XSS(Cross-Site Scripting)란?

크로스 사이트 스크립팅(사이트 간 스크립팅 )은 [SQL injection](https://namu.wiki/w/SQL%20injection)과 함께 웹 상에서 가장 기초적인 취약점 공격 방법의 일종으로, 악의적인 사용자가 공격하려는 사이트에 스크립트를 넣는 기법을 말한다. 주로 다른 웹사이트와 정보를 교환하는 식으로 작동하므로 사이트 간 스크립팅이라고 명칭한다.

이 취약점은 웹 애플리케이션이 사용자로부터 입력 받은 값을 제대로 검사하지 않고 사용할 경우 나타나며, 공격에 성공하면 사이트에 접속한 사용자는 삽입된 코드를 실행하게 된다. 보통 의도치 않은 행동을 수행시키거나 쿠키나 세션 토큰 등의 민감한 정보를 탈취한다.

크로스 사이트 스크립팅은 자바스크립트를 사용하여 공격하는 경우가 많다. 공격 방법이 단순하고 가장 기초적이지만, 많은 웹사이트들이 XSS에 대한 방어 조치를 해두지 않아 공격을 받는 경우가 많다. 여러 사용자가 접근 가능한 게시판 등에 코드를 삽입하는 경우도 많으며, 경우에 따라서는 메일과 같은 매체, 심지어는 닉네임에 코드를 심기도 한다.

공격 방법에 따라 Stored XSS와 Reflected XSS로 나뉜다. Stored XSS는 사이트 게시판이나 댓글, 닉네임 등 스크립트가 서버에 저장되어 실행되는 방식이고, Reflected XSS는 보통 URL 파라미터(특히 GET 방식)에 스크립트를 넣어 서버에 저장하지 않고 그 즉시 스크립트를 만드는 방식이다. 후술된 내용 대부분은 Stored XSS라고 생각하면 된다. Reflected XSS의 경우 브라우저 자체에서 차단하는 경우가 많아 상대적으로 공격을 성공시키기 어렵다.

CSRF(Cross-Site Request Forgery)란?

사이트 간 요청 위조(또는 크로스 사이트 요청 위조, CSRF, XSRF)는 웹 사이트 취약점 공격의 하나로,

사용자가 자신의 의지와는 무관하게 공격자가 의도한 행위(수정, 삭제, 등록 등)를 특정 웹사이트에 요청하게 하는 공격을 말한다.

유명 경매 사이트인 옥션에서 발생한 개인정보 유출 사건에서 사용된 공격 방식 중 하나다.

[사이트 간 스크립팅](https://ko.wikipedia.org/wiki/%EC%82%AC%EC%9D%B4%ED%8A%B8_%EA%B0%84_%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%8C%85)(XSS)을 이용한 공격이 사용자가 특정 웹사이트를 신용하는 점을 노린 것이라면, 사이트간 요청 위조는 특정 웹사이트가 사용자의 [웹 브라우저](https://ko.wikipedia.org/wiki/%EC%9B%B9_%EB%B8%8C%EB%9D%BC%EC%9A%B0%EC%A0%80)를 신용하는 상태를 노린 것이다. 일단 사용자가 웹사이트에 [로그인](https://ko.wikipedia.org/wiki/%EB%A1%9C%EA%B7%B8%EC%9D%B8)한 상태에서 사이트간 요청 위조 공격 코드가 삽입된 페이지를 열면, 공격 대상이 되는 웹사이트는 위조된 공격 명령이 믿을 수 있는 사용자로부터 발송된 것으로 판단하게 되어 공격에 노출된다.

CSRF는 공격자가 사용자의 컴퓨터를 감염시키거나 사이트 해킹을 해서 이뤄지는 공격이 아니므로 공격이 성공하려면 다음 조건을 만족해야 한다.

1. 위조 요청을 전송하는 서비스(ex : 페이스북)에 희생자가 로그인 상태여야함.

2. 희생자는 공격자가 만든 피싱 사이트에 접속해야함.

XSS vs CSRF 비교

| ▶ XSS(Cross-Site Scripting) | ▶ CSRF(Cross-Site Request Forgery) |
| --- | --- |
| ○ 개요 :  악성 스크립트가 클라이언트에서 실행됨  ○ 공격 대상 : 클라이언트  ○ 목적 :  쿠키ㆍ세션 갈취, 웹사이트 변조 등 | ○ 개요 :  권한을 도용당한 클라이언트가  가짜 요청을 서버에 전송  ○ 공격 대상 : 서버  ○ 목적 : 권한 도용(위조:forgery) |

XSS공격과 CSRF는 사용자(피해자)의 브라우저를 목표로 하는 비슷한 공격이지만  
XSS는 사이트변조나 백도어를 통해 클라이언트에 대한 악성공격을 하고,  
인증된 세션이 없어도 공격을 할 수 있으며, 자바 스크립트를 실행 시키는 공격.

CSRF는 요청을 위조하여 사용자의 권한을 이용해 서버에 대한 악성공격을 하고,  
인증된 세션을 악용하며, 특정한 행동을 실행 시키는 공격이라는 점에서  
두가지 공격의 차이가 있다.

방지대책

| XSS의 방지 대책 | CSRF의 방지 대책 |
| --- | --- |
| 1. 쿠키에 중요한 정보를 담지 않고 서버에 중요정보를 저장하는 방법 2. 정보를 암호화 하는 방법 3. httponly 속성 on (자바스크립트의 document.cookie를 이용해서 쿠키에 접속하는 것을 막는 옵션으로, 쿠키를 훔쳐가는 행위를 막기 위한 방법이다.) 4. Secure coding (str\_replace <치환함수로 XSS방지 대책이지만 미흡한 부분이 있다 -> url encoding>htmlspecialchars <html 엔터티로 변환해줌> ..등 보안과 관련된 함수를 사용한다.)  ▶ Encoding : 특정 플랫폼에서 문자를 표현하기 위한 규약 | 1. Referrer 검증 (Back -end 단에서 request 의 referrer을 확인하여 domain이 일치하는지 검증하는 방법) 2. Security Token 사용 (Reffer검증이 불가한 환경일 시, Security Token을 활용한다. 사용자 세션에 임의의 난수 값을 저장하고, 사용자의 요청마다 해당 난수 값을 포함 시켜 전송한다.이후 Back -end 단에서 요청을 받을 떄마다 세션에 저장된 토큰 값과 요청 파라미터에 전달되는 토큰 값이 일치한지 검증하는 방법) 두 방법 모두 같은 도메인 내에 XSS취약점이 있다면 CSRF 공격에 취약 해질 수 있다. |