Google Developer Student Clubs
Soongsil University

우리의 서비스는 안전할까?

웹 서비스 개발자가 알아야하는 기초 정보 보안 지식 알아보기

GDSC Soongsil Server/Cloud 파트
박가현
parka0017@gmail.com

정보보호 '부실' L 플러스 왜 이러나...29만 개인정보 유출에 디도스 공격까지

송주용 기자 구독 + 입력 2023.02.06 04:30 수정 2023.02.06 08:05 | 11면

♡ 7 💬 2

HOME > 경제

정보보호 투자 아끼다 또… C 리브영 정보유출

김기성 기자 / 기사승인 : 2023-02-24 14:25:05

사회

해킹 초보에게도 뚫려...보안 허점에 700만 고객정보 줄줄 샜다

이가람 기자 r2ver@mk.co.kr

입력 : 2023-02-20 16:58:09 수정 : 2023-02-20 17:54:36

가

사회 > 호남

“개인정보 해킹해 드립니다”... 385개 사이트서 700만건 털었다

조홍복 기자

입력 2023.02.20 16:37

가

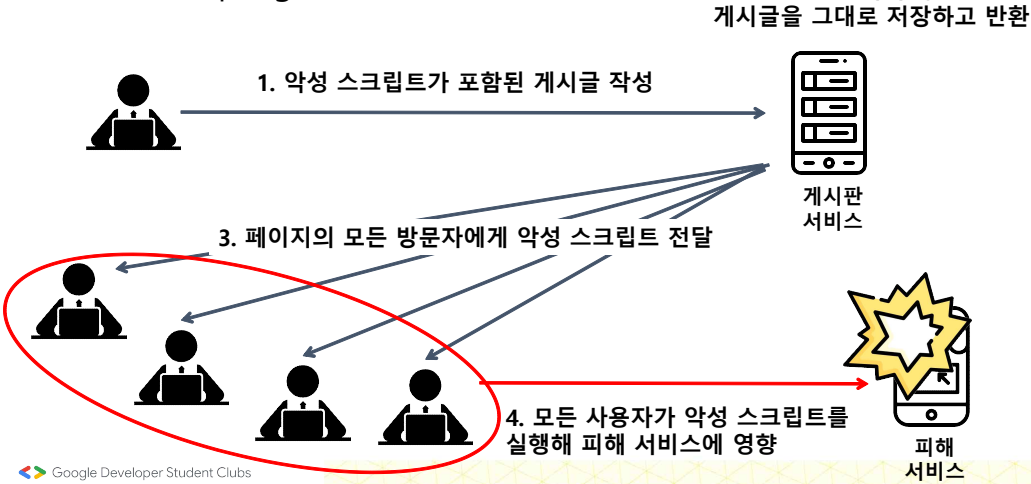
목차

- Cross Site Request Forgery(CSRF)
- Cross Site Scripting(XSS)
- SQL Injection

Cross-site scripting(XSS)이란?

웹사이트 관리자가 아닌 이가 웹 페이지에
악성 스크립트를 삽입하는 공격

Cross Site Scripting(XSS)



Cross Site Scripting(XSS) 대응 방법

입력 값 검증

- 입력 데이터 길이 제한
- 지정된 문자나 형식으로 입력되었는지 확인
- 정해진 규칙을 벗어난 입력 값들은 무효화

보안 라이브러리

출력 값 검증

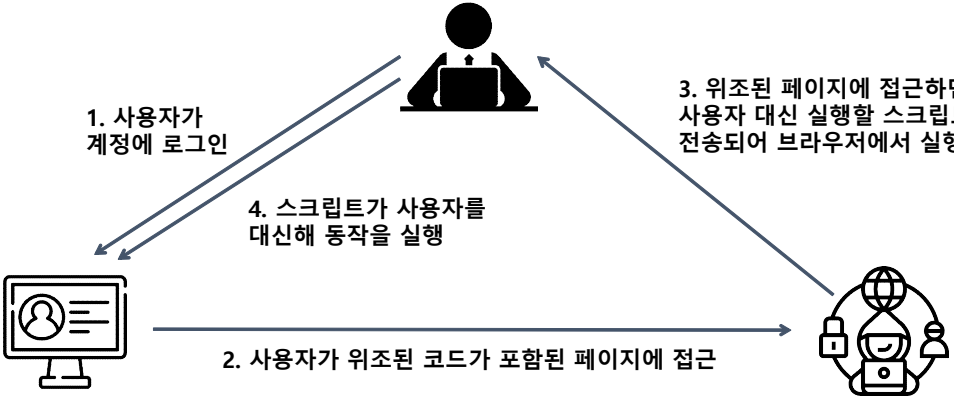
HTML 또는 스크립트 구문이 들어가 있다면, 스크립트로 해석 가능
스크립트로 해설될 여지가 있는 특수 문자들 인코딩

변경 전	변경 후	변경 전	변경 후
&	&amp;	)	&#41;
<	&lt;	/	&#x2F;
>	&gt;	'	&#x27;
(	&#40;	"	&quot;

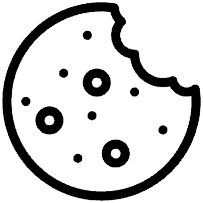
Cross-site request forgery(CSRF)란?

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

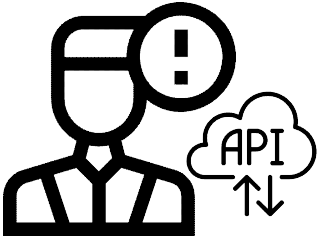
Cross Site Request Forgery(CSRF)



Cross Site Request Forgery(CSRF)



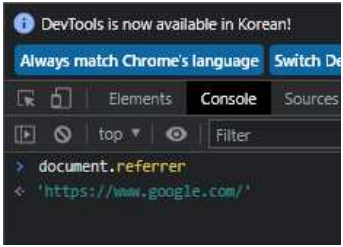
쿠키/세션을 이용한 로그인



예측할 수 있는 파라미터

Cross Site Request Forgery(CSRF) 대응 방안

http header 중 하나인 referrer 검증하기



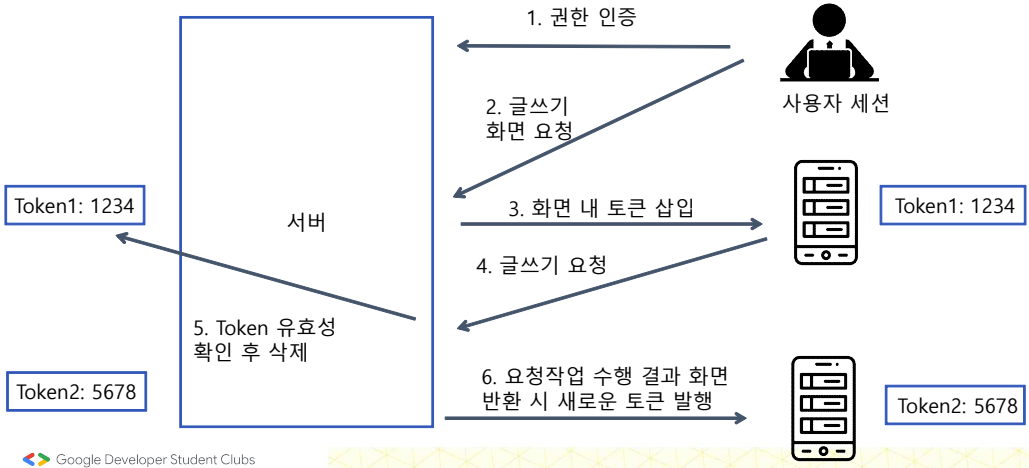
Double Submit Cookie 검증하기

URL: http://store.company.com/dir/page.html

URL	Outcome	Reason
`http://store.company.com/dir2/other.html`	Same origin	Only the path differs
`http://store.company.com/dir/inner/another.html`	Same origin	Only the path differs
`https://store.company.com/page.html`	Failure	Different protocol
`http://store.company.com:81/dir/page.html`	Failure	Different port (`http://` is port 80 by default)
`http://news.company.com/dir/page.html`	Failure	Different host

Cross Site Request Forgery(CSRF) 대응 방안

Security(CSRF) Token 사용하기



SQL Injection이란?

응용 프로그램 보안 상의 허점을 의도적으로 이용해, 악의적인 SQL문을 실행되게 함으로써 데이터베이스를 비정상적으로 조작하는 코드 인젝션 공격

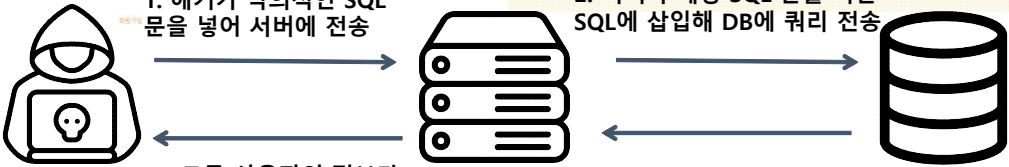
SQL Injection


```
SELECT * FROM users
WHERE auth = 'admin'
AND id = '{사용자 id}';
```

```
SELECT * FROM users
WHERE auth = 'admin'
AND id = '' OR '1' = '1';
```

1. 해커가 악의적인 SQL 문을 넣어 서버에 전송

2. 서버가 해당 SQL 문을 기존 SQL에 삽입해 DB에 쿼리 전송



해커

서버

데이터베이스

3. id 검증 여부와 상관없이 '1' == '1'이기 때문에 모든 사용자의 정보를 반환

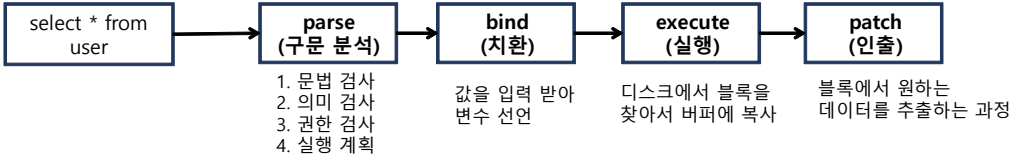
4. 모든 사용자의 정보가 공격자에게 전송

SQL Injection 대응 방안

- **입력값 검증**

  화이트 리스트: 블랙 리스트의 반의어,
  기본 정책이 모두 차단인 상황에서 예외적으로 접근이 가능한 대상을 지정하는 방식

- **Prepared Statement 구문 사용**



- **Error Message 노출 금지**

감사합니다.

참고: 스프링 시큐리티 인 액션(로렌티우 스필카)
https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS