□ 단국대학교 > 사이버보안 학과 > 인터넷 보안 > 보고서 템플릿

25년 1학기 인터넷 보안 수업

XSS

2025년 4월 1일

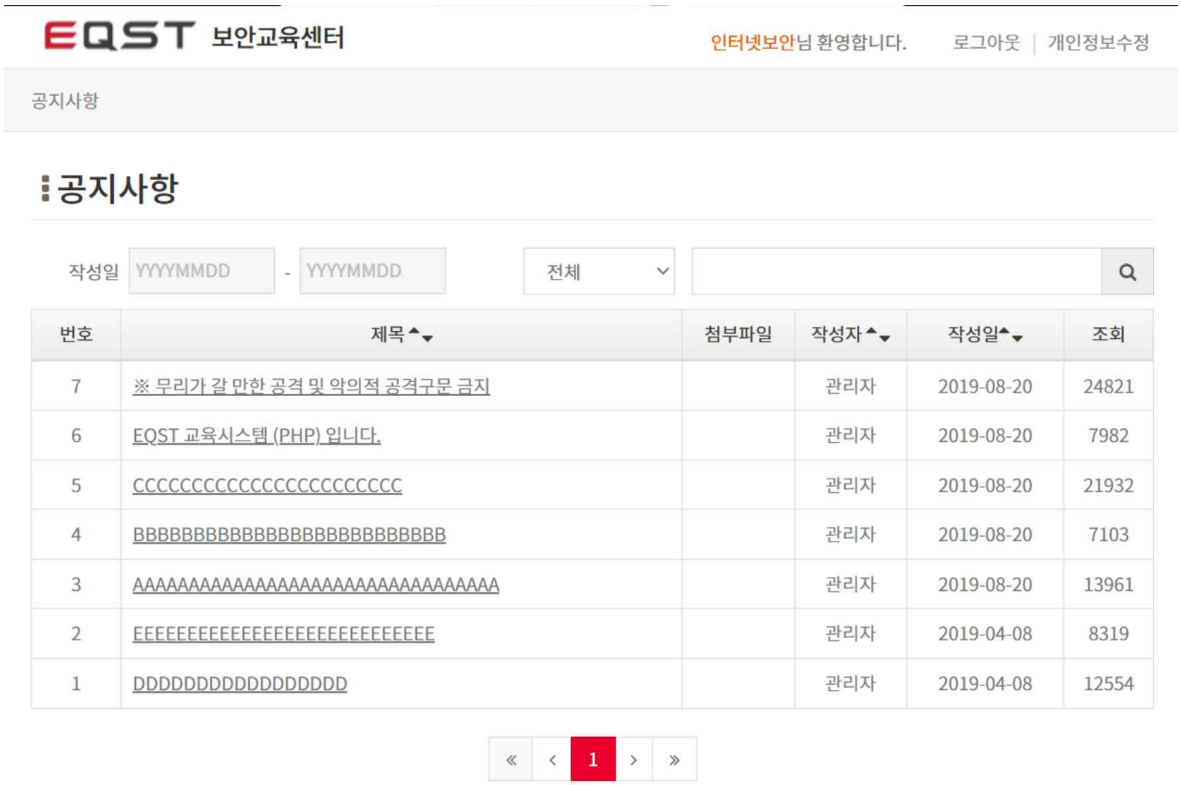
학번 : 32230324
이름 : 김가람

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



(다음 페이지에 이어서)

Step 2. 파라미터 확인

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

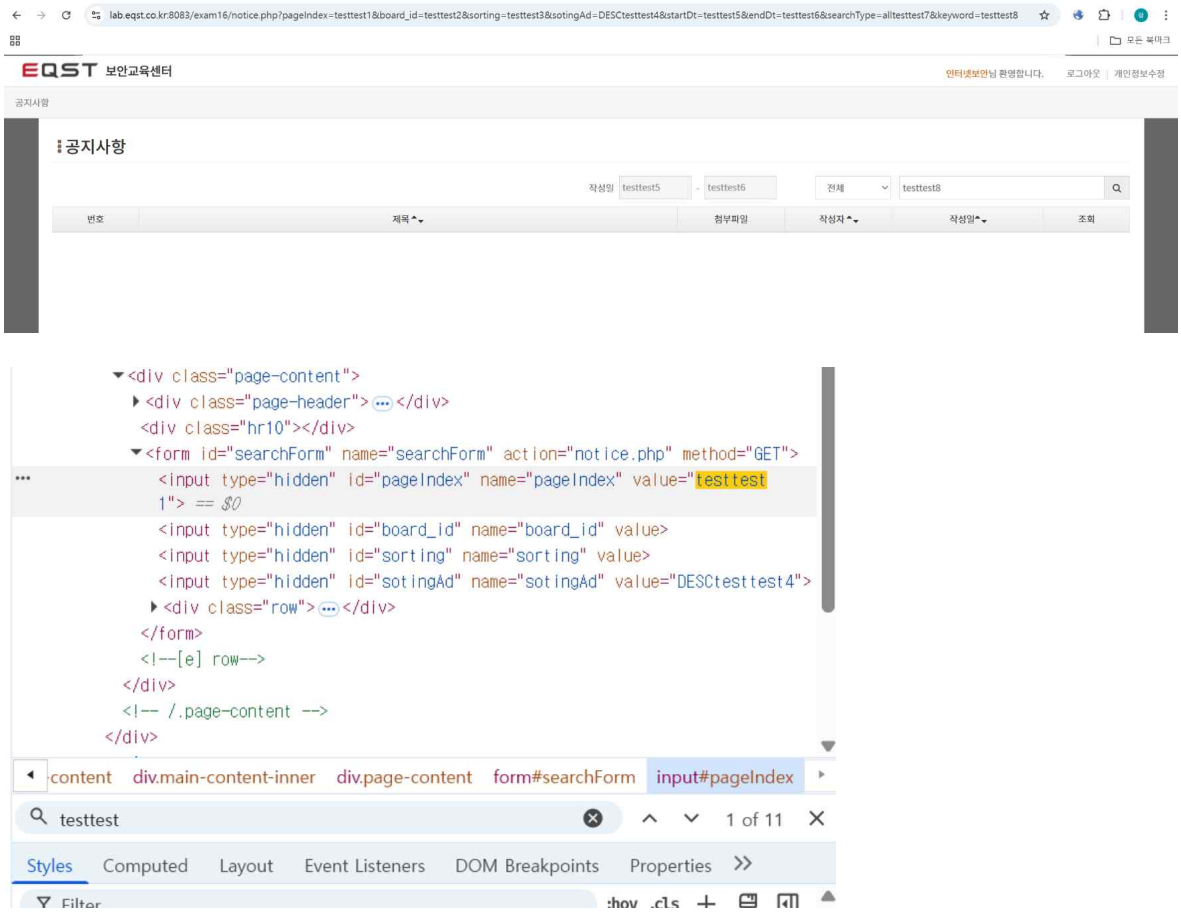
Step 3. 응답에 삽입되는 파라미터 찾기

: 각 파라미터가 응답의 어디에 반영되는지 찾기 위해 각각에 눈에 띄는 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



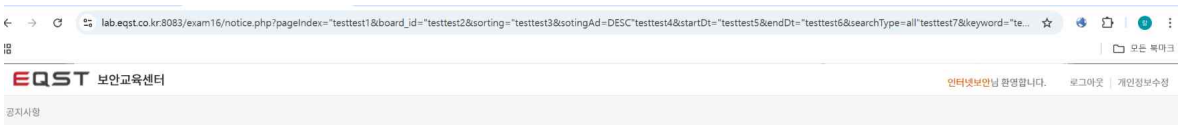
Step 4. 필터링 확인

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



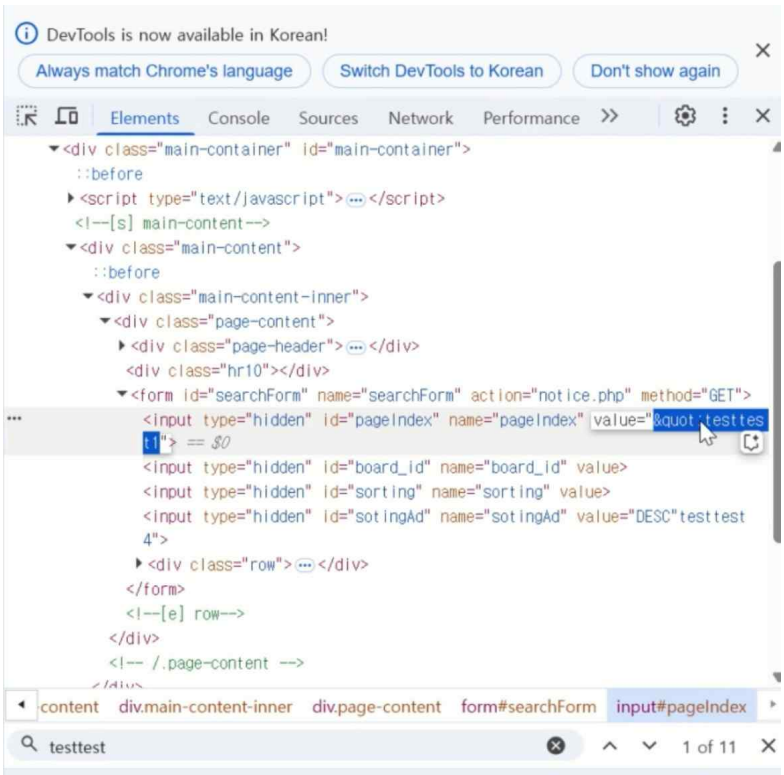
4-2. 필터링 되는지 확인

: 개발자 모드를 이용하여 각각의 문자열 앞에 넣은 "들이 필터링 되는지 확인함

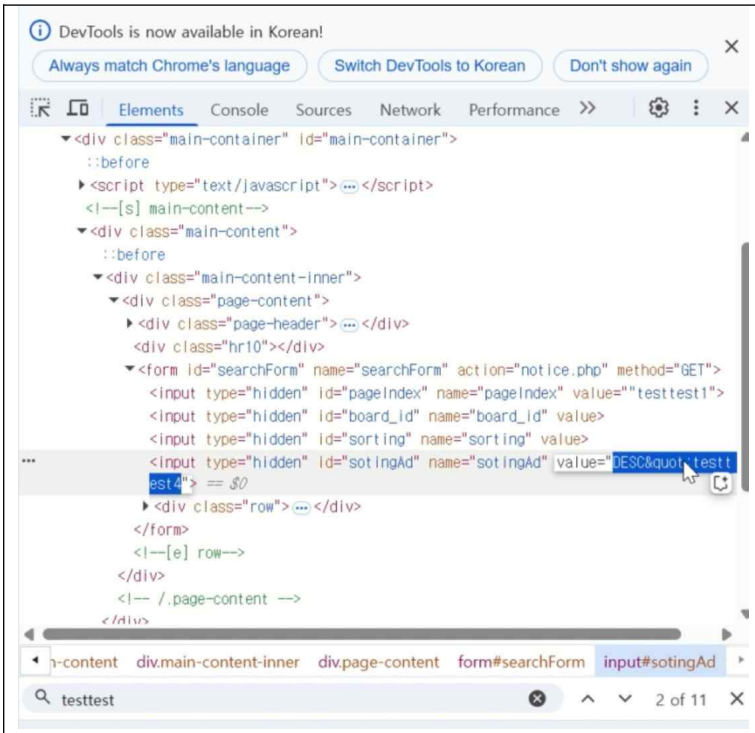
문자열을 더블클릭하면 앞에 &quot가 오는 것을 볼 수 있는데, 이는 "가 필터링됨을 뜻함

계속 확인하다보면 6번째 검색 결과로 나오는 testtest1 앞에 &quot가 오지 않고 testtest1만 있는 것으로(value의 "와 파라미터에 입력한 "가 짝이 맞았기 때문에) 보아 "가 필터링되지 않음을 알 수 있음

- 필터링 O의 경우

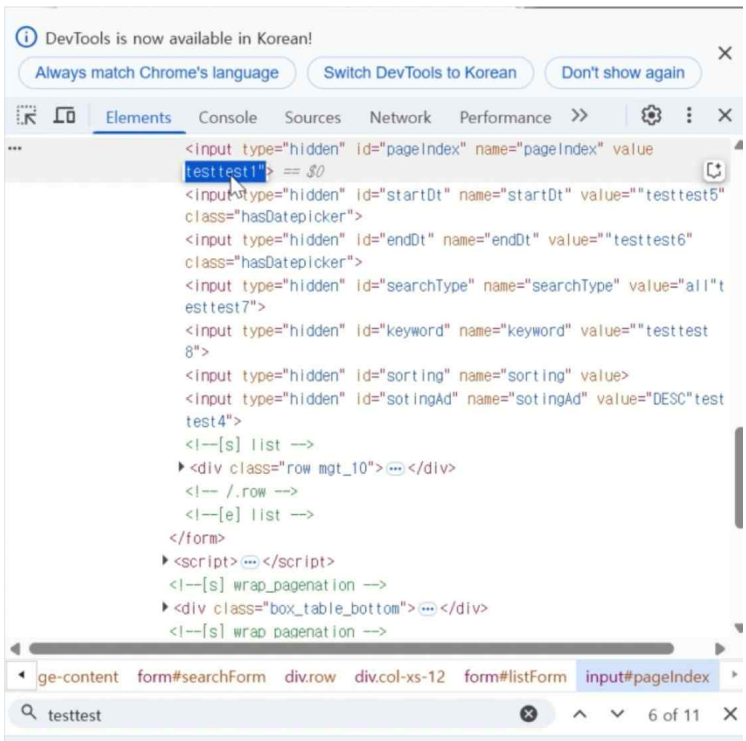


(다음 페이지에 이어서)



- 필터링 X의 경우

: 파라미터에 입력했던 "이 기존 HTML에 있던 "와 짝이 되어 사라짐

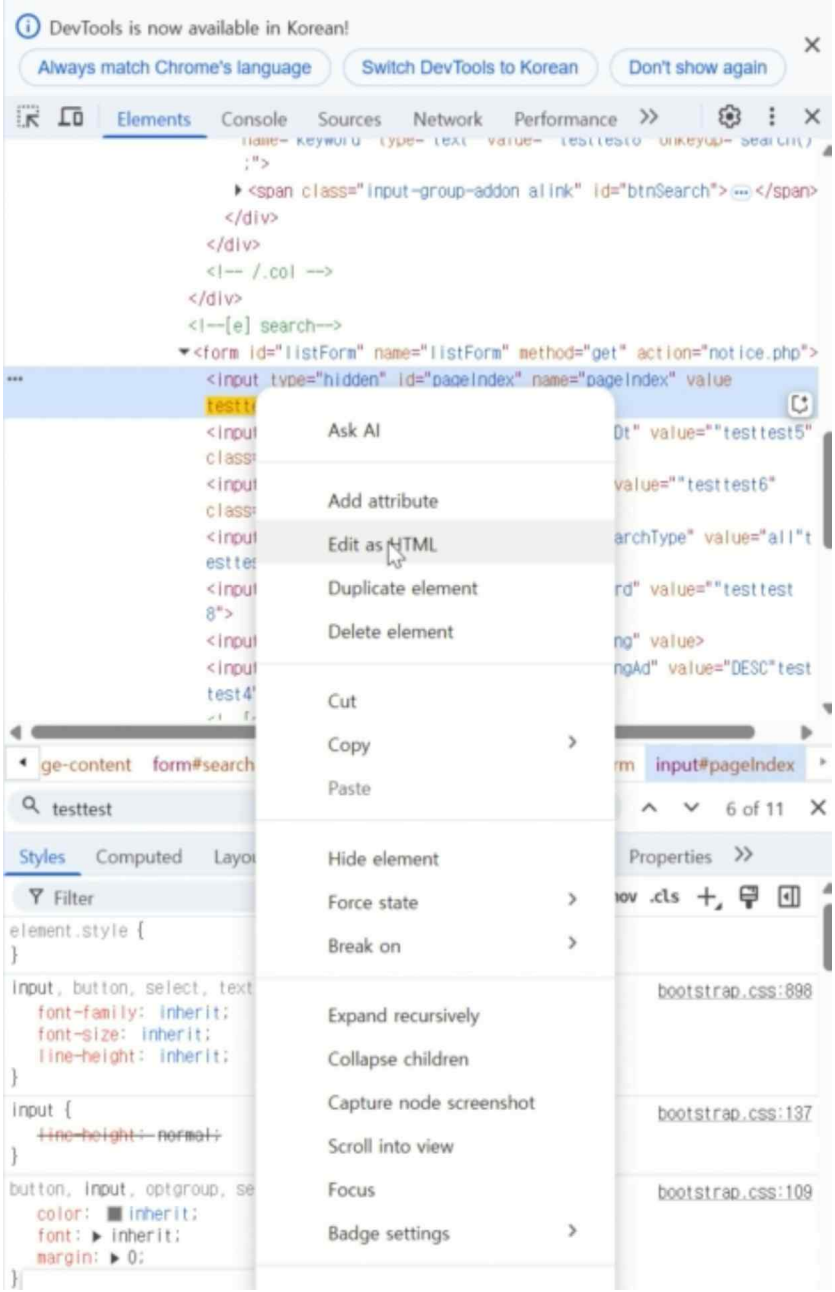


(다음 페이지에 이어서)

Step 5. 스크립트 작성

5-1. Edit as HTML 누르기

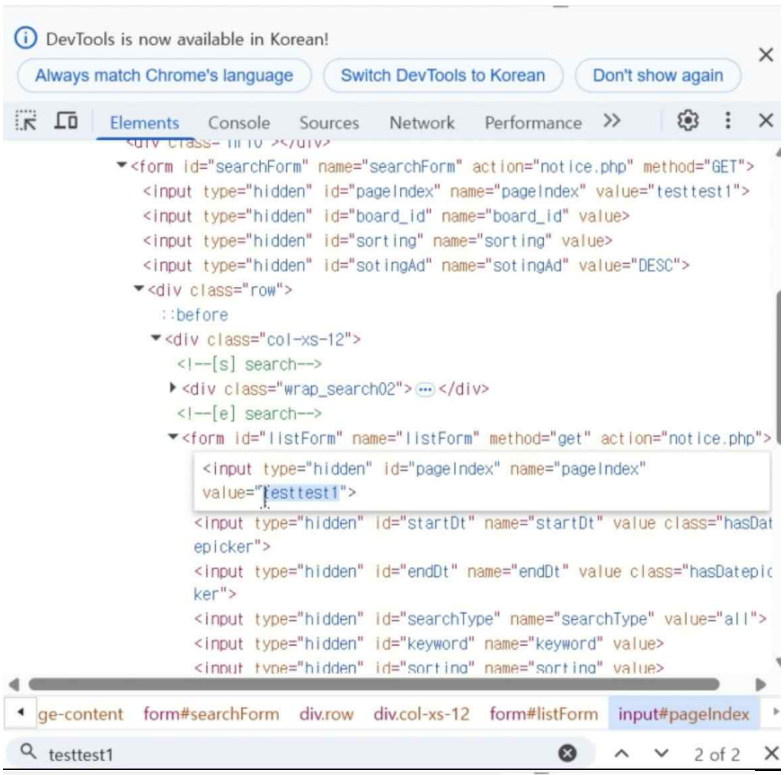
: Step 4에서 찾은 필터링이 되지 않는 곳을 우클릭하여 Edit as HTML을 누름

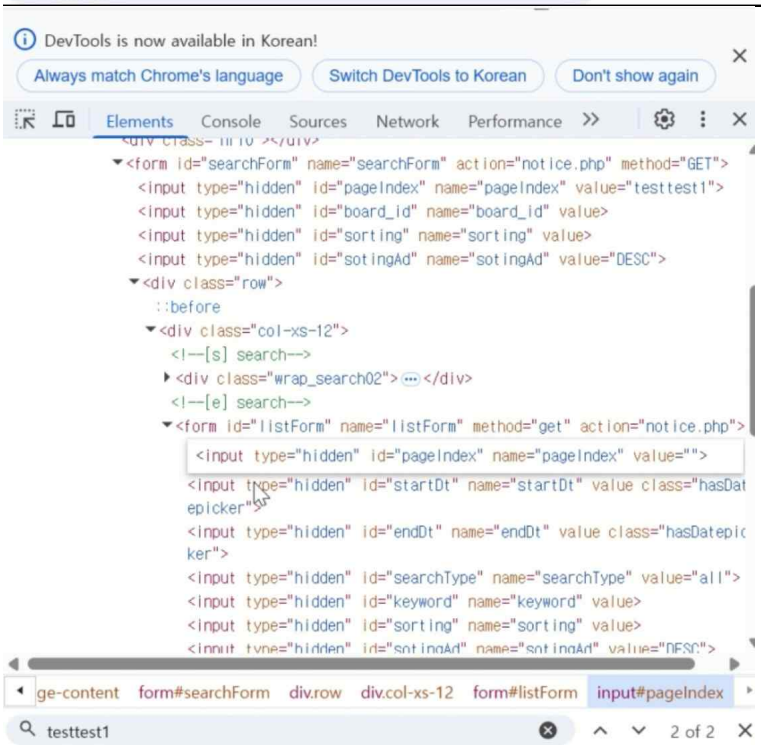


(다음 페이지에 이어서)

5-2. value 값 지우기

: 앞서 입력했던 문자열(testtest1)을 value 값에서 지움





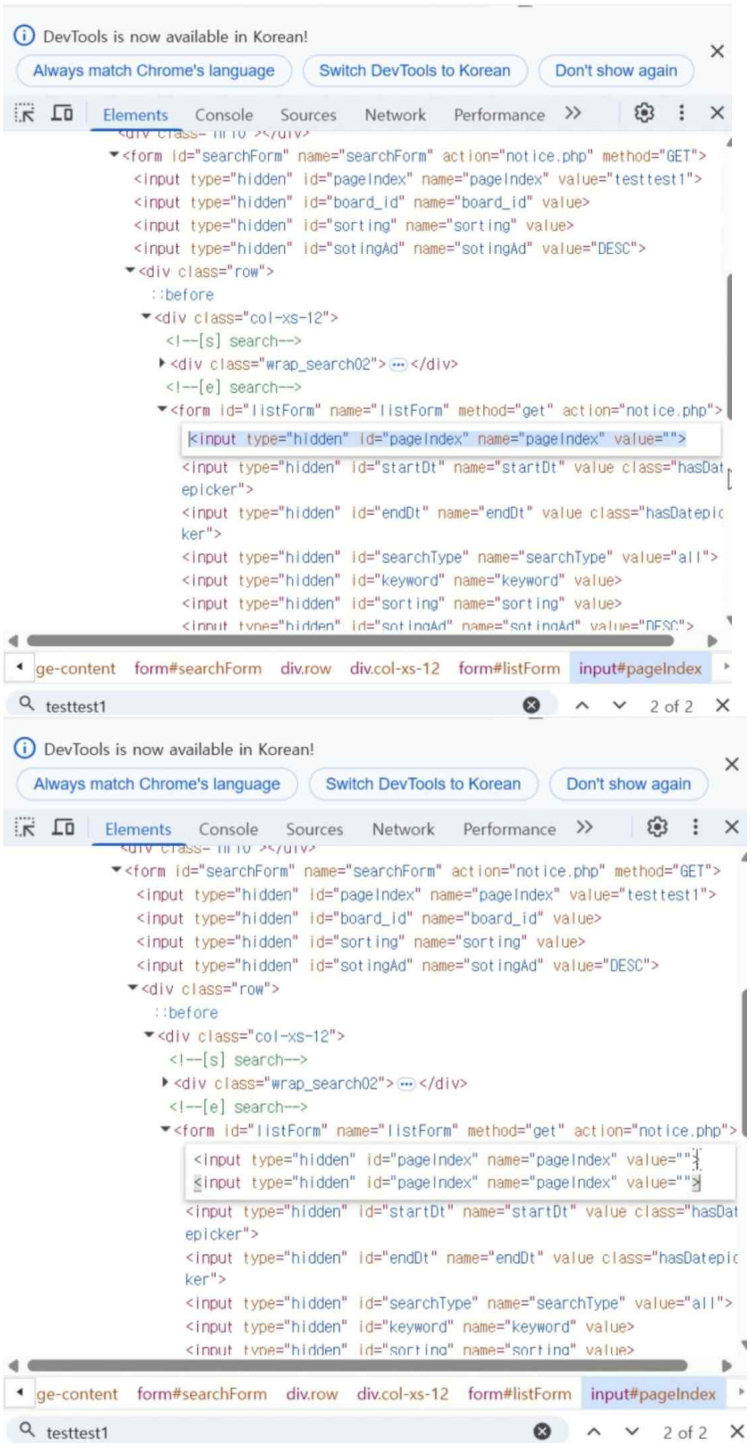
(다음 페이지에 이어서)

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

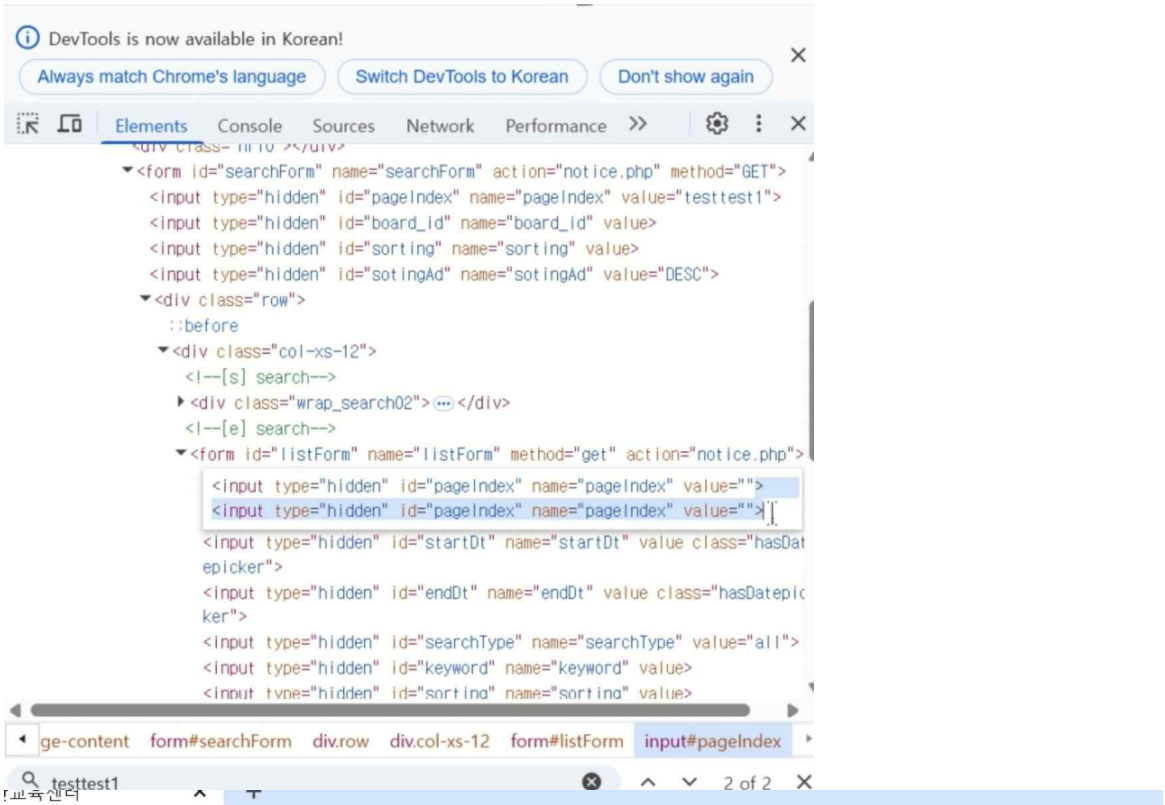
원래의 <input> 태그를 복사하여 바로 뒤에 넣는 방식으로 닫아주는 이유는 브라우저가 구조를 망가뜨리지 않게끔 하기 위해서임

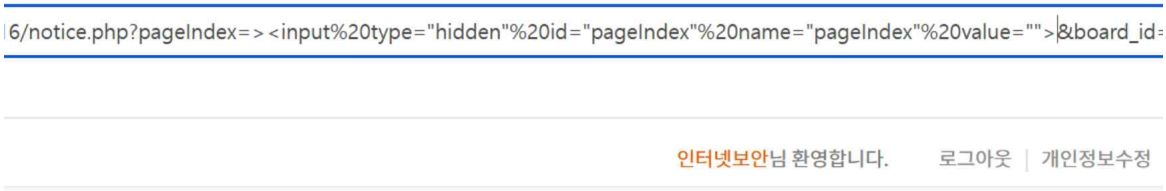


(다음 페이지에 이어서)

5-4. 파라미터에 넣기

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





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 팝업 창 띄우기
</script> : 스크립트 닫기

Step 7. 정답 출력

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

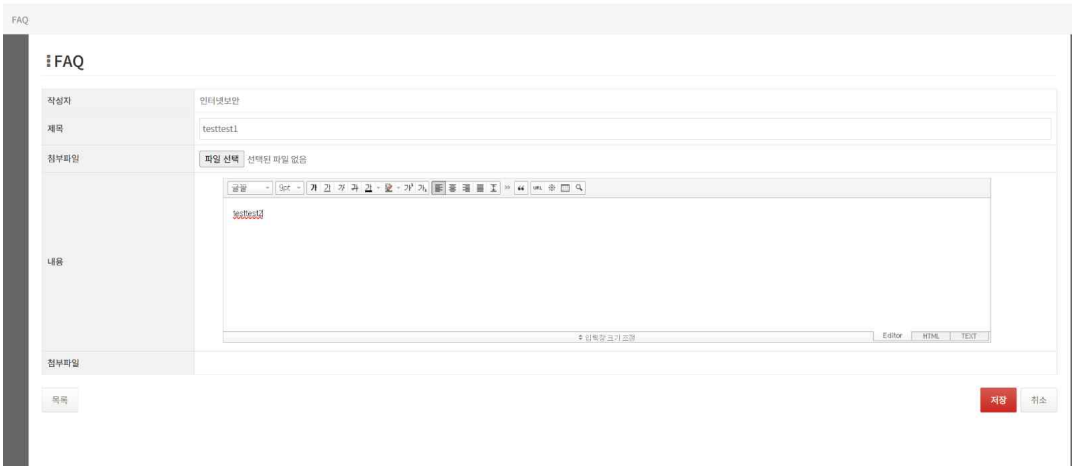
정답 : skinfosec

<저장 XSS>

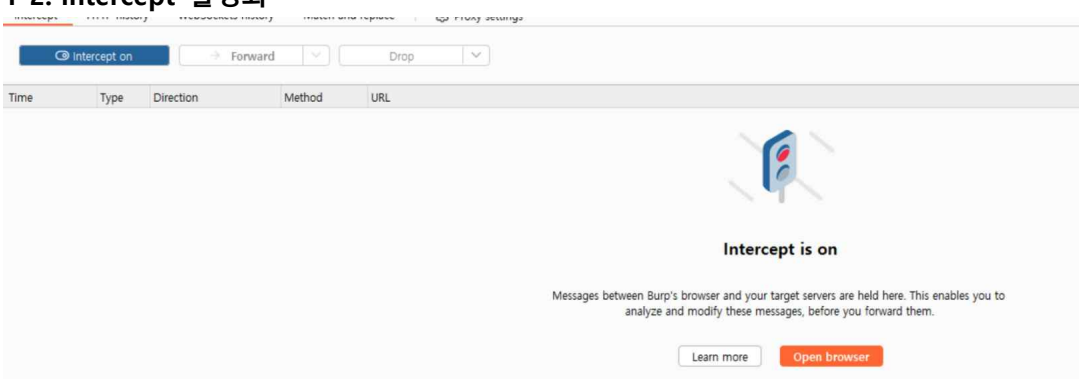
Step 1. 요청 내용 수정

: 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

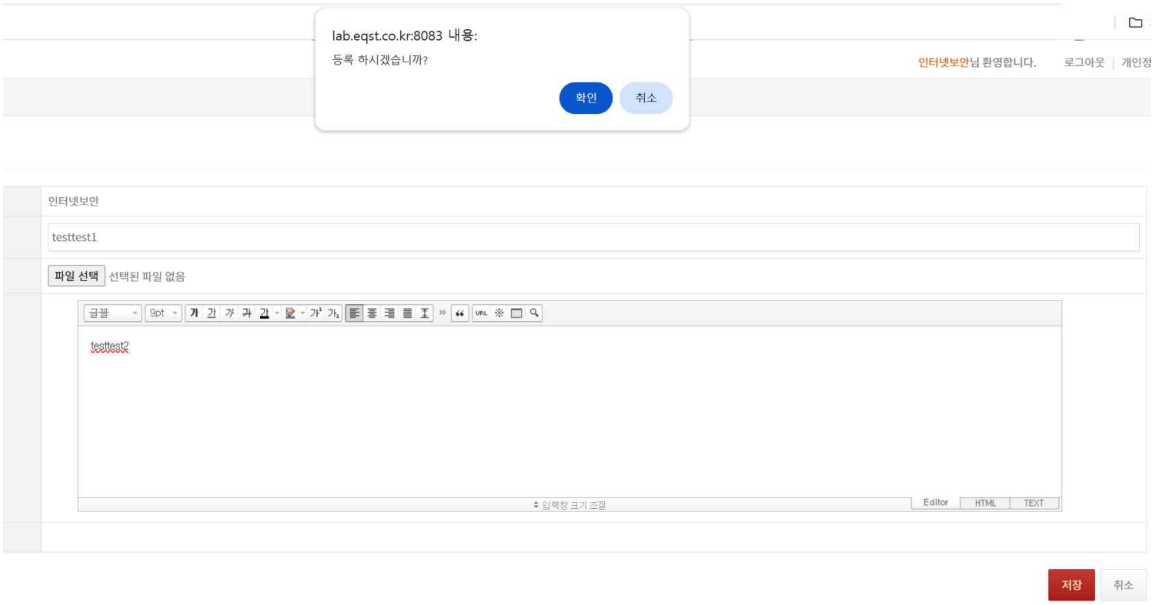


1-2. intercept 활성화

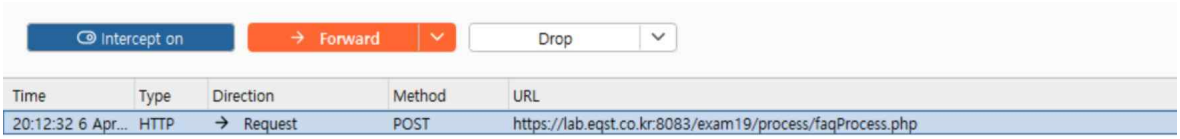


(다음 페이지에 이어서)

1-3. 등록 버튼 클릭



1-4. intercept 목록에서 확인



1-5. 요청 내용 확인

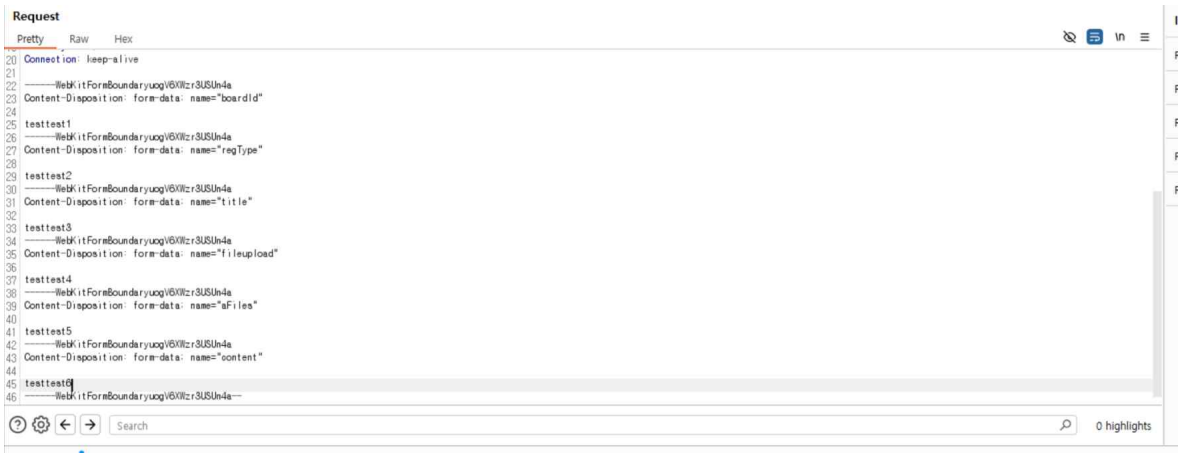
: testtest1은 title 부분, testtest2는 <p>태그에 감싸진 채 content 부분에 나타남을 알 수 있음

이때, <p>태그란 문단을 표현할 때 사용하는 HTML 태그임

(다음 페이지에 이어서)

1-6. 요청 내용 수정

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

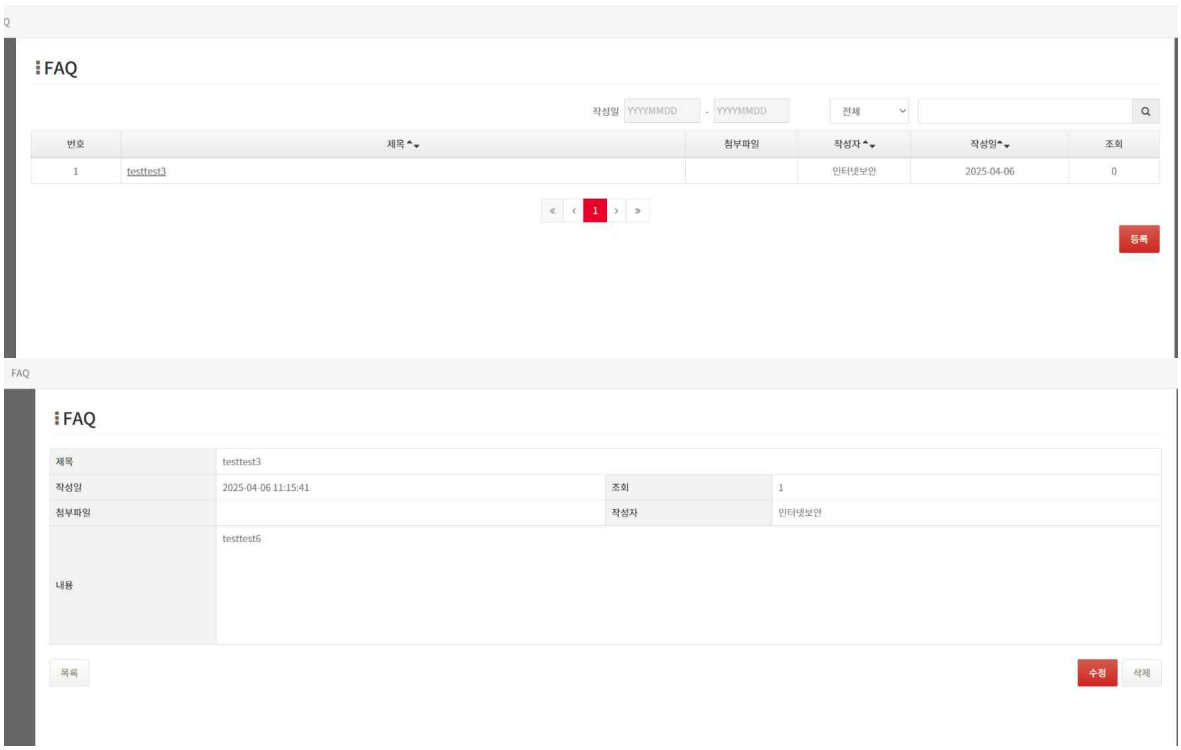


1-7. 게시물 등록

: 1-6의 결과로 게시글이 등록됨

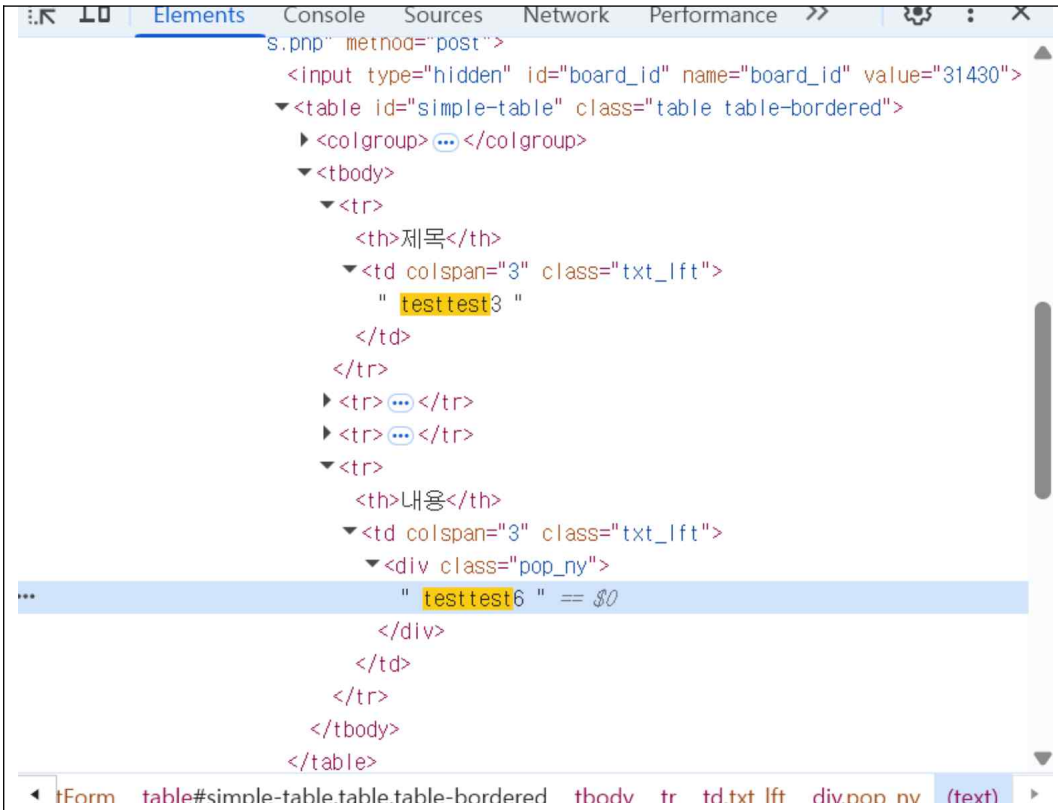
1-8. 게시물 확인

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



Step 2. 입력값 확인

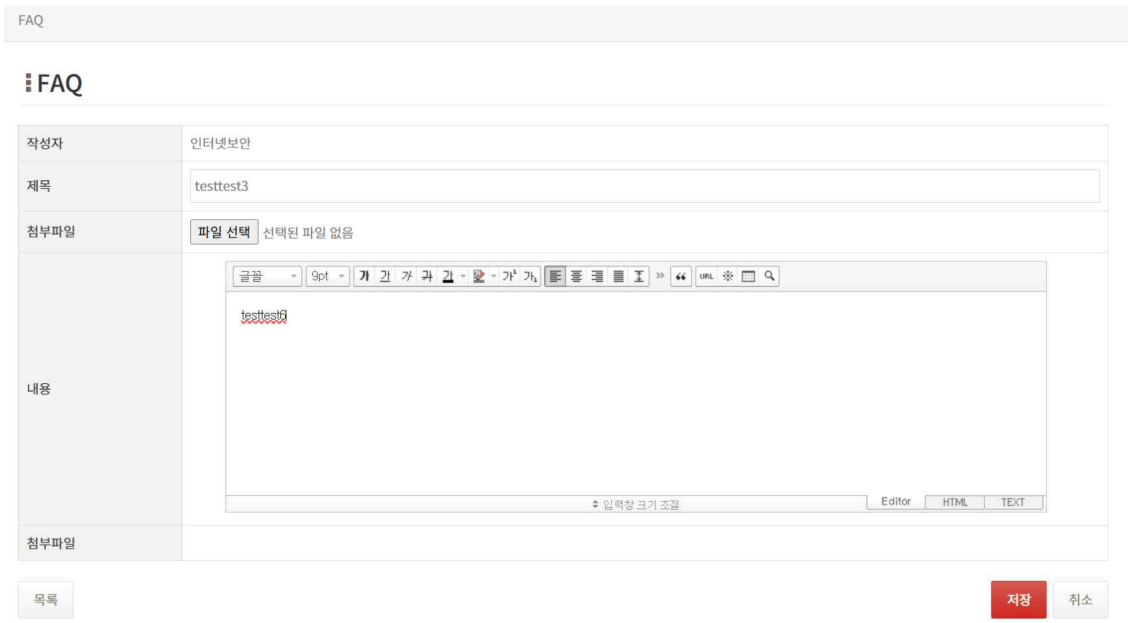
: F12를 눌러 개발자 모드를 실행한 후 Ctrl+F로 testtest를 검색해보면 2개의 결과가 나옴 하나는 title 파라미터에 넣은 testtest3, 나머지 하나는 content 파라미터에 넣은 testtest6임



Step 3. 필터링 확인

3-1. 게시물 내용 작성

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



3-2. 개발자 모드

: 등록된 게시물에 들어가 개발자 모드(F12)를 켬

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

내용 부분은 앞에 열려있는 를 닫은 후, 스크립트를 넣고 다시 를 열어서 뒤에 있는 닫힌 와 짝을 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

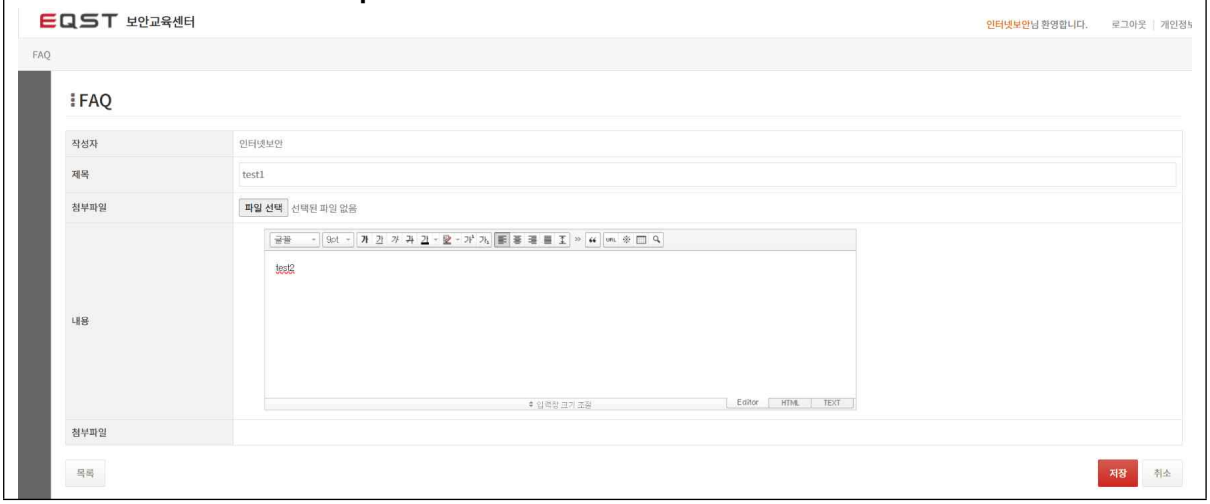
는 division으로 웹페이지를 묶음이나 블록 단위로 나눌 때 사용함

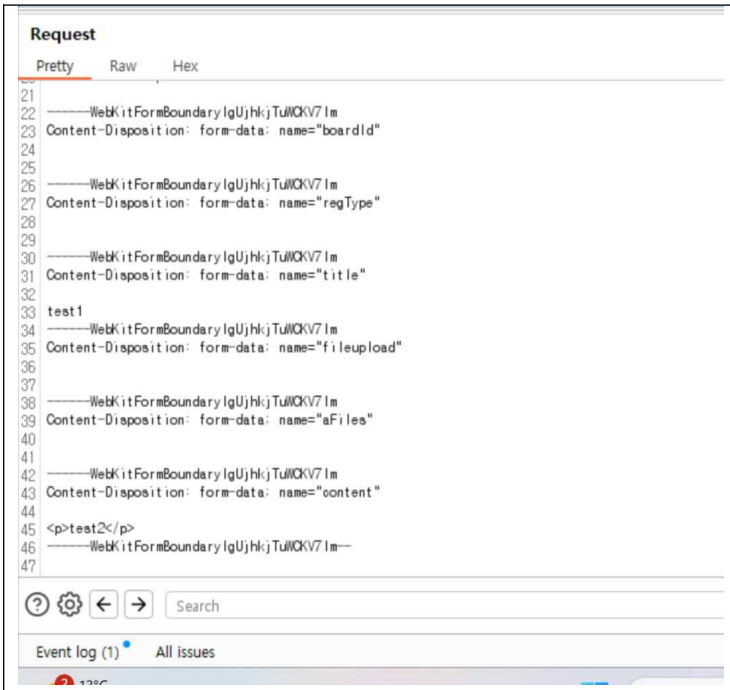
```
<tbody>
  <tr>
    <th>제목</th>
    <td colspan="3" class="txt_lft">
      " testtest3 "
    </td>
  </tr>
  <tr> ... </tr>
  <tr> ... </tr>
  <tr>
    <th>내용</th>
    <td colspan="3" class="txt_lft">
      <div class="pop_ny"> == $0
        <p>testtest6</p>
      </div>
    </td>
```

Step 4. 스크립트 입력

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

4-1. 게시물 등록 & intercept



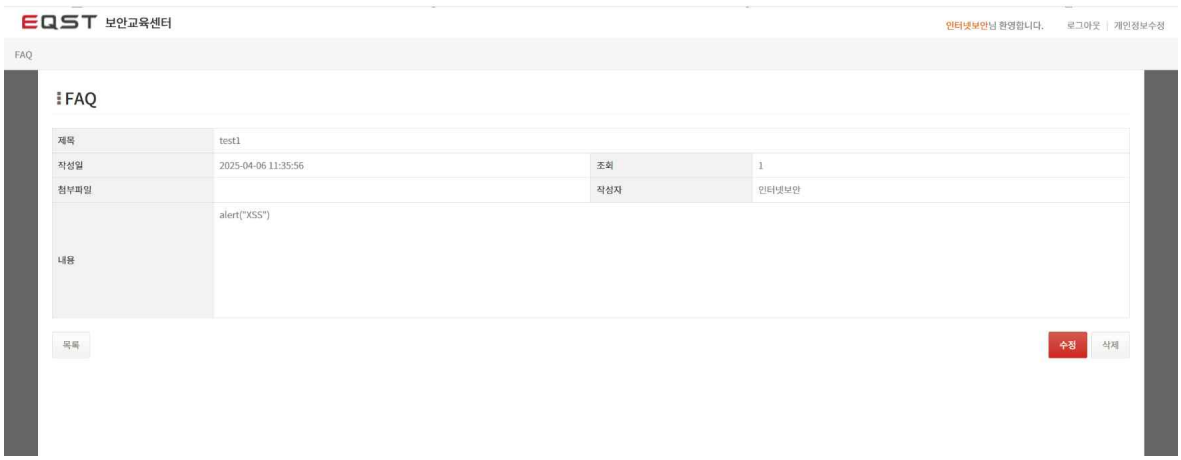


4-2. content에 스크립트 입력

** <script>alert("XSS")</script> **

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

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



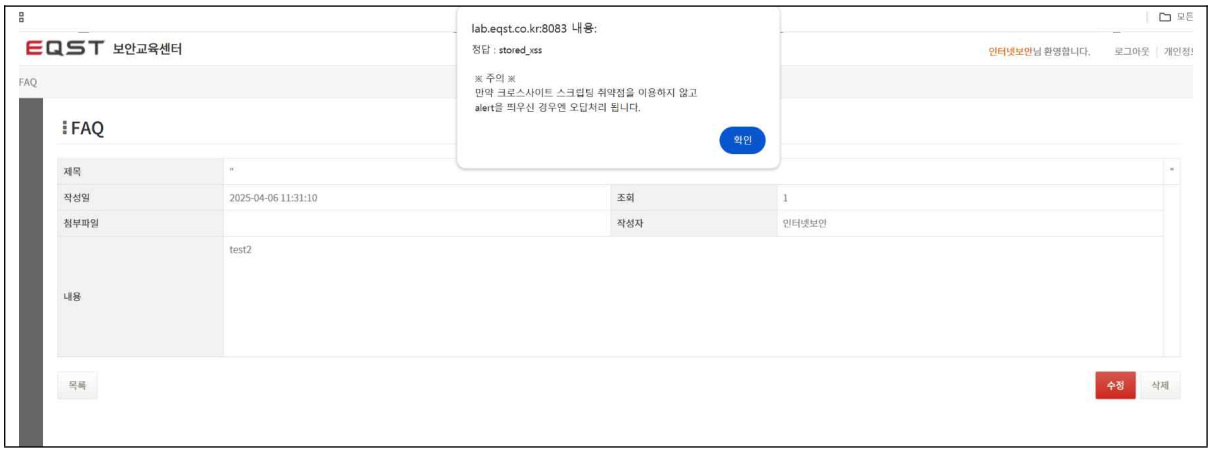
4-3. title에 스크립트 입력

"</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 짝을 맞춰주기 위해 <td>를 넣고, 뒤 "도 닫아줌

제목이 입력한 스크립트로 바뀐 것을 확인할 수 있음

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



<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	웹 페이지 구조와 동작 원리, 보안 취약점이 어떻게 생기고 악용되는지를 느낄 수 있었음 파라미터가 어떻게 서버로 전달되는지, <input>이나 <script> 태그 등이 어떤 역할을 하는지, 자바스크립트가 어떤 원리로 실행되는지에 대해 알 수 있었음 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 이번 실습에서는 태그나 "을 열고 닫는 방식으로 XSS을 찾아냈는데, 이벤트 핸들러를 이용해 찾아내는 방식에 대한 이해는 아직 부족한 상태라 관련된 예제를 풀어보고, 자바스크립트에 대해서도 스스로 공부해봐야겠다는 생각을 함