CSRF & XSS

CSRF

> Cross Site Request Forgery

웹 어플리케이션 취약점 중 하나로, 인터넷 사용자가 자신의 의지와는 무관하게 공격자가 의도한 행위 (modify, delete, register 등)를 특정한 웹사이트에 request하도록 만드는 공격을 말한다.

주로 해커들이 많이 이용하는 것으로, 유저의 권한을 도용해 중요한 기능을 실행하도록 한다.

우리가 실생활에서 CSRF 공격을 볼 수 있는 건, 해커가 사용자의 SNS 계정으로 광고성 글을 올리는 것이다.

정확히 말하면, CSRF는 해커가 사용자 컴퓨터를 감염시거나 서버를 해킹해서 공격하는 것이 아니다. CSRF 공격은 아래와 같은 조건이 만족할 때 실행된다.

- 사용자가 해커가 만든 피싱 사이트에 접속한 경우
- 위조 요청을 전송하는 서비스에 사용자가 로그인을 한 상황

보통 자동 로그인을 해둔 경우에 이런 피싱 사이트에 접속하게 되면서 피해를 입는 경우가 많다. 또한, 해커가 XSS 공격을 성공시킨 사이트라면, 피싱 사이트가 아니더라도 CSRF 공격이 이루어질 수 있다.

대응 기법

- **리퍼러(Refferer) 검증**

  백엔드 단에서 Refferer 검증을 통해 승인된 도메인으로 요청시에만 처리하도록 한다.

- **Security Token 사용**

  사용자의 세션에 임의의 난수 값을 저장하고, 사용자의 요청시 해당 값을 포함하여 전송시킨다. 백엔드 단에서는 요청을 받을 때 세션에 저장된 토큰값과 요청 파라미터로 전달받는 토큰 값이 일치하는 지 검증 과정을 거치는 방법이다.

> 하지만, XSS에 취약점이 있다면 공격을 받을 수도 있다.

XSS

> Cross Site Scription

CSRF와 같이 웹 어플리케이션 취약점 중 하나로, 관리자가 아닌 권한이 없는 사용자가 웹 사이트에 스크립트를 삽입하는 공격 기법을 말한다.

악의적으로 스크립트를 삽입하여 이를 열람한 사용자의 쿠키가 해커에게 전송시키며, 이 탈취한 쿠키를 통해 세션 하이재킹 공격을 한다. 해커는 세션ID를 가진 쿠키로 사용자의 계정에 로그인이 가능해지는 것이다.

공격 종류로는 지속성, 반사형, DOM 기반 XSS 등이 있다.

- **지속성** : 말 그대로 지속적으로 피해를 입히는 유형으로, XSS 취약점이 존재하는 웹 어플리케이션에 악성 스크립트를 삽입하여 열람한 사용자의 쿠키를 탈취하거나 리다이렉션 시키는 공격을 한다. 이때 삽입된 스크립트를 데이터베이스에 저장시켜 지속적으로 공격을 하기 때문에 Persistent XSS라고 불린다.
- **반사형** : 사용자에게 입력 받은 값을 서버에서 되돌려 주는 곳에서 발생한다. 공격자는 악의 스크립트와 함께 URL을 사용자에게 누르도록 유도하고, 누른 사용자는 이 스크립트가 실행되어 공격을 당하게 되는 유형이다.
- **DOM 기반** : 악성 스크립트가 포함된 URL을 사용자가 요청하게 되면서 브라우저를 해석하는 단계에서 발생하는 공격이다. 이 스크립트로 인해 클라이언트 측 코드가 원래 의도와 다르게 실행된다. 이는 다른 XSS 공격과는 달리 서버 측에서 탐지가 어렵다.

대응 기법

- **입출력 값 검증**

  XSS Cheat Sheet에 대한 필터 목록을 만들어 모든 Cheat Sheet에 대한 대응을 가능하도록 사전에 대비한다. XSS 필터링을 적용 후 스크립트가 실행되는지 직접 테스트 과정을 거쳐볼 수도 있다,

- **XSS 방어 라이브러리, 확장앱**

  Anti XSS 라이브러리를 제공해주는 회사들이 많다. 이 라이브러리는 서버단에서 추가하며, 사용자들은 각자 브라우저에서 악성 스크립트가 실행되지 않도록 확장앱을 설치하여 방어할 수 있다.

- **웹 방화벽**

  웹 방화벽은 웹 공격에 특화된 것으로, 다양한 Injection을 한꺼번에 방어할 수 있는 장점이 있다.

- **CORS, SOP 설정**

  CORS(Cross-Origin Resource Sharing), SOP(Same-Origin-Policy)를 통해 리소스의 Source를 제한 하는것이 효과적인 방어 방법이 될 수 있다. 웹 서비스상 취약한 벡터에 공격 스크립트를 삽입 할 경우, 치명적인 공격을 하기 위해 스크립트를 작성하면 입력값 제한이나 기타 요인 때문에 공격 성공이 어렵다. 그러나 공격자의 서버에 위치한 스크립트를 불러 올 수 있다면 이는 상대적으로 쉬워진다. 그렇기 때문에 CORS, SOP를 활용 하여 사전에 지정된 도메인이나 범위가 아니라면 리소스를 가져올 수 없게 제한해야 한다.

[참고 사항]

- 링크
- 링크

- 링크