**크로스 사이트 스크립팅 공격**

- 공격자가 악의적인 스크립트 코드를 웹 애플리케이션에 삽입한 후, 웹 사용자의 웹 브라우저에서 해당 코드가 실행되도록 만드는 공격

- 다른 공격들은 취약점을 가지고 있는 서버 쪽 공격

- 이 공격은 서버의 취약점을 이용하여 클라이언트 쪽 공격

- 예시 중 하나로 세션 쿠키 탈취가 있음 -> 웹 브라우저에 있는 세션 쿠키를 자바스크립트를 이용하여 읽음

- 자바스크립트 코드에 따라서 다양한 형태의 공격 시도 가능

- 실제 취약점을 가지고 있는 웹 서버와 상관없는 웹 사이트의 정보도 노릴 수 있음

- 스크립트 코드의 실행 시기에 따라 리플렉티드(reflected) 또는 스토어드(stored) 크로스 사이트 스크립팅 공격으로 구분할 수 있음.

**1. 리플렉티드 크로스 사이트 스크립팅(Reflected XSS) 공격 개요**

- 웹 브라우저에 사용자가 입력한 결과가 출력되는 경우, 적절한 입력값 검증 또는 대응 방안이 없음

-> 사용자가 <script>와 같은 스크립트 태그를 포함한 자바스크립트 코드를 입력

-> 웹 페이지에 이 스크립트가 그대로 출력

-> 웹 브라우저는 사용자에 의해 입력된 자바스크립트와 기존 웹 애플리케이션 개발자가 의도한 자바스크립트 구별 불가

-> 사용자가 입력한 자바스크립트 실행

- 요청 메시지에 입력된 스크립트 코드가 즉시 응답 메시지를 통해 출력되는 취약점

- 게시판에 글을 남기거나 이메일 피싱을 이용하여 악의적인 스크립트 코드가 담긴 요청을 사용자가 실행하도록 만듦.

텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

**<이메일 피싱 공격 예시>**

1) 공격자가 사용자에게 이메일 등으로 피싱. (포털 사이트의 게시판, SNS 등 공격자가 링크를 남길 수 있으면 모두 가능) 피싱 시, 세션 쿠키 탈취 스크립트 코드를 삽입한 HTTP 요청 링크 포함

2) 사용자가 링크 클릭 -> 스크립트 코드가 삽입된 요청이 웹 사이트로 전송

3) 리플렉티드 크로스 사이트 스크립팅 취약점이 있는 웹 사이트는 입력된 스크립트를 반사하여 웹 페이지에 출력

4) 웹 페이지를 읽는 웹 브라우저는 자동으로 스크립트 실행 -> 그 결과인 세션 쿠키를 공격자에게 전달

5) 공격자가 이 세션 쿠키를 사용하여 해당 사용자의 권한으로 접속

**2. 리플렉티드 크로스 사이트 스크립팅(Reflected XSS) 공격 실습**

텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

<보안 레벨 – Low / XSS (Reflected)>

- 해당 페이지는 이름을 물어보는 폼 표시

- 입력란에 이름을 입력하면 다음과 같은 문구 출력

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

- Hello 뒤에 입력 값이 그대로 다시 출력

- 어떤 웹 애플리케이션이 사용자가 입력한 값을 그대로 출력하는 경우, 리플렉티드 크로스 사이트 스크립팅 취약점이 존재할 가능성 높음

- 해당 취약점이 존재하는지 확인하기 가장 간단한 방법은 스크립트 태그 입력해보는 것

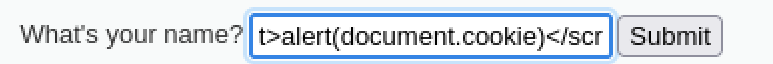


텍스트, 스크린샷, 폰트, 멀티미디어이(가) 표시된 사진

자동 생성된 설명

- 자바스크립트 함수 alert()에 의해 1이란 입력 값 출력 -> 입력한 자바스크립트가 웹 브라우저에서 실행

- 스크립트를 입력할 수 있고, 입력된 스크립트가 응답되어 리플렉티드 크로스 사이트 스크립팅 취약점이 존재한다는 것을 확인



텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

- 스크립트를 위와 같이 입력하면, 쿠키를 출력

- PHPSESSID는 PHP에서 생성하는 세션 쿠키

- 크로스 사이트 스크립팅 공격은 주로 세션 쿠키를 알아내는 데 초점