Weak Session IDs & Xss(DOM)

웹 모의해킹 실습 5주차

배준호

CONTENTS

목차

1. Weak Session IDs

- Session IDs: 웹 응용 프로그램에서 사용자의 세션을 식별하기 위해 사용되는 고유한 식별자

- Weak Session IDs는 예측하기 쉬운 세션 ID이다.
- 사용자의 세션 ID 알아내어 쿠키나 세션ID를 조작하여 세션 하이재킹, 세션 위조 , 세션 탈취 등 보안 위험이 발생할 수 있다.

2. Weak Session IDs 실습

Weak Session IDs Source

vulnerabilities/weak_id/source/low.php

```php
<?php

$html = "";

if ($_SERVER['REQUEST_METHOD'] == "POST") {
    if (!isset ($_SESSION['last_session_id'])) {
        $_SESSION['last_session_id'] = 0;
    }
    $_SESSION['last_session_id']++;
    $cookie_value = $_SESSION['last_session_id'];
    setcookie("dvwaSession", $cookie_value);
}
?>
```

- Generate 클릭할 때 마다
- last_session_id 값이 1씩 증가하는 것을 확인

Vulnerability: Weak Session IDs

This page will set a new cookie called dvwaSession each time the button is clicked.

Generate

- Generate 클릭

2. Weak Session IDs 실습

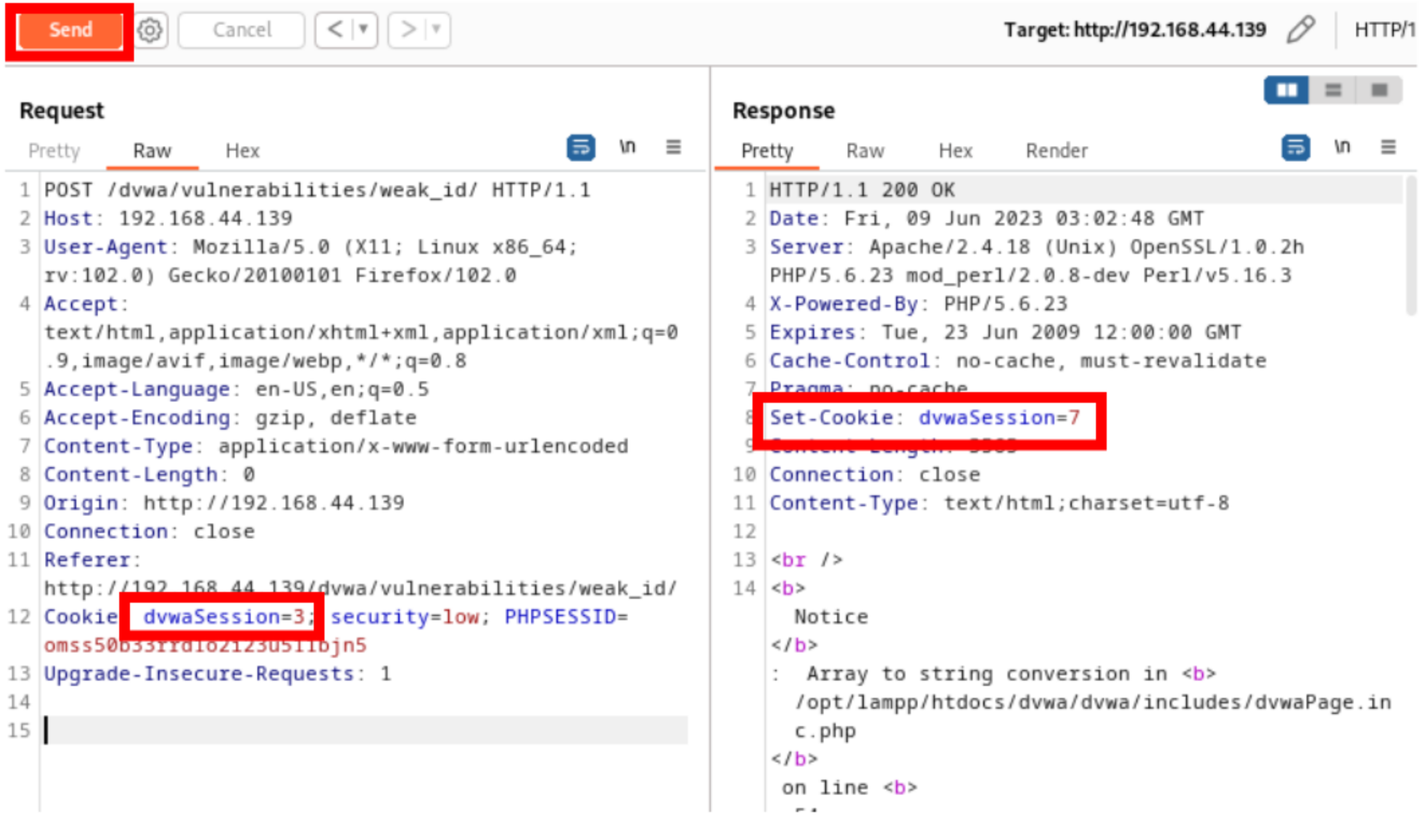
Name	Value	Domain	Path	Expires / Max-Age	Size
dvwaSession	1	192.168.44.139	/dvwa/vulnerabilitie...	Session	12
PHPSESSID	omss50b33rrd1o2i23u511bjn5	192.168.44.139	/	Session	35
security	low	192.168.44.139	/	Session	11

Name	Value	Domain	Path	Expires / Max-Age	Size
dvwaSession	2	192.168.44.139	/dvwa/vulnerabilitie...	Session	12
PHPSESSID	omss50b33rrd1o2i23u511bjn5	192.168.44.139	/	Session	35
security	low	192.168.44.139	/	Session	11

Name	Value	Domain	Path	Expires / Max-Age	Size
dvwaSession	3	192.168.44.139	/dvwa/vulnerabilitie...	Session	12
PHPSESSID	omss50b33rrd1o2i23u511bjn5	192.168.44.139	/	Session	35
security	low	192.168.44.139	/	Session	11

- 개발자도구(F12) - Stroage 확인
- Generate 클릭 시
- dvwaSession 값이 1씩 증가하는 것을 확인
  => 간단한 패턴으로 세션 값을 추측하여 보안이 취약하다

2. Weak Session IDs 실습



- 또한 Burp Suite를 통해 세션값이 증가하는 것을 확인할 수 있다.

- Send를 누를 때마다 Session 값이1씩 증가하는 것을확인

-> 해당 화면은 7번 클릭

2. Weak Session IDs 대응방안

- 세션 id를 공격자가 추측하기 어렵게 랜덤 생성 알고리즘 사용

- 프레임 워크를 사용하여 쿠키 서명을 한다
  - ex) hashlib를 이용한 비밀키 사용

3. Xss(DOM)

- XSS DOM - 악의적인 사용자가 웹 애플리케이션에서 발생할 수 있는 취약점을 이용하여 자바스크립트 코드를 삽입하고 실행하는 공격.

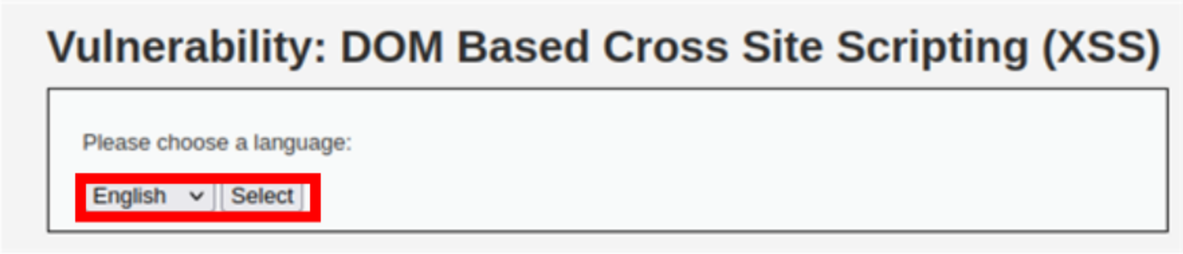
- DOM 기반 XSS는 서버 측에서 발생하는 것이 아니라 클라이언트 측에서 발생하는 취약점.

3. Xss(DOM)

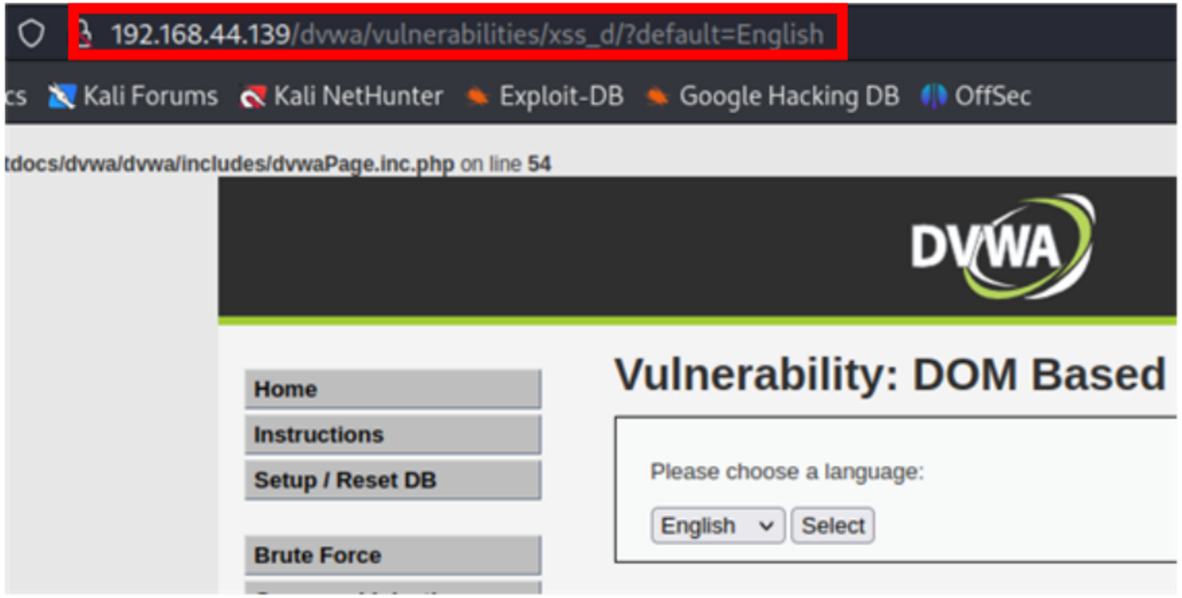
XSS(DOM) 공격 상황

- URL 파라미터: 악의적인 사용자가 웹 페이지로 전달되는 URL의 파라미터에 조작된 스크립트를 삽입할 수 있다.
- HTML 요소의 값: 웹 페이지의 입력 필드나 기타 HTML 요소의 값에 악의적인 스크립트를 삽입할 수 있다.
- 이벤트 핸들러: 웹 페이지에서 이벤트 핸들러를 등록할 때, 악의적인 사용자가 조작된 스크립트를 등록할 수 있다.

4. Xss(DOM) 실습



- English를 선택후 Select 클릭 시



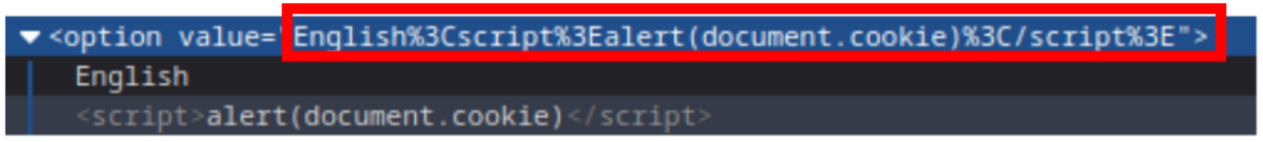
- English를 선택후 Select 클릭 시

http://192.168.44.139/dvwa/vulnerabilities/xss_d/

->
http://192.168.44.139/dvwa/vulnerabilities/xss_d/?default=English

- default 파라미터 값에 English가 추가된 것을 확인

4. Xss(DOM) 실습



- 해당 파라미터 부분에 JavaScript 구문 삽입
- http://192.168.44.139/dvwa/vulnerabilities/xss_d/default=<script>alert(document.cookie)</script> 실행
- alert가 정상적으로 실행된 것을 보아 공격이 성공한 것을 확인
  ->document.cookie를 통해 쿠키 값을 알 수 있다.
- 또한 개발자 도구(F12) - Inspector를 통해 해당 스크립트가 정상적으로 실행된것을 확인

4. Xss(DOM) 대응방안

- 입력 값의 검증과 이스케이프: 사용자 입력 값을 검증하고 적절한 이스케이프 처리를 수행하여 악의적인 스크립트의 삽입을 방지한다.
- 안전한 DOM 조작: 웹 페이지에서 동적으로 DOM을 조작할 때, 안전한 API 및 메소드를 사용하고 신뢰할 수 있는 데이터만을 사용한다.
- Content Security Policy (CSP): CSP를 사용하여 웹 페이지에서 로드되는 리소스의 출처를 제한하고, 스크립트 실행 등 보안 정책을 설정한다.
- 웹 애플리케이션의 보안 강화: 웹 애플리케이션의 취약점을 검사하고, 보안 패치 및 업데이트를 수행하여 XSS 공격을 예방한다.