

웹해킹과 보안

4차과제

학과: 컴퓨터공학과

학번: 18011549

이름: 박태정

□ XSS with WebGoat (100 pts)

- Run WebGoat developer version.

1) Solve the problem ([Cross-Site Scripting (XSS)] - [LAB: Cross Site Scripting] - [Stage 1: Stored XSS]). (10 pts)

1번 문항은 Stored XSS를 사용하는 것이다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

Tom 이 Stored XSS 을 이용하여 Jerry 에게 Edit Profile 페이지에서 공격하려 한다. 예상 되는 시나리오는 Tom 이 입력 필드에 script 를 작성하고 Jerry 는 페이지를 불러오는 것으로 스크립트를 실행시키게 될 것이다.

Tom 으로 로그인하여 Edit Profile 에 접근 한 모습이다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

아마도 Tom 은 Edit Profile 에서 자신의 정보에 script 를 저장하여 Jerry 가 자신의 정보를 로딩 할 때 해당 스크립트를 실행시키도록 할 모양이다. 따라서 Tom 의 이름이 포함되어있는 곳에 script 를 저장하도록 하겠다.

<script>alert(“You are Hacked!”)</script> 를 저장해보도록 하겠다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

<<이렇게 했더니 통과가 되지 않아서 강의에서 처럼 alert cookie 를 사용했습니다.>>>

해당 스크립트를 저장하고, Update Profile 을 하였더니 바로 그 결과가 저장이 되고 브라우저를 통해 User 에게 나타나는 모습을 볼 수 있다.

이제 Tom 의 이름 정보에는 다음과 같은 경고창을 나타내는 script 가 저장되어 (Stored Cross-Site Script) 있는 것이다. Jerry 로 로그인 해서 Tom 의 정보에 접근해보도록 하겠다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

Tom 의 Profile 에는 script 가 저장 되어 있었고, Jerry 가 Tom 의 이름 정보에 접근할 때 script 가 실행되는 것 이다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

초록불이 하나 들어왔다.

그렇다면 우리는 방금 사용했던 <script>alert(“ message “ )</script>, <script>alert(document.cookie)</script> …. 이런 Script 가 무엇인지 알아 보아야할 필요가 있다.

**Script ?**

HTML 의 구성인 Script 요소는 클라이언트 사이드 스크립트와 같은 코드를 HTML 문서 내에 직접 장성하거나 불러올때 사용한다. script 요소느 문서 내 어디에라도 놓일 수 있으며, 그 횟수에 제한이 없다.

문서의 머리에 script 요소를 입력하면 페이지 로딩과 함께 스크립트가 실행 된다. WebGoat 의 소스를 예시로 살펴보면

스크린샷이(가) 표시된 사진

자동 생성된 설명

<head> </head> 사이의 <script>…</script> 가 문서의 로딩과 함께 실행되는 script 코드 인 것이다.

따라서 문서 전체에 걸쳐 기능을 제어해야 하거나 문서 본문을 처리하기 전에 스크립트가 실행될 필요가 있을 때 문서의 머리에 두게 된다.

문서의 </body>-- 직전에 두면 문서의 마지막에 스크립트가 실행 된다. 이는 접속자들의 화면서 HTML 의 구조와 CSS 의 디자인을 먼저 적용하고 렌더링 되기 때문에 문서 로딩의 체감 시간을 줄일 수 있다.

그리기이(가) 표시된 사진

자동 생성된 설명 맨 마지막에 실행 된다는 것.

이 Script 요소에는 여러가지 속성이 존재하는데 하나씩 따져 보기 보다는 어떤 것이 있는지 짚고 넘어 가도록 하겠다.

type 속성: 스크립트의 MIME 타입을 지정한다. javascript 인지, vbscript 인지 브라우저에게 프로그래밍의 언어인지를 알려주는 것이다. HTML5 에서는 javascript가 기본값이다.

defer 속성 : 스크립트의 실행 연기를 지정한다. 브라우저가 src 속성을 만나게 되면 외부 스크립트 파일을 가져오기 위해 HTML 문서의 처리를 잠시 미루는데 이는 유저에게 실행속도의 불만족을 일으킬 수 있다. 따라서 중요하지 않은 스크립트의 경우에는 HTML 의 파싱을 다 끝낸 뒤 스크립트를 처리할 수 있도록 한다. 이때 defer 는 외부 스크립트 파일에 한정되므로 반드시 src 속성이 같이 사용되어야 한다.

async 속성 : 스크립트의 비동기 실행을 지정한다. defer 와 비슷하지만, defer 의 상위 호환이다. defer 의 경우 스크립트를 만나면 스크립트를 배제하고 HTML 문서 파싱에 집중하지만 async 의 경우 스크립트를 만나면 스크립트를 실행하지는 않고 수신과 HTML 파싱을 동시에 수행한다. 그런 다음 외부 스크립트 파일의 수신이 완료되면 스크립트를 실행시킨다.

위 실습에서 사용했던 기본적인 script 실행 인자들은 document , alert 이지만 window.open 등의 적극적인 script 이용방식들도 존재한다.

2) Solve the problem ([Cross-Site Scripting (XSS)] - [LAB: Cross Site Scripting] - [Stage 2: Block Stored XSS using Input Validation]). (10 pts)

스크린샷이(가) 표시된 사진

자동 생성된 설명

2번 문항의 경우 1번과 같다. 하지만 이번에는 같은 방식으로 공격당하지 않는 모습을 보여야 성공하는 것이다.

입력값 검증을 통한 stored XSS 를 막는 방법이다. elipse 를 이용해 코드를 손 봐야할 것 같다. Input 값이 script 에 관한 처리를 전혀 하지않고 서버로 저장되는 것 같은데, 이 부분을 검증을 통해서 서버로 값이 저장되도록 처리하면 될 것 같다.

그렇다면 일단 문제가 생겼던 EditProfile 에 들어간다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

코드를 보니 Edit Profile 에는 관련 정보가 없다. 다음으로 Update Profile 에 접근흘 해보고록 하겠다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

그 중에서 사용자의 입력값에 따라 pasing 하는 함수를 살펴본다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

많은 변수들이 있다. 여기서 이 변수들에게 script 가 포함되었는지, 확인을 해서 Update Profile 을 진행 할 필요가 있다.

다음은 XSS filtering 코드이다. 정규 표현식으로 쓰여 있다.

스크린샷, 조류이(가) 표시된 사진

자동 생성된 설명

이 코드를 문자열의 검증이 필요한 위치에 넣을 필요가 있다. 이 위치는 웹으로부터 getstring 을 이용해 string 변수에 저장하고 Employee 객체로 저장하기 직전에 검증을 하면 될 것 같다.

스크린샷, 조류이(가) 표시된 사진

자동 생성된 설명

필터링 코드를 작성하고, 저장해준다. 그리고 문제에서 말한 것 처럼 David 로 Profile 을 Edit 해서 Eric (manager ) 가 저장된 정보에 접근할 떄 script 가 실행되는지 확인 해보자.

1160 Prescott Rd <script>alert(document.cookie)</script> 앞서 작성한 것 처럼 Street 란에 script 를 작성하고 저장하도록 한다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

초록색 불이 들어왔다. manager id 로 접근하지 않아도 stored xss 가 작동하지 않는 것을 볼 수 있다. 이번에도 정규 표현식에 대해서 알아 볼 필요가 있다.

**정규 표현식**

정규표현식은 String 이나 문자가 준수해야하는 패턴을 표현하기 위한 특정한 표현식이다.

교안에서 제공한 설명이 있지만 방금 사용한 정규표현식을 분석해보자.

스크린샷, 조류이(가) 표시된 사진

자동 생성된 설명

String regex = “[\\s\\w-,]\*”

[] : 대괄호 안의 문자열들은 정규식에서 사용 할 패턴이 되는 문자들로 가능한 문자열의 집합과 일치 시킨다는 것이다. 따라서 공백과, w (word), - 만으로 이루어진 정규식을 사용하여 그것과 관련 된 문자열을 일치시킨다 ( 포함시킨다 )는 것이다.

“ \* ” : 0 번 이상 다음의 패턴들이 등장하면 일치시킨다는 것

따라서 교안에서 제공하는 정규식은 정규식에서 제공하는 문자들이 있다면 포함시켜 일치시킨다는 의미가 되겠다.

pattern 을 을 regex 로 제작하여 validate 인자로 script 포함되어있을 수도 있는 문자열들과, pattern 을 제공하여 유효한 스트링으로 인식한다는 것이다.

나이프이(가) 표시된 사진

자동 생성된 설명

3) Solve the problem ([Cross-Site Scripting (XSS)] - [LAB: Cross Site Scripting] - [Stage 3: Stored XSS Revisited]). (20 pts)

스크린샷이(가) 표시된 사진

자동 생성된 설명

문제에서 요구하는 것은 David profile 에서 입력값 검증이 아닌 저장값 검증에 관한 문제를 보여주고 4번에서 이미 저장 된 script 를 검증하라는 문제인 것 같다. … 4번을 위한 문제인 것 같다.

4) Solve the problem ([Cross-Site Scripting (XSS)] - [LAB: Cross Site Scripting] - [Stage 4: Block Stored XSS using Output Encoding]). (20 pts)

스크린샷이(가) 표시된 사진

자동 생성된 설명

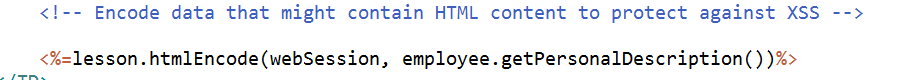
4번에서는 script 가 서버에 저장 되어 있고, 그것을 불러올 때 발생할 수 있는 XSS 이다. 따라서 2번에서는 입력 값을 검증 한 것이었고, 이번에는 출력 값을 검증 해야 할 필요가 있다는 것.

js 를 검증해보자. eclipse 의 javascript 을 확인해본다.

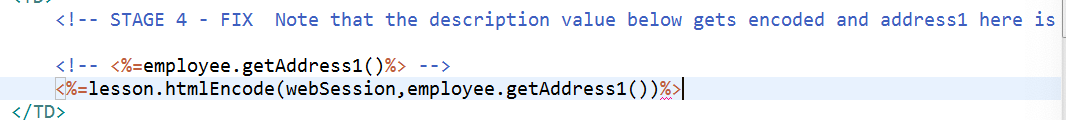
스크린샷이(가) 표시된 사진

자동 생성된 설명

다음과 같이 변수에 get~() 함수를 사용해서 값을 받는 것으로 보여진다. 저 값들을 검증하면 될 것 같다.



받아들인 정보들을 Encode 를 하는 것을 보여진다. script의 특징인 꺽쇠(<,>) 를 encoding 하면 script가 실행되지 않을 것 같다.



함수 그대로 가져다 script 가 포함 된 문장에 대헤 인자만 교체 해준다.

사실 모든 get 함수에 encoding 을 해주어야 하는데, 우리는 이미 어디서 script 가 포함 된 것인지 알고 있으므로 , Address 에서만 인코딩을 해준다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

script 가 실행되지 않고 문자열로 표현되는 것을 볼 수 있다. 새로 고침을 하니 4단계도 초록 불이 들어왔다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

5) Solve the problem ([Cross-Site Scripting (XSS)] - [LAB: Cross Site Scripting] - [Stage 5: Reflected XSS]). (20 pts)

스크린샷이(가) 표시된 사진

자동 생성된 설명

5번 문제까지 순항 중이다.

5번 문항은 Search Staff page 에 script 를 포함시켜 reflect xss 를 발생시키는 것이다. 다른 employee 들이 그 링크를 로드할 경우 xss 가 발생함을 보인다.

Serach Staff page 의 경우 xss 에 대한 방어가 없는 듯 하다. 따라서 아무 계정으로 로그인하여 Search Staff 로 접근해보겠다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

search staff 로 접근해서 search 란에 script 를 작성해보겠다.

<script>alert("Your are Vulnerable!")</script> 스크립트를 작성해서 find 요청을 보내면

스크린샷이(가) 표시된 사진

자동 생성된 설명

script 가 실행이 되고 , 5단계가 성공한것을 볼 수 있다.

6) Solve the problem ([Cross-Site Scripting (XSS)] - [LAB: Cross Site Scripting] - [Stage 6: Block Reflected XSS]). (20 pts) 스크린샷이(가) 표시된 사진

자동 생성된 설명

6번은 5번 처럼 url 에 포함 된 , 그러니까 입력값을 저장하거나 저장 된 값을 불러오는 것이 아니라 html 에 직접 집어 넣은 xss 가 reflected 되는 것을 막는 것이다.

위에서 한 것 처럼 eclipse 에서 serach staff js 파일을 찾아보도록 하겠다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

이 자바 파일에서는 입력으로 받은 name 을 Findprofile action 에 그대로 넘겨준다. 다시 Find profile java 파일을 찾아보도록 하겠다. 스크린샷이(가) 표시된 사진

자동 생성된 설명

잘못찾았다. introduction 이였다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

문제의 find profile 을 작동하는 함수를 찾았다. 여기서 get name 함수를 찾으면 다음과 같다.

스크린샷, 조류이(가) 표시된 사진

자동 생성된 설명  
위의 introduction 에서 나온 정규식을 여기서 사용하면 된다.

스크린샷, 조류이(가) 표시된 사진

자동 생성된 설명

BackUp 이 있지만 그래도 주석을 사용해서 기존의 코드들을 살려준다.

저장한 후 다시 reflect XSS 를 실행해보도록 하겠다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

validation 이 제대로 작동한다. 6번 문제도 성공했다. 초록 불이 많이 들어왔다.

한줄로 나온 이 정규식을 살펴보지 않을 수 없겠다.



이 정규식이 위에서 사용한



이 정규식과 다른 게 없다.

[,] 를 이용한 원하는 문자를 일치시킨, 패턴으로 일치시키는 것 이다. 그리고 그 패턴 객체를 이용한 함수인 xx.get(name) 을 사용한 것이다. 그리고 validate 의 함수 인자로 기존의 name 을 그대로 get 한 문자열과 설정 한 패턴을 인자로 넘겨주어 검증한 값을 반환 하는 것 이다.

스크린샷이(가) 표시된 사진

자동 생성된 설명