OWASP 웹 애플리케이션 보안 사이트 설계 및 구현

김선혁, 김유진, 최태용, 전보경, 김선민, 안수윤, 정기현*

Design and Implementation of OWASP Web Service System

Sun-Hyuk Kim, Yu-Jin Kim, Tae-Yong Choi, Bo-Kyung Jeon, Sun-Min Kim, Soo-Yoon Ahn, Ki-Hyun Jung*

요 약

본 연구에서는 웹 서비스의 수요 증진과 더불어 각종 웹 보안 위협이 여러 방향으로 대두되는 현 상황에 대응하여, OWASP Top 10을 기반으로 한 웹 애플리케이션 보안 종합 포털사이트를 설계하고 구축하였다. 본 서비스는 웹 개발자와 일반 이용자에게 웹 애플리케이션 보안 취약점을 식별하고 대응할 수 있는 지침을 제공하며, 웹 애플리케이션 보안 수준의 제고를 목표로 하기 위해, OWASP Top 10에 명시된 10가지 웹 애플리케이션 보안 취약점을 분석한 정보를 제공하고, OWASP ZAP을 활용하여 보안 취약점을 자동으로 분석하는 점검 도구와 웹 크롤링 기술을 이용하여 수집된 실제 웹 애플리케이션 보안 관련 피해 사례를 제공하는 서비스를 구축하였다. 구축된 시스템은 사용자가 웹 사이트의 보안 취약점을 실시간으로 파악하고 필요한 보안 조치를 즉시 취할 수 있도록 지원하는 서비스를 제공함으로써 웹 애플리케이션 보안에 대한 일반 사용자의 접근성과 이해도를 높이고, 웹 사이트의 안전성을 향상시킬 수 있을 것이다.

Abstract

In response to the increasing demand for web services and the current situation where various web security threats are emerging from various directions, we have established a comprehensive web application security portal based on the OWASP Top 10. This service provides guidelines for web developers and general users to identify and respond to web application security vulnerabilities and aims to improve the level of web application security. To this end, we built a service that provides information analyzing the 10 web application security vulnerabilities specified in the OWASP Top 10, a checking tool that automatically analyzes security vulnerabilities using OWASP ZAP, and actual web application security-related damage cases collected using web crawling technology. Based on the research results, we have successfully developed a service that enables users to identify security vulnerabilities in websites in real time and take necessary security measures immediately, which will increase the accessibility and understanding of web application security for general users and improve the safety of websites.

Key words

Web Service, Web Application Security, OWASP Top 10, Cybersecurity Threats

---

국립안동대학교, shinsang1234@naver.com, *국립안동대학교, kingjung@anu.ac.kr (교신저자)

※ 본 연구는 2023년 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업의 연구결과로 수행되었음 (2019-0-01113)

Ⅰ. 서 론

최근 웹 서비스 기술의 중요성이 증대되는 현 상황에서 웹 애플리케이션 서비스에 대한 수요는 지속적으로 증가하고 있다. 디지털화의 확산에 따라 웹 서비스 기술에 대한 사용자들의 요구가 상승하고 있으며, 이에 대응하기 위해 기업들은 웹 개발에 특화된 웹 개발 전문가들을 적극적으로 채용하고 있다. Github에서 발표한 2015년부터 2018년까지의 통계에 따르면, 개발자 수는 매년 평균 1.5배의 증가율을 보이고 있으며, 기업들은 소프트웨어 분야의 사업 진흥을 위해 관련 역량을 보유한 개발자 확보와 인재 양성에 주력하고 있는데, 이는 웹 개발자의 수와 그들의 산업 내 비중이 증가하고 있음을 나타낸다[1].

Stack Overflow의 설문조사 결과에 따르면, JavaScript, Node.js 그리고 Python 등의 웹 개발 기술은 현재의 기술 생태계에서 가장 높은 비중을 차지하고 있음을 확인하였다. 이는 웹 개발자의 수의 증가를 시사한다[2].

코로나19 팬데믹 이후로는 비대면 활동의 증가에 따라 사이버 공간의 위험성이 상승하였다. 특히, 악성코드와 악성 도메인의 증가는 사이버 공격의 위험성을 높이고 있으며, 기업의 소프트웨어 공급업체나 네트워크 하드웨어의 취약점을 통한 공격 사례 또한 다수 발생하고 있다[3].

웹 서비스 기술에 대한 수요가 꾸준히 증가하고 있는 현 상황에서 웹 애플리케이션 보안에 대한 중요성은 더욱 강조되어야 하며, 현재까지도 웹 애플리케이션 보안 문제에 대해 충분한 경각심을 갖지 못하고 있는 상태에서 다수의 피해 사례가 발생하고 있다는 점을 고려할 때, 본 연구의 필요성은 더욱 높아질 것으로 보여진다.

본 논문은 해당 배경 하에 OWASP Top 10을 기반으로 웹 애플리케이션 보안 종합 포털사이트 홈페이지를 구축하여 웹 애플리케이션 공격 및 취약점의 원인을 분석하고 그에 대한 대응 방안을 제시하고자 한다. 본 서비스는 웹 개발자와 이용자에게 웹 애플리케이션 보안 취약점을 식별하고 대비할 수 있는 지침을 제공하며, 웹 애플리케이션 보안 수준의 제고 및 안전한 웹 서비스 상태의 향상을 도모하고자 한다. 나아가 웹 애플리케이션 보안의 중요성을 강조하고, 보다 안전한 웹 애플리케이션 환경 구축에 기여하고자 한다.

Ⅱ. 관 련 연 구

2.1 OWASP 기존 사이트 분석

OWASP (Open Web Application Security Project)는 웹 애플리케이션 보안 향상을 목표로 하는 국제적 비영리 단체이며, 본 논문에서는 OWASP가 제공하는 웹 애플리케이션 보안 가이드라인 및 리소스를 참조하여 웹 애플리케이션의 보안 취약점을 철저히 분석하였다[4].

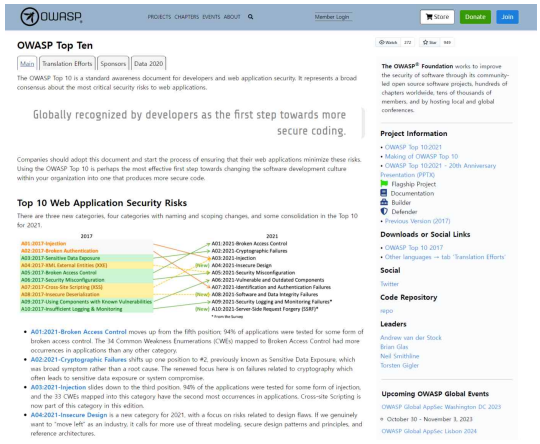

그림 1. OWASP 정보제공 사이트 (owasp.org)
Fig. 1. OWASP information site (owasp.org)

해당 사이트에 명시된 OWASP Top 10은 웹 애플리케이션의 주요 보안 취약점 10가지를 선정한 목록이며, 3~4년의 주기로 갱신된다. 이때, 10가지 취약점은 실제 발생 빈도와 공격의 영향 범위, 악용 가능성 등 다양한 기준에 의거하여 선정되며, 본 연구에서는 이를 기반으로 웹 애플리케이션 보안 상태를 점검하고, 해당 취약점들에 대한 각각의 공격 기법 및 진단 방안, 대응 방안을 사용자가 자가 진단 할 수 있는 서비스 제공 방안에 대하여 연구하였다.

	취약점 목록
1	접근 권한 취약점(Broken Access Control)
2	암호화 오류(Cryptographic Failures)
3	인젝션(Injection)
4	안전하지 않은 설계(Insecure Design)
5	보안설정오류(Security Misconfiguration)
6	취약하고 오래된 요소(Vulnerable and Outdated Components)
7	식별 및 인증 오류(Identification and Authentication Failures)
8	소프트웨어 및 데이터 무결성 오류(Software and Data Integrity Failures)
9	보안 로깅 및 모니터링 실패(Security Logging and Monitoring Failures)
10	서버 측 요청 위조(Server-Side Request Forgery)

그림 2. 2021 OWASP Top 10 목록
Fig. 2. 2021 OWASP Top 10 List

2.2 OWASP ZAP

OWASP ZAP (Zed Attack Proxy)은 웹 애플리케이션의 보안 취약점을 탐지하기 위한 오픈소스 보안 스캐너이다. 본 연구에서는 OWASP ZAP을 이용하여 웹 애플리케이션의 보안 취약점을 정밀하게 식별하고, 분석 진단 결과에 대한 데이터를 웹 서비스로 제공할 수 있는 방안에 대하여 연구하였다.

Ⅲ. 시스템 설계 및 구현

본 연구의 목표는 사용자가 본 서비스 내에서 웹 애플리케이션 보안 점검을 손쉽게 실행하고 그에 대한 진단 결과를 실시간으로 제공 받을 수 있도록 지원하는 것이다. 사용자는 본 서비스를 통하여 웹 사이트의 보안 취약점을 실시간으로 파악할 수 있고, 필요한 보안 조치를 즉시 취할 수 있다. 또한, 본 연구에서는 웹 애플리케이션 보안에 관한 다양한 자료를 취합하여 사용자가 필요로 하는 정보만을 제공할 수 있도록 설계하였다.

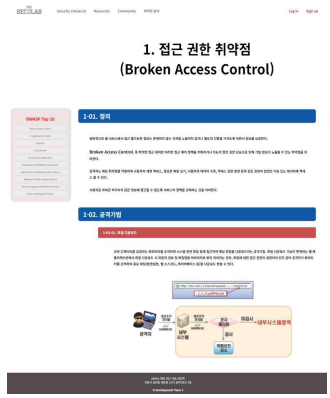

그림 3. OWASP 취약점 정보 제공 페이지 (일부)
Fig. 3 OWASP vulnerability information page (partial)

3.1 취약점 자동화 분석

본 연구는 OWASP ZAP을 이용하여 웹 애플리케이션의 보안 취약점을 자동으로 분석하는 통합 침투 테스트 웹 서비스를 구축하였다. Python 언어를 활용하여 ZAP의 API를 호출하며, 프록시 설정을 조정하여 로컬 웹 서버에서 실행되는 ZAP 프록시 서버와 통신이 가능케 하였다. 이어서 본 연구에서 구현한 Frontend 서비스를 통해 사용자로부터 URL을 입력받아 해당 URL에 대한 웹 보안 취약점을 Spider와 Scan하는 프로세스를 자동화하였다. Spider는 지정된 URL에 연결된 모든 페이지와 리소스를 크롤링하며, 이후 Passive Scan과 Active Scan을 실행하여 보안 취약점을 체계적으로 탐색한다.

```
import django
django.setup()
from zap.models import LogZap
import time
from zapv2 import ZAPv2

target = [redacted]

if len(sys.argv) > 1:
    target = sys.argv[1]

apikey = [redacted]

zap = ZAPv2(apikey=apikey, proxies={'http': 'http://127.0.0.1:[redacted]', 'https': 'http://127.0.0.1:[redacted]'})

zap.urlopen(target)
time.sleep(2)

scanid = zap.spider.scan(target)
time.sleep(2)

while (int(zap.spider.status(scanid)) < 100):
    time.sleep(2)

while (int(zap.pscan.records_to_scan) > 0):
    time.sleep(2)
scanid = zap.ascan.scan(target)

while (int(zap.ascan.status(scanid)) < 100):
    time.sleep(5)

alerts = zap.core.alerts()
for alert in alerts:
    message = 'Alert: {}'.format(alert)
```

그림 4. 웹 취약점 자동화 분석 코드 (일부)
Fig. 4 System Flowchart

3.2 취약점 분석 결과 제공

스캔을 통해 얻은 웹 애플리케이션 취약점 결과

정보는 OWASP의 분류 기준에 따라 High (높음), Medium (중간), Low (낮음), Informational (참조)의 네 가지 단계로 분류된다 [5]. 분류된 각 취약점 정보는 json 형식으로 수집되며, Django Web Framework를 활용하여 데이터베이스에 저장한다. 본 연구에서는 저장된 json 형식의 취약점 정보를 직접 설계한 데이터 가공 모듈을 통해 사용자 친화적인 형태로 변환 후 웹 서비스를 통해 사용자에게 실시간으로 제공함으로써, 사용자는 필요한 보안 조치를 신속하게 취할 수 있는 기반을 마련한다.

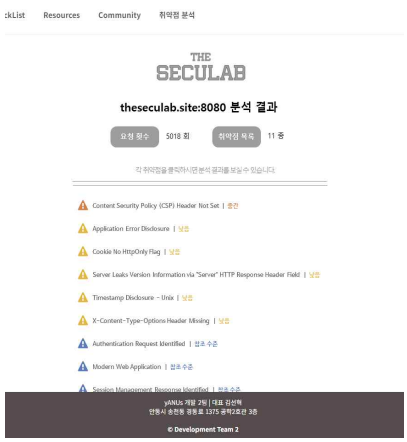

그림 5. ZAP을 이용한 취약점 분석 결과 제공 페이지
Fig. 5 Vulnerability analysis results using ZAP page

3.3 관련 보안 피해 사례 제공

뿐만 아니라 본 연구에서는 사용자에게 실제로 발생하였던 웹 애플리케이션 관련 보안 피해 사례 기사를 제공하기 위해 웹 크롤링(Crawling) 기술을 채택하여 NAVER 뉴스를 대상으로 한국어로 작성된 기사의 이미지, 본론, 작성자 등의 정보를 수집하고자 하였다. 이를 위해 Python 언어와 BeautifulSoup 패키지를 활용하였다.

3.4 웹 크롤링 (Web Crawling) 과정

본 연구는 키워드 기반의 웹 크롤링 알고리즘을 개발하였다. NAVER 사이트의 URL을 생성 방식을 활용하여 특정 키워드에 대한 기사를 자동으로 수집하는데, 해당 과정에서 두 가지 단계의 키워드 리스트를 이용하는 방식을 설계하였다.

첫 번째 단계에서의 키워드 리스트에는 “접근 권한 취약점“, “암호화 실패“, “암호화 오류“, “SSRF” 등 본 논문에서 다루는 OWASP TOP 10 웹 애플리케이션 10가지 취약점에 관련된 키워드가 포함되어 있으며, 해당 단계에서의 키워드 리스트를 생성 시 검색할 페이지 수를 함께 지정한다. 리스트에 저장된 키워드를 순서대로 검색하고 지정된 페이지 수만큼 URL을 생성하여 각 기사의 HTML 요소를 이용하여 기사의 제목, 본문, 작성자, 사진 URL, 그리고 날짜를 수집한다.

두 번째 단계에서의 키워드 리스트는 “피해액“, “피해 사례“, “피해자“, “피해 그룹“ 등으로 구성되어 있으며, 이는 첫 번째 단계에서 수집된 기사 데이터 중 실제 피해 사례와 관련된 내용을 선별하기 위한 목적으로 사용된다. 수집된 데이터 중 2차 키워드 리스트를 통하여 본문에 해당 리스트의 요소가 포함된 기사들만 최종 리스트에 저장한다. 저장된 리스트를 Python의 pandas 라이브러리를 활용하여 DataFrame으로 가공한다.

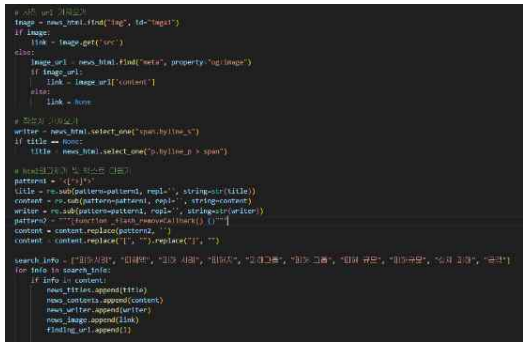

그림 6. 2차 필터링을 이용하여 저장하는 코드 (일부)
Fig. 6 Saving code with secondary filtering (partial)

3.5 수집된 데이터 저장

DataFrame에 저장된 데이터는 중복 행 데이터 수집 여부를 검사한 후, CSV 파일로 저장됨과 동시에 Django ORM을 활용하여 서버의 데이터베이스에 가공된 데이터들을 추가하는 방식으로 활용된다.

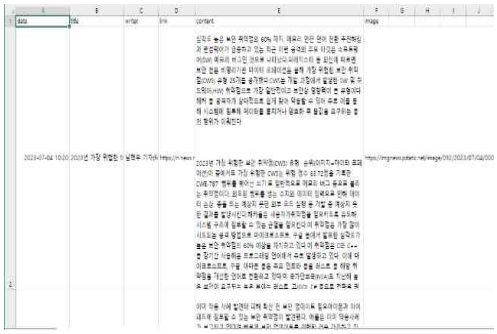

그림 7. 수집된 크롤링 뉴스 데이터 CSV화
Fig. 7 CSV the collected crawled news data

Ⅳ. 결 론

본 연구에서는 웹 기술의 발전에서 비롯된 각종 사이버 보안 문제에 대응하기 위해 10가지 취약점에 대한 공격 기법, 진단 방안 및 대응 방안이 포함된 정보와 웹 크롤링을 이용한 실시간 관련 보안 피해 사례를 제공하고, OWASP ZAP을 활용하여 웹 애플리케이션의 보안 상태를 점검할 수 있는 웹 애플리케이션 보안 종합 포털사이트를 개발하였다.

향후 연구에서는 현재 분석에 포함된 취약점이 OWASP Top 10에 국한되어 있는 한계를 인지하고, 실제 이용자들의 사용이 제한적이어서 충분한 데이터 수집이 이루어지지 않았다는 점을 해결하기 위해 보다 다양한 웹 애플리케이션 보안 취약점에 대한 분석을 추가적으로 수행하며 사용자 피드백을 적극적으로 수집하여 서비스의 사용성과 기능성을 개선할 계획이다.

참 고 문 헌

[1] 권영환, “오픈소스관련 SW일자리 동향”, SPRI 소프트웨어정책연구소, May 2019.

[2] 유재홍, “Stack Overflow의 글로벌 SW개발자 현황 조사 결과 분석“, SPRI 소프트웨어정책연구소, May 2019.

[3] 이용필, 이동근, “코로나19 관련 사이버 공격 및 대응현황 분석, Journal of Information Technology Services, October 2021.

[4] “The Open Web Application Security Project (OWASP)”, owasp.org, September 2011.

[5] 이규해, 창병모, and 최광훈, “OWASP ZAP을 이용한 다계층 프로그래밍 언어 Links의 웹 취약점 분석“, 한국정보과학회 학술발표논문집, 개최지, pp. 865-867, December 2021.