웹 프레임워크의 보안 취약성 연구

연동현 정민우 오광주

경희대학교 컴퓨터공학과

dusehdgus12@naver.com p1nkjelly@pinkjelly.cat ohkwang1234@khu.ac.kr

Study on Security Vulnerabilities in Web Framework

Donghyeon Yeon, Minwoo Jeong, Kwangju Oh

Department of Computer Science and Engineering, Kyung Hee University

요약

과거 인터넷에서는 정적인 이미지와 텍스트만 있으면 웹을 만들 수 있었다. 이처럼 웹을 만드는 단순한 방법이 현대에 들어서 인터넷 금융, 전자 상거래 등을 통해 웹을 구현하는 기술의 난이도가 높아지고, 이를 해결하기 위해 웹 프레임워크가 등장하였다. 중요한 개인 정보 및 다양한 정보들의 교환이 웹 프레임워크로 만들어진 웹사이트라는 공간에서 이루어지고 있다. 웹이라는 공간에서 이루어지는 정보의 교환은 사용자들에게 편리함을 주지만, 웹은 우리 눈에 보이지 않는 심각한 보안의 위험성이 존재한다. 이에 본 논문에서는 이러한 보안의 취약성을 설명하고 이러한 취약성의 위험성을 분석하며, 보안의 위험성을 사전에 감지할 수 있는 방안을 소개해 웹 프레임워크의 보안 취약성을 통한 공격을 방지하고자 한다.

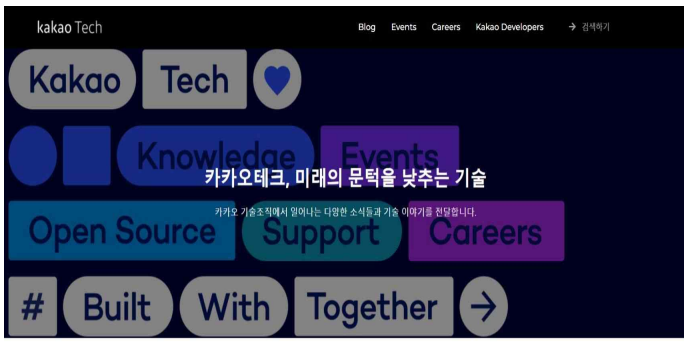
1. 서론

웹 프레임워크(Web Framework)[1]란 동적인 웹 페이지나 웹 애플리케이션, 웹 서비스 개발 보조용으로 만들어지는 애플리케이션 프레임워크의 일종이다. 웹 페이지를 개발하는 과정에서 겪는 어려움을 줄이는 것이 주 목적으로 통상 데이터베이스 연동, 템플릿 형태의 표준, 세션 관리, 코드 재사용 등의 기능을 포함하고 있다. 월드와이드 웹은 설계 시 동적이지 않았으며 손으로 코딩한 HTML로 이루어진 초기의 하이퍼텍스트로 이루어졌다. 이후 1993년 공통 게이트웨이 인터페이스 표준이 외부 애플리케이션과 웹 서버 간 통신을 위해 도입되었고, 이로 인해 사용자의 입력을 반영한 동적 웹 페이지를 만들게 되었다. 대표적으로 워드프레스, Django, Spring Framework 등이 있다. 동적인 웹 프레임워크의 발전으로 인해 현재 사용하는 거의 모든 인터넷은 웹 프레임워크로 만들어졌다. 이처럼 웹 프레임워크는 현대 웹사이트에서 가장 중요한 요소가 되었으며, 많은 개발자들이 다양한 웹 프레임워크로 수많은 웹사이트를 제작하고 있다. 1990년대까지만 해도 소프트웨어나 다양한 상품을 구입하려면 물리적인 매개체가 필요했다. 하지만 웹 기반 서비스가 시작됨에 따라 유통의 과정이 혁신적으로 줄면서 사용자들에게 많은 편리함을 제공했고, 이로인해 웹 기반 서비스를 통한 유통이 폭발적으로 발전하면서 HTML, CSS로 이루어진 전통적인 정적 웹을 넘어 동시 접속자 수와 데이터의 양이 증가된 동적 웹이 발전되었다. 이렇게 웹 프레임워크와 웹 기반 서비스의 발전으로 인해 개발자와 사용자는 쉽게 인터넷을 만들 수 있고 사용할 수 있지만, 웹을 통해 정보를 교환할 때 눈에 보이지 않는 다양한 취약점들이 존재한다. 사용자의 개인 정보를 공격자가 탈취해 금전적인 이익을 얻거나 개발자의 서버에 존재하는 데이터베이스를 공격해 사용자들의 개인 정보를 빼돌려 많은 사용자들의 정보가 유출되는 등 심각한 문제가 발생한다. 실제로 미국의 금융 기관인 캐피탈 원(Capital One)은 2019년에 웹 사이트 공격으로 인해 약 1억 6백만명의 개인정보가 유출되었고, 2008년 대한민국의 옥션에서는 같은 공격으로 1천 80만명의 이름과 전화번호등 개인정보가 유출되었다. 이에 본 논문에서는 사용자 및 개발자에게 심각한 피해를 주는 보안 취약점인 SQL Injection, SSRF 공격, CSRF 공격에 대해 소개하고, 웹 프레임워크 중 워드프레스를 이용해 JSON, PHP, 정규표현식, 파이썬을 통한 취약점을 사전에 방지할 수 있는 스캐닝 툴을 소개하고자 한다.

2. 기존연구

2.1 워드프레스(WordPress)

워드프레스(WordPress)[2]란 2003년 맷 뮬렌웨그에 의해 공식 배포된 웹사이트, 쇼핑몰, 블로그, 홈페이지 제작 관련 오픈소스 웹 프레임워크 프로그램이다. 현재 전 세계 웹사이트의 40% 이상이 워드프레스로 이루어져 있는 만큼 워드프레스는 웹 프레임워크의 핵심이다. 워드프레스로 제작 가능한 웹 사이트는 서비스 소개 홈페이지, 개인 포트폴리오 홈페이지, 쇼핑몰, 게시판, 수익형 블로그등 다양한 유형들을 복수로 조합해서 제작할 수 있다.



[그림 1]워드프레스로 제작된 카카오테크 홈페이지

워드프레스는 코어(Core), 테마(Theme), 플러그인(Plugin)으로 구성되어있다. 또한 Content Management System으로서 웹사이트를 쉽게 호스팅할 수 있도록 제공되는 오픈소스 솔루션이다. php파일로 구성되어 있으며 보통 Apache, mysql, mariadb와 같이

연동되어 구성된다. wp-admin, wp-content, wp-includes와 같은 핵심 디렉토리 밑에 하위 디렉토리들로 나뉘어져있다. 디렉토리들은 PHP코드, CSS, JS, HTML로 이루어져있다. 워드프레스로 웹이 동작하는 방법은 다음과 같다.

1. 사용자가 워드프레스 기반 웹사이트의 URL에 들어가 HTTP 요청을 한다.
2. HTTP 서버인 Apache는 요청을 받아 워드프레스의 PHP모듈에 동적인 HTML페이지를 요청한다.
3. 워드프레스는 이 요청을 받아 동적인 HTML을 만들고 HTTP서버에게 보낸다(필요하면 DBMS에서 데이터를 가져오거나 저장함).
4. HTTP 서버는 동적으로 생성된 HTML페이지를 사용자에게 전달한다.

2.2 정규표현식

정규표현식(REGEX)[3]는 프로그래밍에서 문자열을 다룰 때, 문자열의 일정한 패턴을 표현하는 형식 언어이다. 일정한 규칙을 가진 텍스트 문자열을 사용할 때 쓰이며, 복잡한 패턴을 가진 문자열을 찾을 때 편리하게 쓰일 수 있다.

```
import re

data = """
park 800905-1049118
kim  700905-1059119
"""

pat = re.compile("(\d{6})[-]\d{7}")
print(pat.sub("\g<1>-*******", data))
```

[그림 2] 정규표현식 예시

2.3 JSON

JSON이란 Javascript Object Notation의 축약으로, 데이터를 저장하고 교환할 수 있는 텍스트 기반의 데이터 교환 방식이다. 다양한 프로그래밍 언어에서 데이터를 읽고 사용할 수 있으며, 서버에서 클라이언트로 데이터를 보낼 때 쓰인다. JSON의 장점은 최소한의 용량으로 데이터 전송이 가능하고, 구조 정의의 용이성과 가독성이 뛰어나다. 본 논문은 JSON의 데이터 반환 형식인

```
{
    "1": {
        "description": "Check for ABSPATH existence",
        "rule": [
            "(require|require_once|include|include_once)\\s*\\([^)]*(?!.*ABSPATH)[^)]*\\)",
            "(require|require_once|include|include_once)\\s*'[^']*'"
        ]
    },
    "4": {
        "description": "Check for exec family functions",
        "rule": [
            "(exec|shell_exec|system|passthru|pcntl_exec|popen|proc_open|new\\s+Process)\\s*\\([^)]*\\)"
        ]
    },
    "5": {
        "description": "Check for SQL execution",
        "rule": [
            "\\$wpdb->(?:query|get_results)\\([^)]*\\)"
        ]
    },
    "9": {
        "description": "Check for SSRF potential using wp_remote_ and curl functions",
        "rule": [
            "wp_remote_(get|post|head|request)\\s*\\([^)]*\\)",
            "curl_setopt\\s*\\([^)]*\\)",
            "file_get_contents\\s*\\([^)]*\\)"
        ]
    }
}
```

[그림 3] JSON 데이터에 있는 문자열 패턴

dictionary를 이용해 문자열 패턴을 JSON 데이터에서 읽어와 정규 표현식을 생성할 것이다.

3. 취약점 분석

워드프레스를 필두로 웹 프레임워크를 이용해 웹 사이트를 구축하고 이용하는 것은 개발자나 사용자에게 편리하고, 현대 사회에서 웹 사이트의 비중은 엄청나다. 그렇지만 비중이 엄청나기에 워드프레스에서의 취약점은 매우 큰 파급효과가 있다. 다음은 웹 프레임워크의 대표적인 취약점들이다.

3.1 SQL Injection

SQL Injection이란 코드 삽입의 한 종류로 입력 값을 조작하여 서버의 데이터베이스를 공격할 수 있는 기법을 말한다. 대부분 클라이언트가 입력한 데이터를 제대로 필터링하지 못할 때 발생된다. 공격 난이도에 비해 파괴력이 큰 기법이다. [그림 4]의 ''OR'1'='1'처럼 논리적 취약점을 이용해 개인 정보를 탈취할 수 있다.

select ID from SITE_USER where PASSWORD="OR'1'='1';

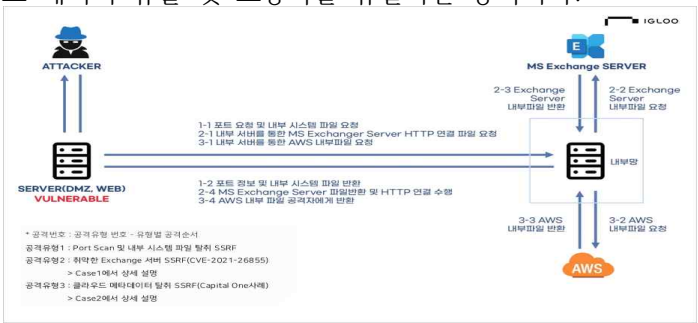
ID
1
25
19
22
34
37

(6 rows, 8 ms)

[그림 4] SQL Injection 예시

3.2 SSRF(Server Side Request Forgery)

SSRF는 서버 측에서 위조된 HTTP요청을 발생시켜 직접적인 접근이 제한된 서버 내부 자원에 접근하여 외부로 데이터 유출 및 오동작을 유발하는 공격이다.



[그림 5] SSRF 대표적인 공격 유형

[그림 5]와 같이 SSRF공격[4]은 내부 서버 포트 스캐닝(file?=http://10.10.10.10:22와 같은 구문)을 통한 서비스 구동여부 확인 및 내부파일 접근해 내부 데이터 탈취, 강제로 세션 연결이 가능한 ProxyLogon을 이용해 별도의 인증 없이 공격자와 exchange server간 HTTP 연결 수행 등 서버 내부에 접근해 사용자들의 정보를 탈취할 수 있다.

3.3 CSRF(Cross Site Request Forgery)

CSRF 공격이란 사용자가 자신의 의지와는 다르게 공격자가 의도한 수정, 삭제, 등록 등의 행위를 사용자가 사용하는 웹사이트에 요청하게 만드는 공격이다. CSRF 공격은 다음과 같이 이루어진다.

[그림5]출처 - SSRF 취약점을 이용한 공격사례 분석 및 대응방안
https://www.igloo.co.kr/security-information/ssrf-%EC%B7%A8%EC%95%BD%EC%A0%90%EC%9D%84-%EC%9D%B4%EC%9A%A9%ED%95%9C-%EA%B3%B5%EA%B2%A9%EC%82%AC%EB%A1%80-%EB%B6%84%EC%84%9D-%EB%B0%8F-%EB%8C%80%EC%9D%91%EB%B0%A9%EC%95%88/

아래 공격들은 XSS 취약점이 있다고 가정한다.

1. 공격자는 이메일이나 게시판에 Java Script가 포함된 게시물을 전송한다.
2. 사용자(관리자)는 공격자가 등록한 Java Script가 포함된 게시물을 확인한다.
3. 사용자(관리자)가 Java Script가 포함된 게시물을 열람하면, 사용자(관리자)의 권한으로 공격자가 원하는 Java Script 요청이 발생한다.
4. 악성 스크립트가 실행되어 사용자(관리자)에게 피해가 간다.

4. 해결방안

[그림 6]은 Python을 이용한 스캐닝 코드이다. 위 코드가 동작하는 원리는 다음과 같다.

```python
import json
import re

# JSON 데이터를 불러오기
with open('pwn_rule.json', 'r') as json_file:
    data = json.load(json_file) #data는 딕셔너리타입

# PHP 파일 경로 설정
php_file_path = 'test.php'

# PHP 파일 읽기
with open(php_file_path, 'r', encoding='utf-8') as php_file:
    php_code = php_file.read()

# 패턴을 JSON 데이터에서 읽어와서 정규 표현식 생성
for rule in data.values(): # rule은 딕셔너리타입

    for pattern in rule['rule']: #'rule' 키 검색
        compiled_pattern = re.compile(pattern)

        # 정규 표현식에 일치하는 부분 모두 출력
        # php파일에서 REGEX와 일치하는 모든 부분 찾음
        matches = compiled_pattern.finditer(php_code)
        for match in matches:
            print(f"Match found for input value: {php_file_path}")
            print(f"Description: {rule['description']}")
            print(f"Rule: {pattern}")
           # 함수 이름과 매개변수 추출
            function_match = re.match(r'([a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*)\s*\(', match.group(0))
            if function_match:
                function_name = function_match.group(1).strip()
                print(f"Function: {function_name}")

            # 매개변수 추출 (변수 또는 문자열)
            parameter_pattern = re.compile(r'\((.*?)\)')
            parameter_matches = parameter_pattern.findall(match.group(0))
            if parameter_matches:
                for parameter_match in parameter_matches:
                    parameters = parameter_match.strip()
                    print(f"Parameters: {parameters}")

            # new line
            print()
```

[그림 6] Python을 이용한 스캐닝 코드

1) JSON 파일에 있는 문자열 패턴을 읽어 정규표현식을 만든다.

```
//1번 Session/Context 없이 실행시킬 수 있는 코드의 유무 파악 (ABSPATH 확인)
\b(require|require_once|include|include_once)\s*\([^)]*(?!ABSPATH)
if ( !defined('ABSPATH') )
//4번 exec 계열 함수들 (php 관련 조사. php 가 아닌 부분도 가능할지?)
(exec|shell_exec|system|passthru|pcntl_exec|popen|proc_open|new\s+Process)\s*\(
//5번 SQL 실행부분 (관련함수들 나열)
\$wpdb->(?:query|get_results)\s*\(
//9번 SSRF 가능성 있는부분 (php 관련함수 조사 하여 패턴검사, https 패턴검사)
wp_remote_(get|post|head|request)\s*\(
curl_setopt\s*\(
file_get_contents\s*\(
```

[그림 7] PHP파일에서 발생 가능한 취약점 공격들

JSON 파일에는 [그림 7]과 같이 PHP파일에서 발생 가능한 취약점들을 스캐닝할 수 있는 문자열 패턴이 들어있다(JSON 파일은 [그림 3]과 동일함).

2) PHP파일([그림 8]과 같은 파일)을 읽어와 PHP파일에서 정규표현식과 일치하는 부분을 모두 찾아, PHP파일에서 발생 가능한 모든 취약점 공격들을 스캔하여 해당 함수와 변수들을 출력한다([그림 9]와 같이 출력).

3) 출력된 함수와 변수들을 통해 취약점 공격이 발생될 수 있는 부분을 사전에 파악하여 강화해 나갈 수 있다.

```
<?php
// PHP 파일 내용 예제

// 1. ABSPATH 확인
if (!defined('ABSPATH')) {
    // ABSPATH가 정의되지 않았을 때
    require 'a.php';
}

// 2. exec 계열 함수 호출
$cmd='ls-la'
exec($cmd);

// 3. SQL 실행
$q='SELECT * FROM table'
$wpdb->query($q);

// 4. SSRF 가능성 있는 부분
$url='https://example.com'
wp_remote_get($url);

// 5. curl 함수 호출
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $url);

// 6. file_get_contents 호출
$file_content = file_get_contents($url);
?>
```

```
Match found for input value: vuln_php.php
Description: Check for ABSPATH existence
Rule: (require|require_once|include|include_once)\s*\([^)]*

Match found for input value: vuln_php.php
Description: Check for exec family functions
Rule: (exec|shell_exec|system|passthru|pcntl_exec|popen|proc_open|new\s+Process)
\s*\([^)]*
Function: exec
Parameters: $cmd

Match found for input value: vuln_php.php
Description: Check for SQL execution
Rule: \$wpdb->(?:query|get_results)\([^)]*
Parameters: $q

Match found for input value: vuln_php.php
Description: Check for SSRF potential using wp_remote_ and curl functions
Rule: wp_remote_(get|post|head|request)\s*\([^)]*
Function: wp_remote_get
Parameters: $url

Match found for input value: vuln_php.php
Description: Check for SSRF potential using wp_remote_ and curl functions
Rule: curl_setopt\s*\([^)]*
Function: curl_setopt
Parameters: $ch, CURLOPT_URL, $url

Match found for input value: vuln_php.php
Description: Check for SSRF potential using wp_remote_ and curl functions
Rule: file_get_contents\s*\([^)]*
Function: file_get_contents
Parameters: $url
```

[그림 8] PHP 파일의 예시 [그림 9] 스캔된 함수와 변수들의 출력

5. 향후 발전 계획

본 논문은 웹 프레임워크 중 전세계에서 가장 많이 쓰이는 워드프레스를 통해 심각한 피해를 줄 수 있는 웹 취약점 공격을 스캐닝하여 공격을 사전에 방지할 수 있는 툴을 소개한다. 위의 [그림 8]은 실제 존재하는 워드프레스로 이루어진 웹사이트에 대한 PHP 파일이 아니다. 이에 향후 실제 오픈소스로 서비스되고 있는 워드프레스 웹사이트의 PHP 파일에 대해 스캐닝을 통해 취약점 공격을 분석하고 통계화할 계획이다. 이를 통해 현대 사회에서 가장 중요한 요소인 웹 사이트의 취약점을 이용한 공격에 대비할 수 있을 것이다.

6. 참고문헌

[1]-웹 프레임워크란
https://ko.wikipedia.org/wiki/%EC%9B%B9_%ED%94%84%EB%A0%88%EC%9E%84%EC%9B%8C%ED%81%AC
[2],그림 1-워드프레스란
https://news.wp-kr.com/wordpress-start-guide/#:~:text=%ED%98%84%EC%9E%AC%20%E2%80%9C%EC%A0%84%EC%84%B8%EA%B3%84%E2%80%9D%20%EC%9B%B9%EC%82%AC%EC%9D%B4%ED%8A%B8%EC%9D%98%2040%25%20%EC%9D%B4%EC%83%81%EC%9D%B4,%EC%9B%8C%EB%93%9C%ED%94%84%EB%A0%88%EC%8A%A4%20%
[3]-정규표현식 예시(그림2)
https://wikidocs.net/1642
[4]-SSRF공격이란 -
https://www.igloo.co.kr/security-information/ssrf-%EC%B7%A8%EC%95%BD%EC%A0%90%EC%9D%84-%EC%9D%B4%EC%9A%A9%ED%95%9C-%EA%B3%B5%EA%B2%A9%EC%82%AC%EB%A1%80-%EB%B6%84%EC%84%9D-%EB%B0%8F-%EB%8C%80%EC%9D%91%EB%B0%A9%EC%95%88/