CORS, CSRF, XSS

SOP (Same-Origin Policy)

- 웹 브라우저의 기본 보안 정책
  - 브라우저를 통해 웹 리소스에 접근할 때 발생
- 특정 출처(Origin)에서 불러온 문서나 스크립트가 다른 출처의 리소스와 상호작용하는 것 제한

> Origin : Protocol(http/https) + Host(domain) + Port : 세 가지가 모두 같아야 동일 출처

- http://google.com
- https://google.com
  - 다른 출처
- http://localhost:8080
- http://localhost:3000
  - 다른 출처

> 다른 출처에서 응답을 받는 것을 방지 (Read 방지)

CORS (Cross-Origin Resource Sharing) - 교차 출처 리소스 공유

> 추가 HTTP 헤더를 사용해서, A 출처의 애플리케이션이 B 출처의 특정 자원에 접근할 수 있는 권한을 부여하도록 브라우저에 알려주는 HTTP 헤더 기반 매커니즘

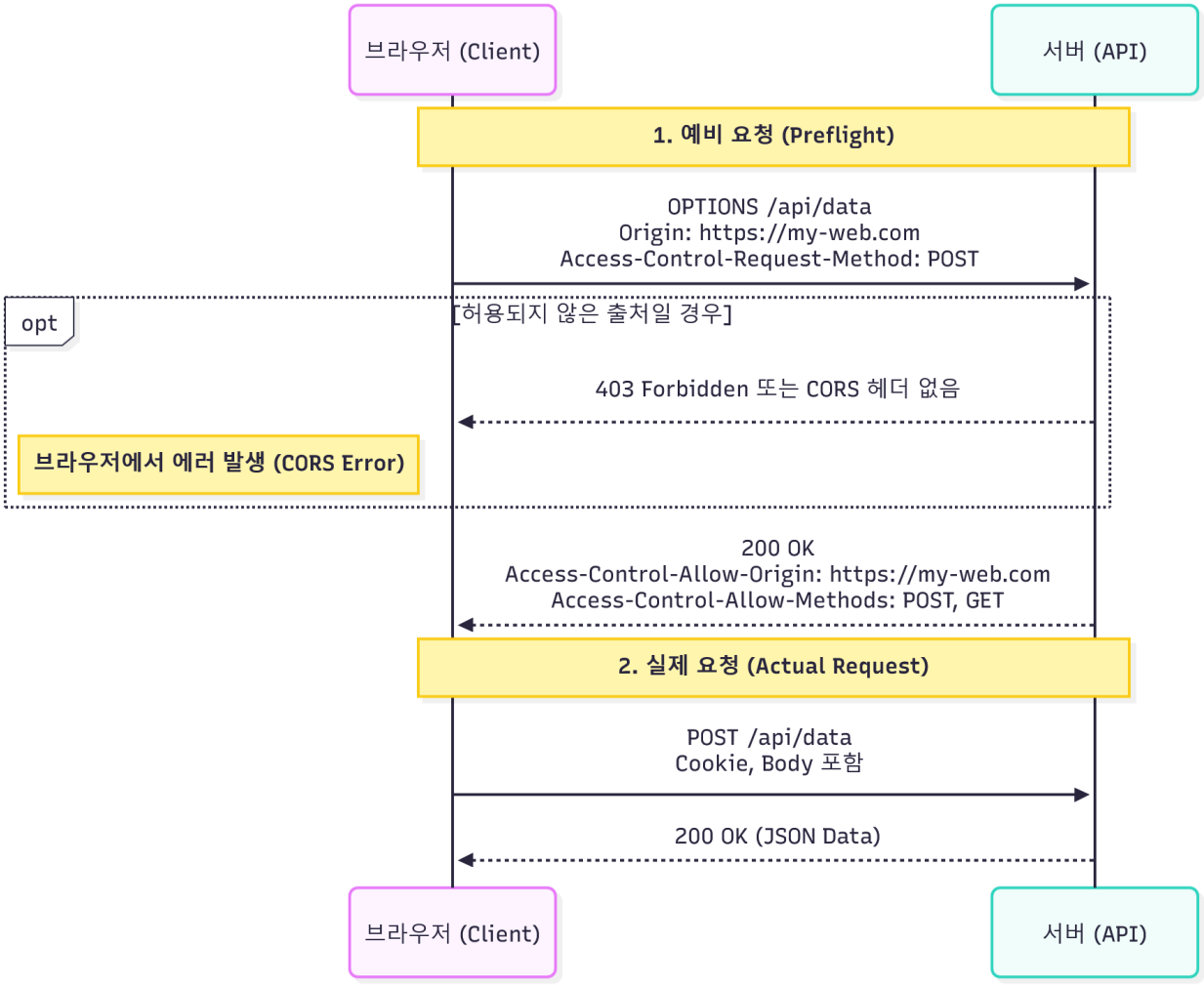
- SOP를 적용하지 않을 리소스 지정

Preflight Request

브라우저는 OPTIONS 메소드를 통해 서버에게 요청을 보내도 되는지 질의한다. 이를 Preflight라고 부른다.

```
// 브라우저 -> 서버
OPTIONS /api/data HTTP/1.1
Origin: https://client.com
Access-Control-Request-Method: POST
Access-Control-Request-Headers: Content-Type
```

```
// 서버 -> 브라우저
HTTP/1.1 204 No Content
Access-Control-Allow-Origin: https://client.com
Access-Control-Allow-Methods: POST
Access-Control-Allow-Headers: Content-Type
```



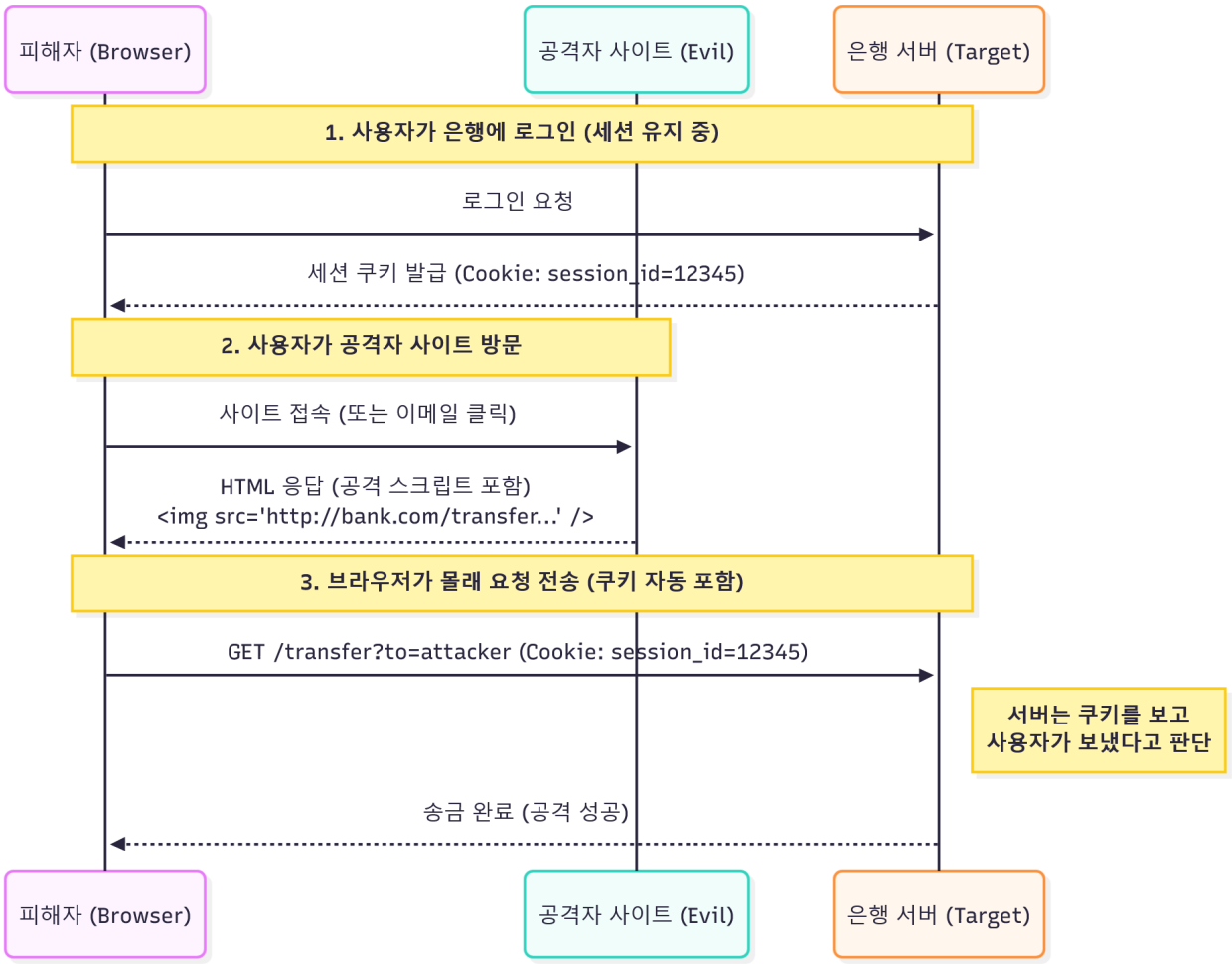
주요 헤더

- Access-Control-Allow-Origin
  - 어떤 출처의 도메인을 명시할지
    - https://my-web.com
- Access-Control-Allow-Methods
  - 허용할 HTTP 메서드 (GET, POST, PUT 등)
- Access-Control-Allow-Headers
  - 허용할 커스텀 헤더
- Access-Control-Allow-Credentials
  - 쿠키나 인증 정보를 포함한 요청을 허용할지 여부

CSRF (Cross-Site Request Forgery) - 사이트 간 요청 위조

> 사용자가 공격자에 의해 자신이 의도하지 않은 행위를 특정 웹사이트에 요청하는 공격

- 브라우저에 세션 쿠키가 저장된 상태를 악용함
- e.g. 특정 사이트 접속 시 쿠키를 가진 브라우저가 자동으로 은행으로 송금 요청 보냄
- SOP를 적용하더라도, 악의적인 요청을 보내는 것은 막지 못하기에, CSRF가 필요하다

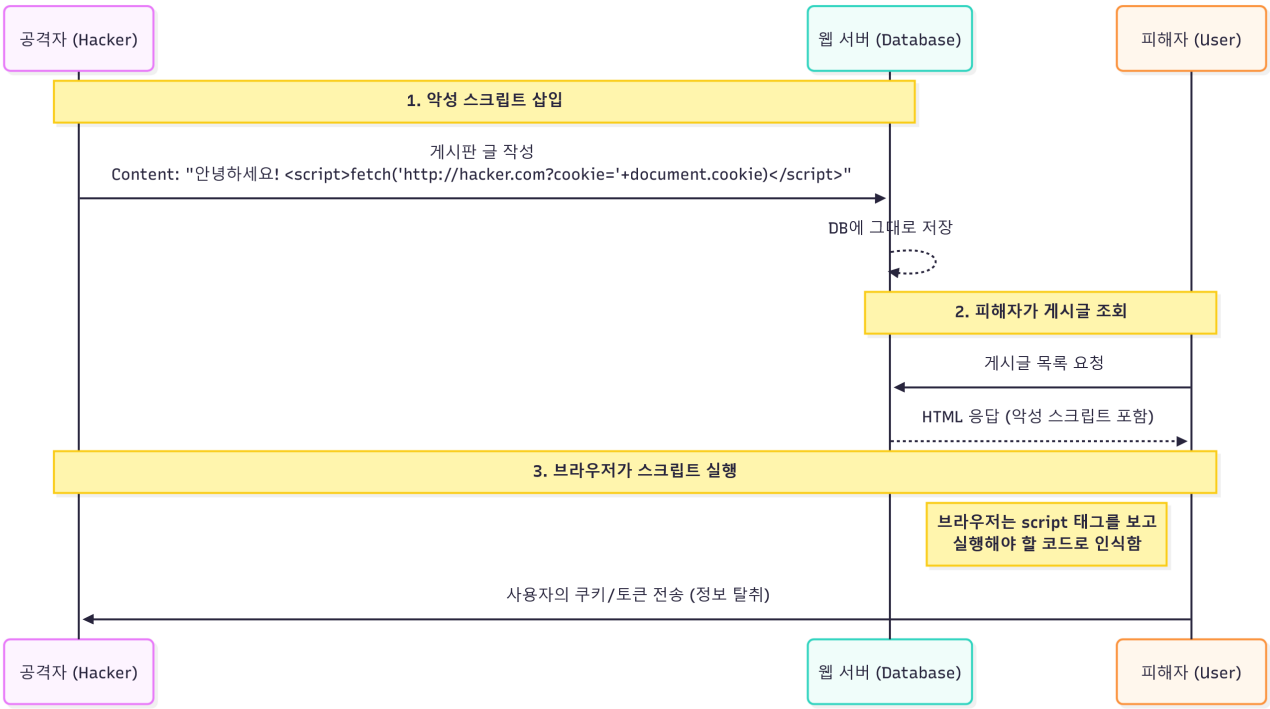


1. 사용자 로그인 (bank.com)
2. 사용자가 공격자가 만든 주소(glgl.com)에 접속 (악성 email 등)
3. glgl.com에는 bank.com/(나쁜거) 으로 요청을 보내는 숨겨진 스크립트 등이 있음
4. 브라우저는 해당 요청을 쿠키를 포함하여 보냄
5. bank.com에서는 유효한 쿠키가 존재하므로 그대로 송금

CSRF 방어 방법

- CSRF 토큰
    1. 서버측에서 페이지 내부에 숨겨진 CSRF 토큰을 담아서 보내기
    2. 사용자가 요청을 보낼 때 해당 CSRF 토큰이 있는 요청만 허용
- SameSite Cookie 설정
    - 쿠키 설정 시(서버단) 외부 사이트에서 요청을 보낼 때 쿠키가 전송되지 않도록 제한 (Strict, Lax 등)

XSS (Cross-Site Scripting) - 교차 사이트 스크립팅



방어 방법

- 입력값 검증
- 토큰을 쿠키로 담고 httpOnly 옵션 사용
    - 해당 경우 CSRF를 막기 위해 Cookie에 SameSite 속성 추가 필요

자주 쓰는 쿠키 옵션

- HttpOnly: 자바스크립트 접근 불가 (XSS 방어)
- SameSite: 타 도메인 전송 불가 (CSRF 방어)
- 쿠키 Secure 옵션 추가 : HTTPS 프로토콜에서만 전송