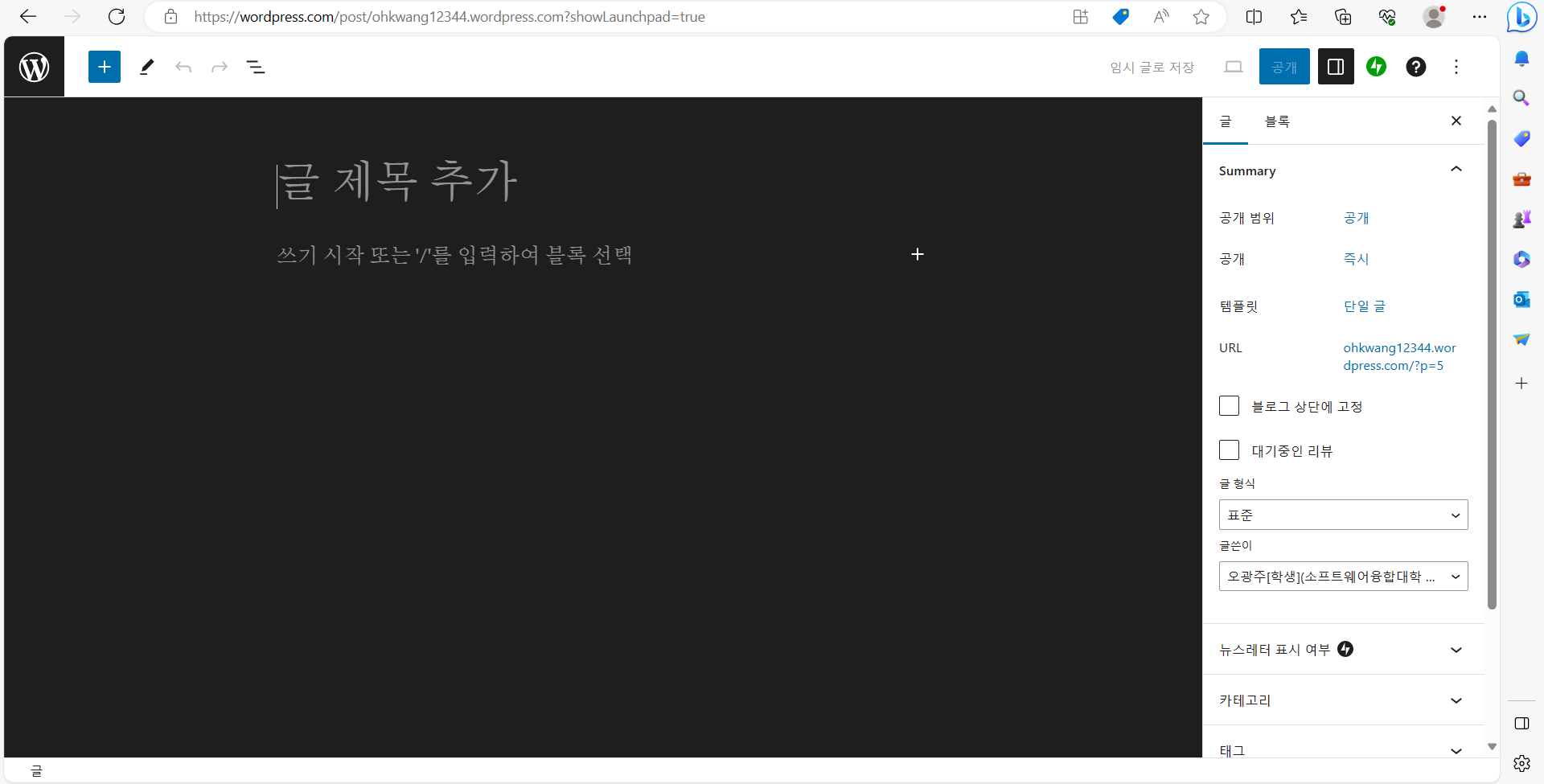
웹 개발 프레임워크의 보안 취약성 개발

**1. 연구배경**

**1.1 웹 개발 프레임워크와 워드프레스 소개**

과거 인터넷에선 정적인 이미지와 텍스트만 있으면 웹을 만들 수 있었다. 즉, HTML만 있으면 충분했고, 자바스크립트를 이용해 동적으로 구성되는 부분을 구현할 수 있었다. 이처럼 단순한 방법으로 웹을 만들 수 있었지만 현대 인터넷에선 금융, 전자 상거래 등 웹을 구현하는 기술의 난이도가 높아졌다. 높아진 난이도를 최소화하기 위해 웹 프레임워크를 만들기 시작했다. 웹 프레임워크란 개발자마다 다른 소스코드나 결과물을 구조화된 스크립트를 통해 개발자의 스크립트 패턴을 정형화하여 필요한 노력을 최소화한 도구이다. 프레임워크는 코드가 체계적이므로 유지보수가 가능하고, 확장성, 코드의 재사용성, 개발 생산성이 높다. 현재 수많은 웹 개발 프레임워크가 존재한다. ‘워드프레스(WordPress)’는 오픈 소스를 기반으로 한 프로그램 언어 PHP, HTML, JS, CSS와 DB가 결합되어 작성된 설치형 블로그, 컨텐츠 관리 시스템으로써 사용자가 전문적인 지식이 없어도 웹사이트에서 컨텐츠를 관리하고 수정할 수 있도록 도와주는 시스템이다. 전 세계 웹사이트의 약 43%가 워드프레스로 만들어져있을만큼 웹사이트에서 워드프레스를 많이 이용하고있다.



워드프레스 글 작성 화면

다음은 워드프레스의 장점이다.

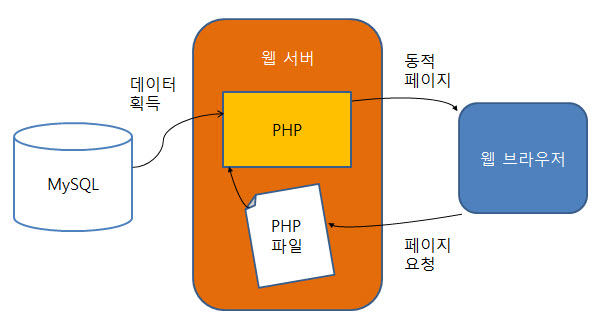
1. 워드프레스는 코딩에 익숙하지 않아도 웹 사이트를 제작할 수 있다.

2. 워드프레스에는 수많은 테마와 플러그인이 존재한다.

3. 오픈 소스이기 때문에 도메인과 호스팅 비용만 지불하면 웹사이트를 제작할 수 있다.

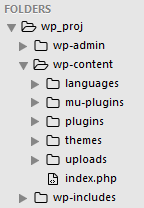
**1.2 워드프레스의 구성과 구조**

워드프레스는 플러그인, 테마, 코어 세가지로 구성할 수 있다. 첫째, 플러그인이란 웹사이트에 어떠한 기능을 추가하거나 확장하고 싶을 때 쓰인다. 워드프레스는 오픈 소스이기 때문에 다양한 플러그인을 제공하고 있고 약 60000개의 플러그인이 존재한다. 각 플러그인마다 사용자의 요구에 맞는 기능을 제공한다. 둘째, 테마는 웹사이트에서 레이아웃이나 디자인을 만들 때 쓰인다. 워드프레스에는 약 7000개의 테마가 존재한다. 테마에서는 콘텐츠를 가져와 해당 테마의 디자인에 표시한다. 단순한 큰 틀에서의 디자인이 아닌 글꼴, 색, 메뉴의 위치 등 디테일한 디자인까지 테마를 통해 정의할 수 있다. front-page.php, header.php, footer.php, page.php등 많은 php들이 모여 테마를 이루고 있다. 플러그인와 테마는 PHP, CSS, HTML, JS로 구성되어있기 때문에, 혹여나 사용자가 기능이 구현된 PHP코드를 찾아 수정할 수 있다. 테마 수정은 차일드 테마(부모 테마의 파생 버전을 만드는 테마 방식)를 통해서 할 수 있다.



워드프레스의 동작 원리

워드프레스의 작동 구조이다. 첫째, 사용자가 워드프레스 기반 웹사이트 URL에 들어가 HTTP 요청을 한다. 둘째, HTTP서버는 요청이 들어오면 요청을 받아 워드프레스 코어의 PHP모듈에 동적인 HTML페이지를 요청한다. 셋째, 코어는 요청을 받아 DB에서 데이터를 이용해 활성화된플러그인과 테마로부터 동적인 HTML 페이지를 만들고 HTTP서버에게 보낸다. 넷째, HTTP서버는 만들어진 동적인 HTML페이지를 사용자에게 보낸다.



워드프레스의 디렉토리 구조

워드프레스는 wp-admin, wp-content, wp-includes 등 여러 디렉토리 하위에 PHP파일, HTML, CSS, JS 등이 존재한다. wp-admin은 관리자페이지를 관리하기 위한 디렉토리이다. wp-admin에 있는 admin-ajax.php 파일을 이용해 워드프레스의 모든 AJAX요청을 처리할 수 있다(AJAX - JS를 이용해 서버에 데이터를 비동기 방식으로 요청). wp-content은 워드프레스에 설치된 플러그인과 테마의 디렉토리, 사이트에 업로드하는 모든 미디어 파일이 포함된다. wp-includes는 function.php등 동작을 제어하는 중요한 디렉토리이다. wp-admin과 wp-includes에는 워드프레스 코어에 해당하는 디렉토리가 위치한다.

**2. 본론**

**2.1 웹 개발 프레임워크의 취약점**

웹 개발 프레임워크의 종류는 많이 존재하고 발전되었지만, 아직 많은 취약점이 존재한다. 최근 웹 애플리케이션은 단순히 인터넷만의 역할만 하는 것이 아닌 서버와 프론트엔드와 백엔드의 혼합체이다. 따라서 웹 애플리케이션에 대한 공격은 큰 피해로 이어진다. 워드프레스는 오픈 소스 플랫폼으로, 테마와 플러그인을 누구나 만들고 수정할 수 있다. 워드프레스의 개방성은 많은 사람들에게 편리함을 주지만, 동시에 보안적 측면에서 문제가 제기된다. 워드프레스의 보안 취약점은 많지만 대표적으로 다음과 같다.

- CSRF

- SQL 삽입

- 멀웨어

- 교차 사이트 스크립팅

- 오픈소스 콘텐츠 관리 시스템

- 디도스 공격

- 구식 워드프레스 및 PHP

이 중, SQL 삽입, 교차 사이트 스크립팅, CSRF는 보안 취약점 중에서 가장 많은 비중을 차지한다.

1. 교차 사이트 스크립팅

교차 사이트 스크립팅(Cross Site Scripting)은 XSS라고 불리며, 관리자가 아닌 권한이 없는 사용자가 웹 사이트에 스크립트를 삽입하는 공격 기법이다. 따라서 공격자는 개발자가 고려하지 않는 기능을 수행하여 사용자의 정보를 탈취할 수 있다. Stored XSS는 스크립트가 서버에 저장되어 실행되는 방식이고, Reflected XSS는 HTTP의 GET방식에 스크립트를 넣어 그 즉시 스크립트를 실행시키는 공격이다. 대부분 자바스크립트를 통해 공격을 한다. 해커의 홈페이지에 PHP파일을 만들고, 공격할 사이트에 스크립트를 심어 사용자가 스크립트가 심어진 페이지를 볼 경우 쿠키나 세션이 해커의 홈페이지에 넘어간다.

예시 1) href 링크 태그 <a href = “javascript : alert(’XSS’)”>XSS</a> javascript : 와 같은 링크는 스크립트를 실행시킨다.

예시 2) script 태그 <script>alert(’XSS’);</script> 스크립트 태그의 스크립트를 실행시킨다.

2. CSRF

CSRF는 Cross Site Request Forgery로, 사용자가 자신의 의지와는 다르게 공격자가 원하는 행위를 웹사이트에서 요청하게하는 공격이다. 즉, 공격자는 희생자의 권한을 도용하여 중요 기능을 실행할 수 있게된다. 공격자의 요청이 사용자의 요청인 것처럼 속이는 공격 방식이다. CSRF 공격이 이루어지기 위해선 다음 두 조건을 만족해야한다.

- 위조 요청을 전송하는 서비스에 희생자가 로그인한 상태여야 한다.

- 희생자가 공격자가 만든 피싱 사이트에 접속해야한다.

공격자가 특정 사이트에서 공격자에게 이득이 되는 어떠한 행동을 위조하고, 위조한 요청은 하이퍼링크 등에 삽입해 웹사이트 자체에 삽입한다. 사용자가 웹사이트의 링크를 누르면 공격자에게 이득이 되는 행동을 요청하고 요청한대로 수행된다.

예시)

<form th:action = “http://aaaaa.aaaaa.com” method = “post”>

<input type = “hidden” name = “a” value = “CSRF 토큰”>

<input type = “submit” value = “click”>

</form>

위 예시는 어떠한 사이트에 예시와 같은 코드와 같은 폼이 전송된다고 가정했을 때, 피싱 사이트에 똑같이 요청하는 폼이 있을 때, 희생자는 피싱 사이트에 접속함으로써 본인의 글에 해당 글이 등록된다.

3. SQL Injection

SQL Injection은 코드 삽입의 한 기법으로 입력 값을 조작하여 서버의 데이터베이스를 공격할 수 있는 공격기법을 말한다. 대부분 클라이언트가 입력한 데이터를 제대로 필터링하지 못할 때 발생된다. 공격의 난이도에 비해 파괴력이 크다. 워드프레스는 MySQL을 사용하므로, SQL문으로 예시를 들겠다.

예시 1) SELECT user FROM SITE\_USER WHERE id=’alice1234’ AND password=’’OR’1’=’1’;

예시 2) URL - <http://abc.com/search?=tom>

주요

위 예시 1의 SQL구문은 논리적 취약점을 이용한 SQL Injection이다. SQL은 OR연산보다 AND연산의 우선순위가 더 높기에 password=’’OR’1’=’1’;구문에서 실제 password는 ‘ ‘가 입력되었고, 그 뒤 OR’1’=’1은 연산자이다. 실제 비밀번호인 ‘ ‘가 DB에 없기 때문에 false가 출력되고, id인 alice1234는 DB에 있다고 가정하면 true and false = false가 출력되어 로그인이 되지 않아야하지만 OR’1’=’1에서 ‘1’과’1은 서로 같기 때문에 true가 출력된다. 실제 비밀번호은 ‘ ‘OR true가 되기에 실제 비밀번호가 무엇이든 SQL문의 최종결과값이 true가 되어 로그인이 된다.

위 예시 2의 URL문에서 최종적으로 수행되는 URL이라고 가정했을 때, 공격자의 이름은 tom이고 alice의 정보로 조회를 하고싶을 때 위 예시 1의 SQL문에서 password=’alice’OR’1’=’1’;로 바꾸면 해당 구문은 참이 되어 결과를 출력해준다.

이렇듯 웹 개발 프레임워크의 보안 취약성은 다양하고 간단하면서도 큰 피해를 줄 수 있고, 사용자가 알아차리기 어렵다.

**3. 연구목표**

위에서 제시한 워드프레스의 보안 취약점을 스캐닝할 수 있는 도구를 제작한다. 스캐너란 네트워크와 시스템, 애플리케이션을 자동으로 검사하여 공격에 노출될 만한 보안 취약점이 없는지 확인하는 도구이다. 스캐너를 이용하면 취약점을 식별, 분류, 완화할 수 있다. 이 도구는 대상 시스템에 HTTP 요청을 보낸 뒤 요청된 애플리케이션의 코드에 집중하여 파이썬을 사용해 정규표현식을 이용하여 취약점을 찾고, SQL Injection, XSS, CSRF 등 중요한 웹 결합 보안문제를 선제적으로 탐지한다.

XSS 취약점 점검

- 게시판이나 검색란에 HTML 코드나 JS 삽입 가능 여부

- 에러 메시지를 조작하여 HTML 코드나 JS 삽입 가능 여부

Injection 점검

- SQL Injection 취약점을 이용해 password 없이 로그인

- 인수에 특수문자를 삽입하여 서버 내부에 명령 전달

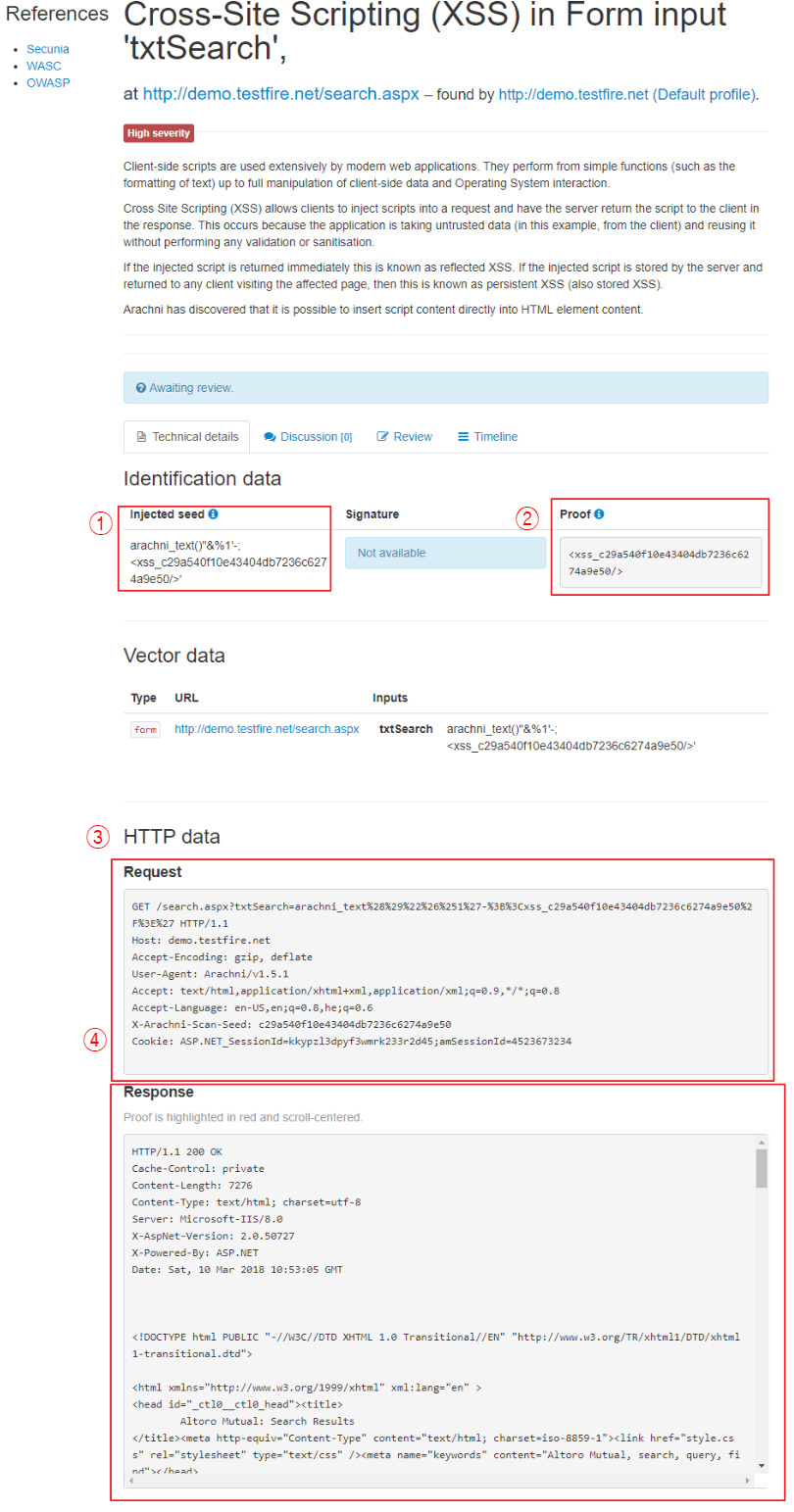
- 외부 스크립트를 서버 내부에 삽입

- 사용자 입력 시 특수문자가 포함되어 있는지 검사

CSRF 점검

- 웹 애플리케이션에 XSS 취약점 존재 여부 확인

- 소스 코드에 적절한 토큰을 사용하는지 확인



Arachni Scanner의 XSS Scan

**참고문헌**

[1]워드프레스의 동작 원리(<https://www.thewordcracker.com/basic/wordpress-page-structure/>)

[2]워드프레스의 디렉토리 구조<https://m.post.naver.com/viewer/postView.naver?volumeNo=6509616&memberNo=20459920>

[3]Arachni Scanner의 XSS Scan <https://xn--ex3bt1ov9l.kr/301>