Enact browser 보안 취약점 진단 툴 개발

추진 목적

Enact browser의 보안 취약점을 드러내고 해결방안을 모색해본다. (보안 취약점이란??)

향후 지속적인 보안 리스크 모니터링을 위한 툴을 개발한다. (툴의 실행방식 및 출력결과??)

진행 과정에서 학습한 브라우저 보안 개론에 대한 내용을 정리한다. (정리가 꼭 필요한 내용??)

진행 단계 (6단계)

<< 배경 지식 스터디 >>

1. 브라우저 및 웹 기술 이해

https://web.dev/

https://developers.google.com/web/updates/2018/09/inside-browser-part1

2. 브라우저 보안 기술 이해

https://web.dev/secure/

https://web.stanford.edu/class/cs253/

<< 개발 환경 구축 >>

1. webOS OSE (라즈베리파이4 기기, 우분투 환경에서 빌드)

- 우분투 환경에서 코드 다운로드 및 빌드 진행 필요
- 라즈베리파이4 기기에 빌드된 이미지를 올려서 실행

2. Selenium (Python)

- 브라우저 기반 자동화 테스트 수행을 위한 프레임워크 (Win10에서도 활용 가능)
- 사용자가 특정 웹 페이지에 접속해서 버튼 클릭 및 화면 조회 등의 자동화 가능

3. Chrome (+DevTools)

- 크로미움 직접 빌드 및 실행

  https://www.chromium.org/developers/how-tos/get-the-code

- 이전 버전 다운로드 링크

  https://www.slimjet.com/chrome/google-chrome-old-version.php

  https://google_chrome.en.downloadastro.com/old_versions/

<< 보안 취약점 도출 >>

기본적으로 Enact browser의 기반 엔진은 크로미움으로 크롬과 동일함

하지만 configuration(build flags, runtime flags) 설정에 따라 차이가 발생

https://peter.sh/experiments/chromium-command-line-switches/

webOS에서는 다음 Flag를 활용해 오고 있음 (Chrome에서는 활용 X)

(--disable-web-security, --no-sandbox, --disable-site-isolation-trials)

BrowserAudit 툴을 통해 브라우저 보안 수준에 대한 리포트 서비스(무료) 활용 가능

https://browseraudit.com/

상기 정보를 참고하여 Enact browser의 보안 취약점을 정리하고 각각의 리스크 수준 도출 필요

<< 보안 취약점 해결 >>

Chrome 대비 Enact browser의 보안 취약점 각각에 대한 해결 방안 정리

해결 방안 도출이 어려운 경우, 각 취약점으로 인해 발생 가능한 보안 위협 시나리오 정리

이 때, 시나리오에 구체적인 재현 경로가 기술되면 좋음

<< 자동화 탐지 툴 개발 >>

해결 방안이 정리된 건에 대한 자동화 탐지 툴 개발

문제점을 드러내는 웹 앱(HTML, JavaScript)을 간단히 만들고, 이를 자동 수행 및 리포트 발행해야 함

<< 최종 보고서 작성 >>

위의 과정들을 수행하는 과정에서 이해하고 산출된 내용들을 최종 보고서 및 발표자료에 수록

주요 일정 (6단계)

1. 배경 지식 스터디 (10/16 금, 4PM)

진행 중 막히는 부분 Q&A

브라우저 보안 기술에 대한 개론 세미나 (30 mins)

Selenium 기반 TC 예제 설명 및 코드 공유

2. 중간 현황 공유 (11/20 금, 4PM)

진행 중 막히는 부분 Q&A

사전 지원 요청해 주신 부분에 대한 F/U

3. 최종 발표 (12/17 목, 4PM)

최종 발표 자료 내용 공유

간담회 (리포트에 담기지 않은 정성적인 의견 수렴)

[참고] Enact browser UI