JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 29 No. 4, pp. 83-89, April 2024
https://doi.org/10.9708/jksci.2024.29.04.083

Propose a Static Web Standard Check Model

Hee-Yeon Won*, Jae-Woong Kim**, Young-Suk Chung*

*Student, Dept. of Computer Engineering, Kongju National University, Cheonan, Korea
**Professor, Dept. of Software Engineering, Kongju National University, Cheonan, Korea
*Lecturer, Dept. of Computer Engineering, Kongju National University, Cheonan, Korea

[Abstract]

After the end of the service of Internet Explorer, the use of ActiveX ended, and the Non-ActiveX policy spread. HTML5 is used as a standard protocol for web pages established based on the Non-ActiveX policy. HTML5, developed in the W3C(World Wide Web Consortium), provides a better web application experience through API, with various elements and properties added to the browser without plug-in. However, new security vulnerabilities have been discovered from newly added technologies, and these vulnerabilities have widened the scope of attacks. There is a lack of research to find possible security vulnerabilities in HTML5-applied websites.

This paper proposes a model for detecting tags and attributes with web vulnerabilities by detecting and analyzing security vulnerabilities in web pages of public institutions where plug-ins have been removed within the last five years. If the proposed model is applied to the web page, it can analyze the compliance and vulnerabilities of the web page to date even after the plug-in is removed, providing reliable web services. And it is expected to help prevent financial and physical problems caused by hacking damage.

▸ **Key words**: Web Standards, Web Compatibility, Web Vulnerabilities, HTML5, Static Analysis

[요 약]

인터넷 익스플로러의 서비스 종료 이후 ActiveX의 사용이 종료됨에 따라 Non-ActiveX 정책이 확산되었다. Non-ActiveX 정책을 바탕으로 정해진 웹 페이지 표준 규약으로 HTML5가 채택되어 사용되고 있다. W3C(World Wide Web Consortium)에서 개발된 HTML5는 다양한 기능을 플러그인 없이 브라우저만으로 쉽게 사용할 수 있고, 기존 HTML에 비해 다양한 요소와 속성이 추가되었으며 API를 통해 더 나아진 웹 응용 환경을 제공하고 있다. 그러나 새로 추가된 기술들로부터 새로운 보안 취약점이 발견되었고, 이러한 취약점으로 인하여 공격 범위가 넓어졌다. HTML5가 적용된 웹 사이트에서 발생할 수 있는 보안 취약점을 찾기 위한 연구가 부족하다.

본 논문은 최근 5년 이내에 플러그인이 제거된 공공기관 웹 페이지를 대상으로 웹 페이지의 보안 취약점을 탐지하고 분석하여 웹 취약점을 가지는 태그 및 속성을 탐지하는 모델을 제안한다. 제안된 모델을 웹 페이지에 적용한다면 플러그인 제거 후에도 현재까지 웹 페이지의 웹 표준 준수 여부 및 취약점을 분석할 수 있어 신뢰성 있는 웹 서비스를 제공할 수 있다. 그리고 해킹 피해로 인한 금전적, 물리적 문제들을 예방하는 데 도움이 될 것으로 기대된다.

▸ **주제어**: 웹 표준, 웹 호환성, 웹 취약점, HTML5, 정적분석

• First Author: Hee-Yeon Won, Corresponding Author: Jae-Woong Kim
*Hee-Yeon Won (why980909@gmail.com), Dept. of Computer Engineering, Kongju National University
**Jae-Woong Kim (jykim@kongju.ac.kr), Dept. of Software Engineering, Kongju National University
*Young-Suk Chung (merope@kongju.ac.kr), Dept. of Computer Engineering, Kongju National University
• Received: 2024. 01. 25, Revised: 2024. 04. 15, Accepted: 2024. 04. 15.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

웹 브라우저 점유율이 가장 높았던 인터넷 익스플로러(IE)가 11버전을 마지막으로 2022년 6월 이후 서비스가 종료되었다[1]. 인터넷 익스플로러는 다양한 웹 서비스를 활용하기 위한 플러그인 기술로 ActiveX를 사용했다[2]. 그러나 브라우저 기반인 확장프로그램 ActiveX는 웹 사이트에 따라 프로그램의 중복 설치로 인해 속도 저하, 프로그램 설치 시 브라우저가 강제 종료로 인한 사용자의 불편 증가와 플러그인의 자체 보안의 취약점 등 문제들이 발생했다. 다양한 웹 서비스를 이용하기 위해서는 ActiveX를 기반으로 필수 프로그램의 설치 해야 했고 이것으로 잠재적 보안 위협이 증가했다[3]. 이를 대체 하기 위해 Non-ActiveX 정책이 확산하였고 다양한 기능을 플러그인 없이 사용할 수 있는 HTML5로 전환했다[4].

2014년 최종 표준안을 발표한 W3C의 HTML5는 지금까지 스마트폰과 태블릿 등 다양한 모바일 기기 등을 대상으로 기존 HTML에 비해 다양한 요소와 속성이 추가된 태그와 web API를 통해 한 단계 진화된 웹 표준 환경을 제공했다[6][7]. 그러나 새로 추가된 기술로 인하여 새로운 보안 취약점이 발생하였고 이러한 취약점들로 인해 공격 범위가 넓어졌다[5]. 이러한 보안 문제를 해결하기 위해 HTML5를 기반으로 만들어진 웹 표준을 각 웹 페이지가 지키고 있는지 분석 후 취약점을 탐지하는 모델이 제안되었다. 그러나 제시된 모델들은 보안 체크에 관련된 연구만을 진행하고 해결책을 제시하지 않고 있다.

본 논문은 정적 웹 표준을 체크하기 위해 웹 url를 통해 HTML5로 이루어진 웹 페이지를 분석 후 웹 표준에 벗어난 태그들을 탐지 후 해결 방안을 출력하는 모델을 제안한다. 해결 방안을 출력함으로써 웹 표준의 취약점이 생기는 곳을 검색하여 웹 호환성 및 보안 문제에 대해 빠르게 파악하여 신뢰성 있는 웹 서비스를 제공할 수 있으며 해킹 피해로 인한 금전적, 물리적 문제들을 예방하는데 도움이 될 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 통해서 웹 표준인 HTML5와 그에 대한 취약점 그 표준안을 발표한 W3C에 대해 설명하고 웹 표준 분석을 위한 정적 분석 방법과 웹 페이지의 실시간 분석을 위한 웹 크롤링을 설명한다. 3장에서는 웹 페이지 정적 태그 분석 설계와 모델이 제안한다. 4장에서는 해당 모델의 결과를 나타낸다. 마지막으로 5장에서는 해당 모델의 결과를 기술하고 향후 연구과제와 그에 따른 연구 방향에 대해 논의한다.

II. Preliminaries

1. Related works

1.1 HTML5

HTML5는 웹 문서를 만들기 위한 기본 프로그래밍 언어인 HTML의 최신 규격이다. 기존 HTML 기능에 다양한 애플리케이션을 표현하였으며 제공하도록 진화한 웹 프로그래밍 언어이다. HTML5의 핵심은 브라우저를 통해 접근할 수 있는 시스템 자원을 API 도구를 통해 대폭 보강하는 것으로써, 웹 브라우저 내부 프로그램에 API를 제공하여, 작성된 코드가 웹 브라우저상에서 다양한 응용 프로그램으로 동작하도록 한다[8].

1.2 Static Analysis

소프트웨어가 실행되지 않은 환경에서 소스 코드의 의미를 분석하여, software defect를 찾아내는 분석 기법이다. software defect는 시스템의 성능을 떨어뜨리거나 멈춤, 오동작을 유발하는 소프트웨어의 구문 및 일반적으로 설계상의 오류도 포함하여 찾아내서 분석하는 기법이다. 또 한 코드의 보안 약점을 점검하여 완성된 소프트웨어의 발생 가능한 잠재적인 취약점을 예방하는 점검 방법이다. 분석을 통하여 데이터 흐름 분석과 같은 다양한 방법을 사용하여, 메모리 누출, 버퍼 오버플로우와 같은 문제를 발견할 수 있다[9].

1.3 HTML5 Security Vulnerability

플러그인 없이 다양한 기능을 구현할 수 있는 HTML5는 기능의 추가와 변경됨에 따라 신규 태그와 API를 악용한 공격들이 나타났다. 크게 HTML5의 보안 취약점은 크게 일반적인 웹 공격인 XSS공격, Click-jacking, Spoofing 공격과 HTML5 API에 따른 보안 위협인 Web Storage나 Application Cache 등이 있다[10].

1.3.1 XSS attack

서버와 클라이언트 사이에서 HTTP 프로토콜 동작 과정에서 발생한다. 서버가 아닌 클라이언트가 웹 애플리케이션에 악성 스크립트를 삽입하여 의도하지 않은 동작을 실행하여 쿠키, 세션이 탈취될 수 있고 타인의 권한을 탈취하여 HTTP 프로토콜 요청을 보내게 할 수도 있다.[15] 이러한 공격의 대응방법은 HTML5의 새로운 태그와 속성 문자를 필터링하거나 세션 히스토리 목록 삭제 및 세션 히스토리의 최대로 저장 가능한 개수를 조정한다.[5]

1.3.2 Clickjacking attack

공격자가 심어놓은 다른 웹 문서를 클릭하도록 유도하는 공격 기법이다. 이러한 공격의 대응방법은 스크립트나, iframe, opacity 속성 필터링을 하거나 드래그 앤 드롭 기능을 제한한다.[5]

1.4 Web crawling

웹 페이지 수집 프로그램인 BeautifulSoup, Selenium, Jsoup과 같은 웹 스크래핑 라이브러리를 이용하여, 인터넷 상에 공개된 웹 사이트의 웹 페이지를 조회하며 웹 페이지 정보를 수집하는 행위이다. 구글의 뉴스 서비스 방식은 검색 로봇을 이용하여 다른 웹 사이트에 게재된 뉴스 기사를 크롤링한 후 크롤링 정보를 이용하여 뉴스 기사 페이지를 구성하여 링크 정보를 제공해 주는 형태를 취하게 되는데 인터넷에서 정보수집을 위해서 일반적으로 사용되는 기술이다[11]. 웹 크롤링은 다양한 분야에 응용되어 사용되고 있다. 딥러닝에 사용될 이미지 수집 및 분류하기 위해서 웹 크롤링을 활용한 연구가 진행되었다[12], 그리고 유해 사이트 정보를 찾기 위해서 웹 크롤링을 적용한 연구가 진행되었다[13].

III. The Proposed Scheme

3.1 Static Web Standard Check Model Flowchart

본 논문에서는 웹 페이지를 분석 후 웹 표준에 벗어난 태그들을 탐지 후 해결 방안을 제시하는 정적 웹 표준 체크 모델이 제안한다.

본 논문에서 제안한 정적 웹 표준 체크 모델의 전체 시퀀스 다이어그램은 다음 Fig. 1.과 같다.

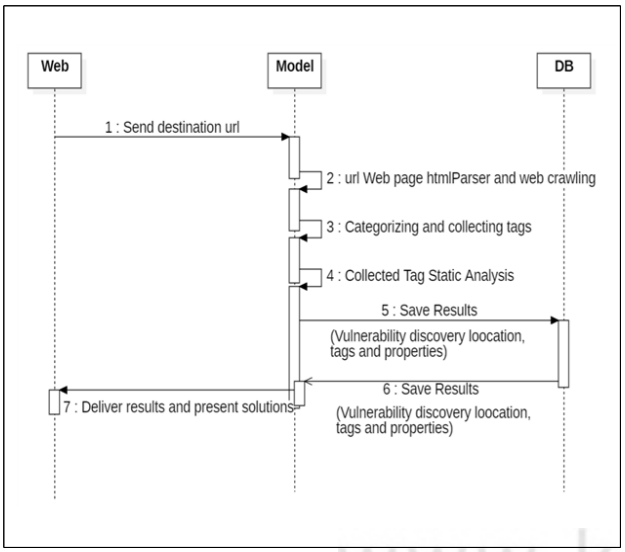

Fig. 1. Static Web Standard Check Model Sequence Diagram

첫 번째, 정적 웹 표준 체크 모델을 적용할 웹 페이지의 URL을 입력하여 선정한다. 선정된 웹 페이지 태그를 수집하기 위해 파이썬의 html.parser을 이용하여 태그를 수집한다.

두 번째, html.parser을 이용하여 수집한 태그들을 대상으로 해당 모델은 정적 분석을 한다.

세 번째, 정적 분석이 완료되면 정적 태그를 대상으로 한 정적 분석 결과를 DB에 저장한다.

마지막으로 사용자에게 정적 분석 결과를 보여주면서 해당 태그의 취약점을 알려준다.

3.1.1 Web Page Address Forwarding and Web Crawling

사용자는 취약점을 검사할 웹 페이지의 URL을 지정하여 웹 페이지에 전송한다. 이때 웹 페이지의 최신 상태의 소스로 확인하기 위해서 웹 크롤링을 진행한다. 먼저 웹 페이지의 htmlParser 분석을 완료하면 해당 페이지는 다음 페이지로 넘어갈 수 있는 하이퍼링크 태그인 <a> 태그를 조회한다. 그리고 다음 웹 페이지로 넘어갈 수 있는지 없는지 판단하고, 넘어가는 웹 페이지의 url이 분석한 웹 페이지들 url과 중복이 되는지 판단 후에 다음 페이지로 넘어간다. 최근 5년 이내에 플러그인이 제거된 공공기관 웹 페이지들을 대상으로 제안된 모델을 적용하여 최대 3 depth까지 증가시켜 크롤링을 진행한다.

3.1.2 Categorizing and collecting tags

정적 분석을 하기 전 분석 대상 태그 및 속성을 미리 정해야 한다.

본 논문에서는 대상 태그로 Clickjacking과 XSS를 선정했다. 정적 분석 대상 태그인 Clickjacking과 XSS는 대표적인 일반적인 웹 공격이다. Clickjacking은 웹 페이지의 정보를 유출할 수 있는 공격이고, XSS는 서버가 클라이언트에서 검증되지 않은 페이로드를 받아서 응답을 보내는 과정에서 발생한다[14-15].

정적 분석 대상 태그의 위험도 순위는 다음 Table.1과 같다.

Table 1. Static Analysis Target Tags and Properties

Tag	Ratio	No
input	67.1%	1
form	65.4%	2
button	30.2%	3
iframe	26.3%	4
select	16.1%	5

태그 위험도는 상위 50개 사이트를 대상으로 정적 대상 태그들을 사용현황을 조사하여 사용빈도가 많은 태그가 공격에 자주 노출됨으로 위험도가 높으므로 태그의 위험도 순위를 나타내었다[9].

정적 태그의 위험도 순위를 바탕으로 정적 분석 대상 태그 및 속성들은 다음 Table 2. 과 같다.

Table 2. Static Analysis Target Tags and Properties

Tag Type	Tag	Property
Clickjacking	iframe	sandbox/srcdoc
XSS	input/select/textarea/ button/keygen	autofocus/onbluer
	button/input	formation
	video/audio	source
	video	poster
	math	href
	iframe	srcdoc

위 Table 2. 에 있는 태그 및 속성들을 대상으로 htmlParser가 분석한다. 분석이 완료된 페이지는 태그들과 속성을 분류하고 다음 태그의 위치를 수집한다. 위의 두 태그 해결 방안으로 HTML 태그와 속성을 필터링해야 하기 때문에 기준으로 선정했다.[8]

3.1.3 Collected Tag Properties Static Analysis

분류된 태그 및 속성을 정적 분석한다. 분류 대상 태그 및 속성 중 XSS와 Clickjacking 공격 방법에 이용될 수 있는 태그 및 속성이 있는지 확인한다. 이러한 페이지 내의 태그와 속성을 검사하여 취약점이 있는지 분석한다.

다음 Fig. 2는 수집된 태그 및 속성을 정적 분석하는 알고리즘이다.

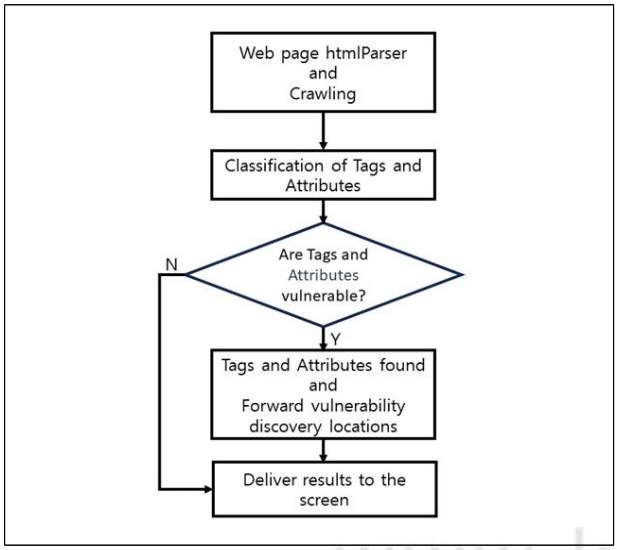

Fig. 2. Algorithms for static web standard check models

첫 번째, 웹 페이지 url을 htmlParser 한 뒤 크롤링을 진행한다. 그리고 크롤링한 웹 페이지에 대해 태그와 속성을 분류한다.

두 번째, 태그와 속성이 존재하는지 확인한 다음 취약점 부분이 발생한 태그와 속성의 발견된 위치를 찾는다.

마지막으로 취약점이 발견되면 그 결과를 화면에 취약점을 보여준다.

3.1.4 Static Analysis Results Output

정적 분석 대상 태그 표를 기준으로 정적 분석은 알고리즘의 의사코드는 다음 Fig. 3.와 같다.

```
def isValidUrl(url):
    url parsing
    return

def getLink(url, soup):
    Link Set Initialization
    for all 'a' Tag:
        Convert relative paths to absolute paths
and add them to a set of links
    return Link Set

def scanPage(url, depth=3):
    if the depth is zero or the URL is invalid:
        return empaty list
    try:
        HTML parsing on successful request
        Perform vulnerability checks
        Page scan recursive calls for all links
    except request failed:
       return empaty list
    return vulnerabilities list

def chkVulnerabilities(soup):
    Initialize vulnerability list
    for tags:
       if vulnerable properties exist add to
vulnerability list
    return vulnerabilities list
```

Fig. 3. Static Analysis Algorithm Pseudocode

첫 번째, 취약점 분석을 할 url 변수에 url이 올바른 형식인지 확인하고 정의한다.

두 번째, 취약점 분석을 위해 주어진 웹 페이지에서 모든 하이퍼링크를 찾기 위해 정의한 a 태그를 찾아 절대 url로 변환 후 반환한다. 그리고 주어진 url을 시작하여 웹 페이지를 스캔하고 취약점을 검사한다. 그리고 스캔하는 웹 페이지의 depth의 수는 지정된 3 depth까지 웹 페이지의 링크를 따라 각 페이지에 대한 취약점을 검사하고

depth가 0이거나 UL이 유효하지 않으면 빈 목록을 반환한다.

세 번째, 웹 페이지의 HTML을 가져오고 정적 분석만 진행하기 때문에 Beautiful Soup을 사용하여 구문분석한다. 취약점 확인 함수(chkVulnerabilities)를 호출하여 현재 페이지의 취약점을 검사한다. 취약점이 발생하면 발견한 정보를 목록에 추가한다.

마지막으로 취약점 목록을 출력하면서 그에 맞는 해결책을 제시한다.

위의 내용을 바탕으로 정적 분석 대상 태그 정적 분석의 알고리즘의 플로우 차트는 다음 Fig. 4.와 같다.

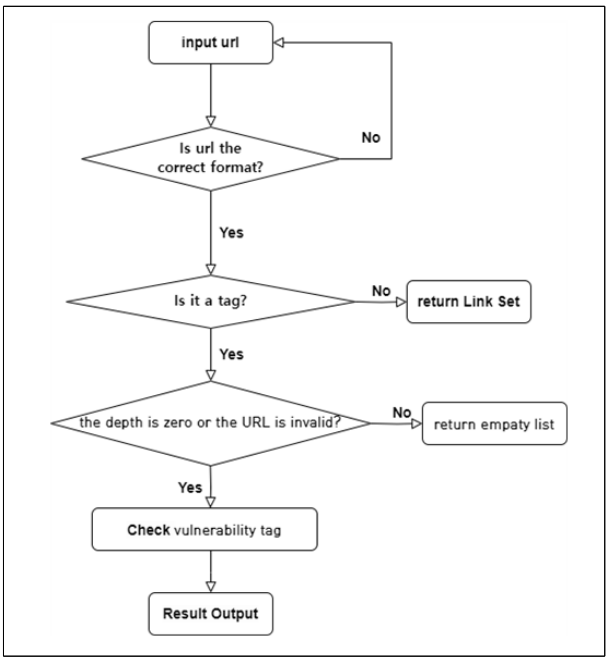

Fig. 4. Static Analysis Algorithm Flowchart

IV. Evaluation

본 논문에서는 제안된 모델을 검증하기 위해 웹 표준화를 진행하기 위해 최근 5년 이내에 플러그인이 제거된 공공기관 웹 페이지들을 대상으로 제안된 모델을 적용했다. 과거 HTML로 구성된 웹 페이지를 HTML5 기반으로 개선 작업이 이루어진 공공기관 사이트 7개를 선정하여 평가했다. 공공기관 사이트 7개를 분석한 결과 취약점은 다음 Table 3. 와 같다.

Table 3. Number of vulnerabilities after static analysis

Site	Vulnerability Count
A	2
B	186
C	0
D	1
E	0
F	0
G	3

취약점 개수는 A 사이트 2개의 취약점 B 사이트 186개 취약점 E 사이트 1개의 취약점이 나왔다. 이는 웹 표준에 맞춰 변환을 하였지만, 현재까지도 취약점이 있다는 것을 보여준다. 이 중 186개의 취약점이 나온 B 사이트는 신청을 입력하는 서비스가 다른 사이트에 비해 더 많이 사용되고 있었다. 이는 input 태그를 많이 사용하는 것을 나타내며 해당 웹 사이트가 다른 사이트에 비해 취약점에 더 노출되어 있다는 것을 보여준다.

정적 분석 후 각 태그의 속성별 태그의 취약점의 개수는 다음 Table 4.과 같다.

Table 4. Number of vulnerabilities in tags by property after static analysis

Tag Type	Tag	Property	Number
Clickjacking	iframe	sandbox /srcdoc	0
XSS	input	autofocus /onbluer	192
	select		0
	textarea		0
	button		0
	keygen		0
	button	formation	0
	input		0
	video	source	0
	audio		0
	video	poster	0
	math	href	0
	iframe	srcdoc	0

분석 결과 클릭재킹(clickjacking)의 악성 유형에 관한 취약점은 발견되지 않았다. 그러나 XSS의 input태그의 autofocus/onbluer 속성이 현재까지도 사용되고 있다.

이는 현재까지도 일반적인 웹 공격인 XSS 공격에 대해서 취약한 것을 보여준다. 이러한 XSS 취약점에 대해 대응하기 위해서는 첫 번째 태그와 속성 문자를 필터링을 하는 것이다. 두 번째 세션 히스토리 목록 삭제 및 히스토리의 최대로 저장이 가능한 개수를 조정하는 것이다. 그 외에 나머지 autofocus/onbluer 속성들을 사용하고 있는 select, textarea, button, keygen 태그들은 이상이 없었

다. button과 input 태그의 formation 속성, video와 audio 태그의 source 속성, video태그의 poster 속성, math 태그의 href 속성, iframe 태그의 srcdoc 속성 모두 이상이 없는 것이 확인했다.

V. Conclusions

HTML5를 기반으로 한 웹 표준의 등장은 기존 HTML에 비해 다양한 요소와 속성이 추가된 태그와 web API를 통해 업그레이드된 웹 환경을 제공했다. 그러나 웹 표준을 적용한 뒤 취약점 점검 프로그램을 적용했으나 웹 페이지에서도 새로운 보안 취약점이 발생하는 것을 보여준다.

본 논문은 웹 페이지가 HTML5를 기반으로 한 웹 표준의 태그 및 속성을 검사하고 보안 취약점을 찾을 수 있는 정적 웹 표준 체크 모델을 제안한다. 제안된 모델을 공공기관 웹 사이트 중 7개를 선정하여 적용한 결과 보안 취약점의 XSS 공격 중 input 태그의 취약점 192개를 발견했다.

본 논문에서 제안한 정적 웹 표준 체크 모델을 웹 개발 후 테스트 과정에서 적용한다면 웹의 취약점 및 보안 약점을 웹 페이지 서비스 전에 발견할 수 있고, 수정함으로써 신뢰성 있는 웹 페이지를 제공하는 데 도움을 줄 수 있다. 그리고 동적 분석을 하기 전 여러 보안 약점에 대해 미리 대응이 가능한 장점이 있다. 그러나 컴포넌트 간 발생할 수 있는 보안 문제점 및 설계, 구조 관점의 보안 약점의 발견이 제한적인 단점이 있다. 향후 연구에서는 정적 분석 기법과 상호 보완적 기능을 수행하는 동적 분석을 추가하여 웹의 보안 문제를 해결할 수 있는 모델을 연구할 예정이다.

REFERENCES

[1] Microsoft, Internet Explorer 11 End desktop application support, https://learn.microsoft.com/ko-kr/lifecycle/announcements/internet-explorer-11-end-of-support

[2] Choi, Eun Woo, “An Alternative Method of ActiveX using WebSocket ”,The Graduate School of Engineering Hanyang University, pp.2-3, 2016.

[3] Kim GunWoo, and Park MinHo, "A Study on Secure Non-ActiveX Implementation," Proceedings of Symposium of the Korean Institute of communications and Information Sciences, pp. 309-310, 2020.

[4] Ministry of Public Administration and Security, Removing the Public Website Plug-in Guidelines, https://www.mois.go.kr/frt/bbs/type001/commonSelectBoardArticle.do?bbsId=BBSMSTR_000000000045&nttId=72778

[5] Moon-Saog Jun, So-Hee Kim, Han-Na You, and Keun-Chang Choi, "The Design proxy signature protocol for transaction information non-repudiation," The Korea Academia-Industrial cooperation Society a collection of papers for academic conferences, pp. 522-524, 2014.

[6] Jeong, Chang-Jin, "Trends in Standards and Testing Certification Technology - W3C Social Web Standardization Trends," Telecommunications Technology Association, pp. 87-91, 2016.

[7] So-Young Chae, Young-suk Lee, Jung-A Kim, and Chang-Yong Lee, "An analysis of web pages‘ compliance with web standard by using web standard validator," Korean Institute of Information Scientists and Engineers a collection of academic papers, pp. 1527-1529, 2015.

[8] Bo Ram Lee. "A Study on Method to Non-ActiveX Implement for Enhanced Security ." Domestic Master's Thesis Soongsil University Graduate School of Information Science, 2015, seoul

[9] Mi Young Bae. "A study on HTML5 security fragility analysis and inspection technique." Domestic Master's Thesis Andong University, 2016. Gyeongsangbuk-do

[10] Kang Sok-Chul, and Park Jeong-Seop, "Security Issues in HTML5 Next Generation Web Standard Environments," REVIEW OF KIISC, Vol. 24, No. 4, pp. 44-55, 2014.

[11] Jin-Hwan Kim, and Eun-Gyung Kim, "WCTT: Web Crawling System based on HTML Document Formalization," Journal of the Korea Institute of Information and Communication Engineering, Vol. 26, No. 4, pp. 495-502, 2022.

[12] Lee-JuHyeok, Kim-Mi Hui, “Image Classification Model using web crawling and transfer learning”, j.inst.Korean.electr.electron.eng , Vol.26, No.4 ,pp639~646 , December 2022

[13] Seungyong Choo, Yeseong Hwang, Sangjin Lee, “Methods for Collecting Harmful Websites Using Web Crawling”, Journal of Digital Forensics, Vo15, No 3, pp127-138, September 2021. DOI : 10.22798/kdfs.2021.15.3.127

[14] Kwon Min Hee, Bae Han Cheol, Kim Hwan-Kuk, “A Study on HTML5 Mobile Web pages Attack”, KICS Winter Conference 2017, pp1,150-1,151, january 2017.

[15] Juchan Kim1, Jihwan Moon, “An Empirical Investigation of a Cross Site Scripting (XSS) Attack With Bypassing a Blacklist of a Web Application”, Summer Annual Conference of IEIE 2023, pp 2,056 - 2,060, June. 2023

Authors

Hee-Yeon Won received the B.S. degrees in Computer Science Engineering from Kongju National University, Cheonan in 2020 respectively. She is currently pursuing a M.S.. degree in Computer Science Engineering from Kongju National University, Cheonan. She is interested in Web Standard Security, Web Security Analysis, algorithms.

Jae-Woong Kim received the bachelor's degree and the M.S. degree in the Department of Computer Engineering from the Jungang University in 1983 and 1988, respectively.

He received the Ph.D. degree in the Department of Computer Engineering from Daejun University in 2000. He has been a professor in the Department of Computer Engineering at Kongju National University since 1992. His current research interests include software engineering.

Young-Suk Chung received the M. S. degree in Multimedia Engineering from Kongju national university, in 2009. Ph. D degree in Computer Engineering from Kongju national university in 2013.

He is currently an adjunct professor in Daejeon Health Sciences College. He is interested in Big data, Cloud computing, Simulation, A.I and Predictive modeling.