브라우저 보안 정책을 우회 해 보자

(Bypassing Browser Security Polices)

김동현(Donghyun, Kim, hackpupu)[1]

[1] *Korea University Graduate School, Seoul, Republic of Korea, ©i2sec*
hackpupu@gmail.com, kdh@i2sec.co.lr

Abstract: 각종 브라우저 보안 정책 기법과 지금까지 발표된 재미있는 우회 테크닉 및 브라우저에서 보안 정책을 적용한 내부적인 테크닉에 대해 소개한다. SOP(Same Origin Policy), CSP(Content Security Policy) 우회 기법과 UXSS(Universal Cross Site Scripting) 테크닉 등에 대해 소개하며, 최종적으로 안드로이드와 Edge, IE 브라우저에서 발견된 1-Day 취약점에 대해 파고든다.

Keyword: Browser Security, SOP, X-XSS-Protection, Browser 1-Day, Android Browser, Edge, IE, Browser Polices By Pass

Content

브라우저에는 각종 보안 정책이 있다. 이런 정책들은 웹 서비스를 이용할 시 조금 더 안전한 환경을 제공하기 위해 나타났으며, 여러 기능을 보완할 수 있는 여러가지 정책이 존재하고 있다.

먼저, SOP 를 소개하자면, 풀 네임은 Same Origin Policy 이다. 한 출처에서 로드 된 문서나 스크립트가 다른 출처 자원과 상호작용 하지 못하게 제한하는 정책으로, 타 출처의 스크립트나 자원을 사용할 수 없게 강제하는 정책이다. 여기서 출처는 도메인 명, 사용 프로토콜(HTTP/HTTPS), 사용하는 포트 등이 같으면 같은 출처, 하나라도 다르면 다른 출처라고 인식하게 된다. 이는 공격자 입장에서 XSS 또는 CSRF 등의 공격을 진행할 때 한단계 귀찮은 작업을 하게 하는 조금 골치 아픈 정책이다.

위와 같은 이유로, 공격자는 보안 정책을 우회하기 위한 연구를 시작했으며, 여기서 재미있는 테크닉들이 많이 발견되었다.

SOP 를 재미난 기법으로 우회한 사례를 먼저 소개하고자 한다.

BrokenBrowser 팀에서 블로그 포스트를 통해 공개하였으며, Edge 의 Reading mode 기능에서 취약점이 발견되었다. Reading mode 기능은 MS Edge 팀에서는 4 월 14 일날 트위터를 통해 Reading Mode 에 대해 공개했다.대부분이 알고있을 기능이라 생각하지만, 생소한 사람들을 위해 간략히 소개하자면 웹 페이지의 문서를 읽을 때 가독성을 높이기 위한 기능으로, 모바일 브라우저에서 먼저 기능이 도입되었고 각종 이미지와 Object 태그 등을 제거해 글만 편하게 읽을 수 있는 기능이다. 다만, 브라우저마다 읽기모드에 대한 구현방안은 조금씩 다를 것이다.

지금 살펴보고있는 Edge 에 대해 조금 보자면 iframe 태그, script 태그 및 기타 가독성을 침범하는 HTML 태그등을 제거하고 내부 읽기 리소스(C:₩Windows₩SystemApps₩Microsoft.MicrosoftEdge_8wekyb3d8bbwe₩Assets₩ReadingView)에서 호스팅 되는 HTML 으로 iframe 태그를 이용해 읽게 된다.

* 위 과정들은 Background 에서 발생하며, Edge 를 사용하는 유저들은 "주소 표시줄"이 변경되지 않기에 원본 웹 사이트에 있다고 생각하게 된다.

그럼, "현재 로딩하고 있는 페이지는 어떤 URL 을 가지고 있는가?"에 대해 의문이 생긴다. 브라우저에서 내장하고 있는 Dev Tools 를 이용하면 매우 편하게 의문에 대해 해결할 수 있다. Console 에서 location.href Jascript 를 호출하면 확인할 수 있다.
확인하면 "read: 원본 Document URL"을 가지고 있음을 보고 있다.

* 위 내용을 보며 난 생각이 들었다. Reading mode 의 경우 SOP 에 대한 정책을 전혀 고려하지 않고 개발이 되었구나!

그러면 Edge 의 Reading Mode 를 강제로 호출하면 주소표시줄은 어떻게 표기하는가? 를 역으로 생각해볼 필요가 있다. location.href = "read:아무 URL"을 Script 로 호출하면 주소 표시줄은 "아무 URL"을 표시하고 있다. 고로, Edge 의 Reading Mode 에서는 스크립트나 HTTP 리다이렉션이 발생했음에도 주소 표시줄을 업데이트 하고 있지 않음을 볼 수 있다.

조금 쉽게 취약점을 테스트하기 위해, 구글의 검색 리다이렉트 기능을 이용해 확인을 해 보자.

* 아래부터는 Edge 버젼에 대해 Dependency 함(05/14 기준 동작하지 않음. 이전 버전의 Edge 가 필요)

읽기모드가 되는 URL 을 구글의 리다이렉트 기능을 통해 Reading 프로토콜을 적용해 연다면 HTML 다큐먼트는 구글이 아닌 읽기모드가 되는 URL 을 표시하고 있으나, URL 과 location.href 의 경우 구글을 가르키고 있음을 확인할 수 있다.

* 이제 취약점 검증과 타 URL 로 Spoofing 할 수 있음을 확인 했다.

이제 Spoofing 에 성공했으니 다음순서인 스크립트를 실행시키고 싶어 진다.

하지만 앞서 설명했듯 읽기모드에서는 iframe 태그나 기타 javascript 태그를 필터링하고 있다. 이제 고전적으로 XSS 를 찾듯 여러가지 태그를 대입해 볼 차례이다.

결론적으로 말하면 Edge 에서는 object 태그를 Reading 모드에서 필터링하고 있지 않다. 그래서 아래와 같은 스크립트를 실행시킬 수 있다.

"http://*/prompt.html"에서는 prompt 를 띄우는 자바스크립트를 입력해놓고 원본 URL 에서 아래와 같은 HTML 을 작성한다.

<object data="http://*/prompt.html>

그러면 정상적으로 Prompt 가 실행되며, google.com 에서 입력을 받는듯한 액션으로 사용자에게 Spoofing 할 수 있다.

조금 더 생각해 볼 부분이, "조금 더 효율적인 공격은 어떻게 해야 할까?"란 질문이 나온다.

사실 Edge 의 Reading 모드를 적용해보면 알겠지만 백그라운드 색깔이 티가 나게 다르며 Document 를 직접 수정해보고 싶은 마음이 든다.

Object 태그를 통해 Script 를 정상적으로 실행시킬 수 있음을 인지하고, 아래와 같은 형식으로 SOP 를 우회해 Google 의 실제 Document 를 수정하는 듯한 행위를 할 수 있다.

```
<object data="data:,<script>
window.onload = function()
{

  document.write(
    '<script>'+
        'top.document.write(₩'Trust me, we are on Google =)₩');'+
        'top.document.close()'+
```

```
    '<₩/script>');
  document.close();
}
</script>"></object>
```

이로써 Edge 의 Reading Mode 에서 발견된 SOP Bypass 사례 소개를 끝낸다.

브라우저의 추가기능들에 대해 연구하고 그로 인해 조금 더 재미난 보안 취약점 및 테크닉이 계속 나오기 기대한다.

*** 추가적으로 설명할 사례는 현재 정리가 완료되지 않음.

*** 실제 발표시에는 조금 더 재미난 기법, 흥미로운 우회 테크닉 등을 소개할 수 있다.

References

1. X-XSS-Nightmare : 1 ; mode=attack XSS Attacks Exploiting XSS Filter
   https://www.slideshare.net/masatokinugawa/xxn-en
2. JSMVCOMFG – To sternly look at JavaScript MVC and Templating Framework
   https://www.slideshare.net/x00mario/jsmvcomfg-to-sternly-look-at-javascript-mvc-and-templating-frameworks/48
3. CSP Bypass in Chrome Canary + AngularJS
   https://html5sec.org/cspbypass/
4. Blackhat Asia 2016 – Bypassing Browser Security Polices For Fun and Profit
   https://www.blackhat.com/docs/asia-16/materials/asia-16-Baloch-Bypassing-Browser-Security-Policies-For-Fun-And-Profit.pdf
5. Broken Browser – sop bypass courtesy of the reading mode(Edge)
   https://www.brokenbrowser.com/sop-bypass-abusing-read-protocol/
6. OWASP Secure Headers Project
   https://www.owasp.org/index.php/OWASP_Secure_Headers_Project
7. Broken Browser – SOP BYPASS / UXSS – Adventures in a domainless world(Edge)
   https://www.brokenbrowser.com/uxss-edge-domainless-world/