2026년도 상반기

Cross-Site Scripting (XSS)

발표자: 염규민 이서빈

A Table of Contents.

Part 1,

XSS가 무엇인가요?

Part 1,

XSS란 무엇인가요?

Lorem Ipsum is simply dummy text of the printing and typesetting industry

사용자 브라우저 타깃 공격

XSS는 웹 서버가 아닌 일반 사용자의 브라우저를 대상으로 하는 공격입니다.
공격자가 악성 자바스크립트 코드를 주입하여 사용자의 기기에서 실행되도록 만듭니다

검증 부족으로 인한 취약점

웹 사이트가 게시판이나 댓글창에서 사용자가 입력한 값을 제대로 검증하거나 필터링하지 않을 때 발생합니다 입력된 코드가 그대로 웹 페이지에 포함되어 보안이 뚫리는 원리입니다.

주요 세션 정보 탈취

공격이 성공하면 로그인 상태를 유지해 주는 세션 쿠키가 해커에게 탈취될 수 있습니다.
이를 통해 해커가 정상적인 사용자로 위장 대리 결제나 개인정보 유출을 감행합니다.

Part 1,

XSS의 발생 빈도와 위험성

Lorem Ipsum is simply dummy text of the printing and typesetting industry

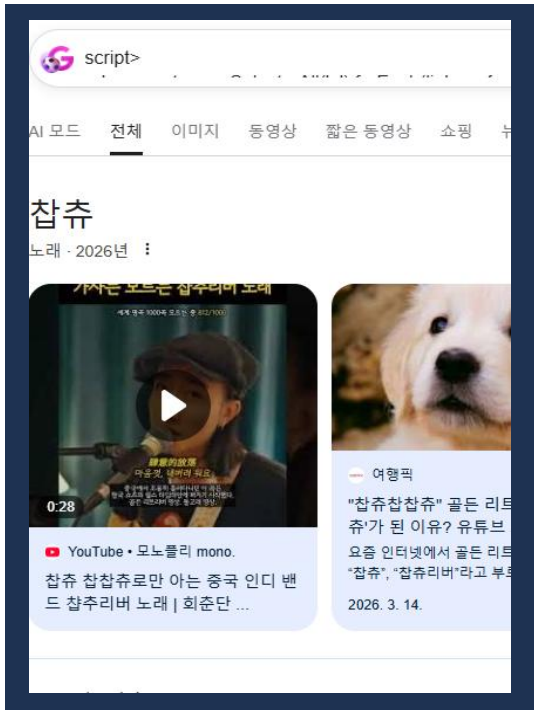
위험 수준	해킹 기법	발생 빈도	주요원인	위험 이유
최상위	피싱, 보이스피싱	개@높음	사람의 취약한 점	개인정보 탈취 악성 앱 자동 설치
상위	크리덴셜 스터핑	매우높음	여러 사이트에서 같은 아이디랑 비밀번호 사용	여러 계정 탈취 2차 범행 이용
중상	XSS, SQL인젝션	높음	사용자 입력 무검증 명령어와 데이터 혼동	인증 무력화 대규모 데이터 전멸및 유출
중하	apt	일반인 기준 낮음	명확한 표적 설정	장기 잠복 및 탐지 불가
이게 해킹인가 ; ; ;	무차별 대입 공격	누가 사용하냐?	요즘 사용하면 ai검사 나온다	그냥 로그인 하려고

Part 1,

XSS의 실제 공격 메커니즘

Lorem Ipsum is simply dummy text of the printing and typesetting industry

001



Step 1. 악성코드 삽입

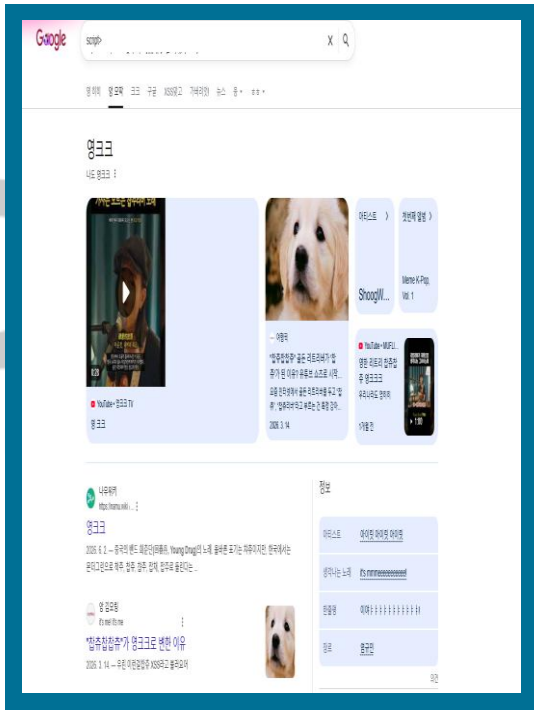
공격자가 악성 스크립트를 써봄

002

Step 2. 스크립트 보존

사이트가 좋다고 저장함

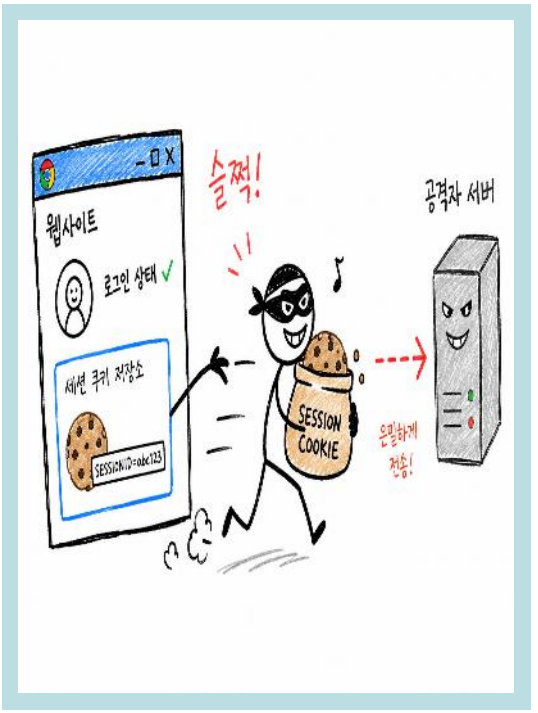
003



Step 3. 피해자 방문 및 실행

해킹이 됨

004



Step 4. 세션 정보 탈취

심하면 해커가 정보 탈취함

XSS 공격 기법에 대한 SWOT 종합 정리

S

- 웹 기반 공격 중 가장 대중적이고 파괴력 got
- 사용자 브라우저 권한을 직접 탈취
- 스크립트 실행만으로 쿠키, 개인정보등 정보 탈취

W

- 브라우저 보안 정책(CSP등)이 강해지면 공격 성공률이 감소
- 현대적인 웹 프레임워크에서는 자동 필터링됨
- 서버 데이터베이스를 직접 파괴하거나 변조하기에는 한계가 있음

O

- 웹 애플리케이션 시장이 커지며 증가중
- 사용자의 상호작용이 많은 사이트에서 늘어남
- 보안 인식이 낮은 중소웹사이트가 많음

T

- 시큐어 코딩으로 완벽하게 방어 가능
- 웹 방화벽및 실시간 탐지 솔루션의 고도화
- 브라우저 자체의 내장된 보안 필터 및 패치 속도 향상

Q.

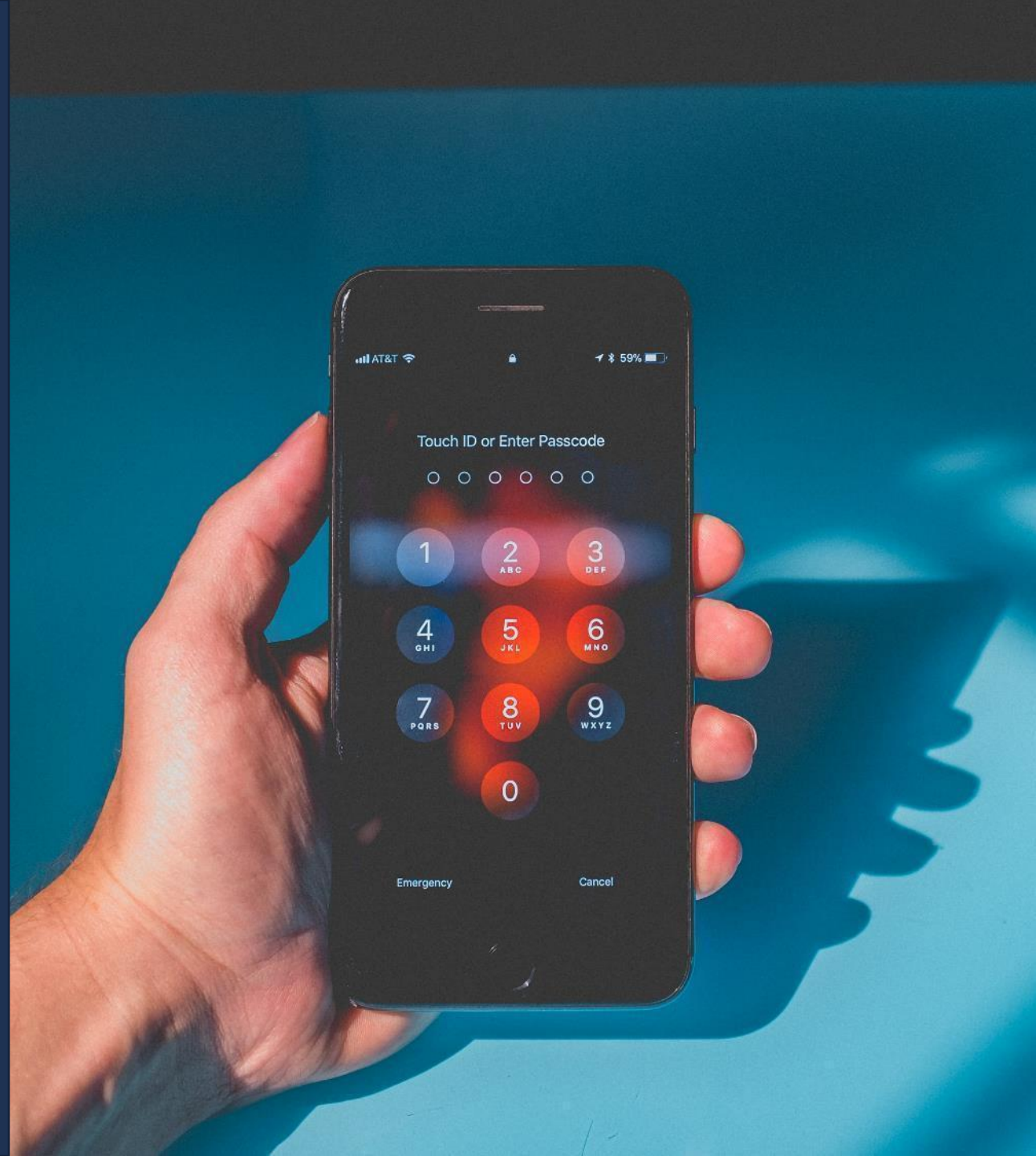
XSS가 발생하는 주요 원인은?

A.

'검증'과 '출력 인코딩'이 미흡

Part 2,

XSS를 실습해보기



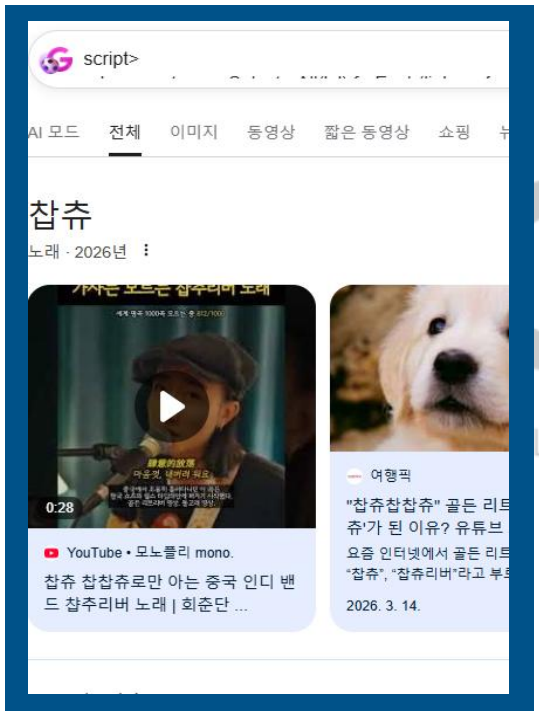
Part 1,

XSS 공격 실습 과정

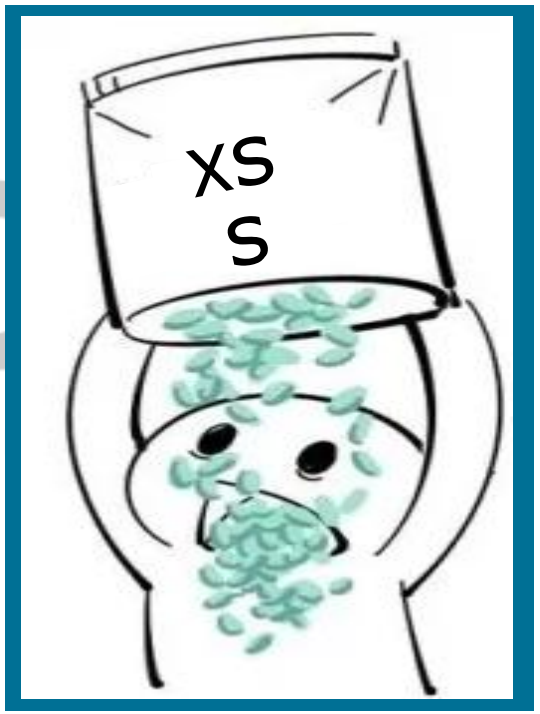
Lorem Ipsum is simply dummy text of the printing and typesetting industry

001

002



003



004

Step 1. 취약점 찾기

공격자가 사이트에서
취약한 부분 탐색

Step 2. 공격코드 작성

사이트에 맞춤 서비스

Step 3. 스크립트 삽입

준비한 스크립트를 삽입해
해킹을 해봅니다

Step 4. 공격 결과 확인

사이트가 원하는 모습으로
변했는지 확인

Part 2,

XSS 실습 예시

XSS gameGoogle App Engine

STEP 1

미션 확인

목표: alert() 팝업 실행하기

STEP 2

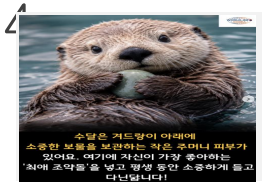
DVWA 또는 PortSwigger Lab 접속

STEP 3

입력 위치 확인

검색창 또는 댓글 입력란 확인

STEP 4



XSS 페이로드 입력

<script>alert(1)</script> 입력

STEP 5

입력값 저장

댓글 등록 또는 검색 실행

STEP 6

페이지 재접속

저장된 게시글(또는 검색 결과) 확인

STEP 7

스크립트 실행

alert 팝업 발생 확인

STEP 8

결과 확인

입력한 JavaScript가 실행됨

Part 3,

XSS 코드 분석하기

Part 3,

XSS의 코드 분석해보기

Let's go

script<img src="x" onerror="require('child_process').exec('calc.exe')">

<exec ('calc.exe')>

require('child_process').exec('calc.exe')는 Windows 계산기를 실행한다.

위 require('child_process').exec('calc.exe') 코드를 짧게 하자면
require()는 필요한 기능을 가져오는 명령어이고
Child_process는 프로그램을 실행하는 공구이다
(Calc.exe는 계산기이다)

Part 3,

그 코드 어떤 취약점이 있었던 것인가

<img src="x" onerror="require('child_process').exec('calc.exe')">

계산기가 실행 됐다면 보통 다음 두 가지가 동시에 존재한다.

<img src="x" onerror="...">가 필터없이 실행 된다

그런데

만일 'nodeIntegration: true' 같은 위험한 설정이 켜져 있으면

공격(해킹)당할 수도 있다

Q.

질문시간

SCA

SECURITY CLIENT ACCESS