YFG91BM101744
- 권장:** Windows Feature Experience Pack 120.2212.2020.0
- 제품 키 변경 또는 Windows 버전 업그레이드:**
- 서비스에 활용되는 Microsoft 서비스 계약 읽기**
- Microsoft 소프트웨어 사용 조건 읽기**