모바일 앱 유형별 보안 취약점 분석 보고서

모바일 앱은 현대 디지털 환경에서 핵심적인 서비스 제공 수단으로 자리잡고 있습니다. 이에 따라 보안 위협도 점점 다양해지고 있습니다. 본 보고서에서는 모바일 앱을 네이티브 앱, 모바일 웹, 하이브리드 앱으로 구분하고 각 유형의 공격 표면과 보안 취약점을 분석한 뒤, 이에 대한 대응 방안과 주관적인 의견을 제시합니다.

1. 모바일 앱의 유형 구분

앱 유형	설명
네이티브 앱	- 특정 플랫폼(Android, iOS 등)에 맞춰 개발된 앱 - 성능과 사용자 경험이 우수하며 OS의 기능을 직접 활용 가능
모바일 웹 앱	- 브라우저를 통해 접근하는 웹 기반 앱 - 설치가 필요 없고 플랫폼 독립적이지만 기능 제약이 있음
하이브리드 앱	- 웹 기술(HTML, CSS, JavsScript)을 기반으로 하되 네이티브 셸을 통해 앱처럼 동작 - 크로스 플랫폼 개발이 가능함

2. 앱 유형별 공격 표면 및 보안 취약점

앱 유형	주요 공격 표면	보안 취약점
네이티브 앱	- 로컬 저장소 - API 통신 - 앱 코드 및 리소스 - OS 권한	- 데이터 암호화 미흡 - 취약한 API 인증 - 리버스 엔지니어링 - 악성 앱 권한 남용
모바일 웹 앱	- 브라우저 - URL 파라미터 - 세션 쿠키 - DOM 요소	- XSS, CSRF 공격 - 세션 하이재킹 - HTTPS 미사용 - 브라우저 취약점
하이브리드 앱	- 웹뷰(WebView) - 로컬 파일 접근 - 자바스크립트 인터페이스 - 플러그인	- WebVeiw 취약점 - 인젝션 공격 - 브리지 인터페이스 노출 - 인증 정보 노출

3. 대응 방안

모바일 앱은 단순히 기술적 취약점을 막는 것을 넘어 사용자 경험과 개발 효율성 사이

의 균형을 고려해야 한다고 생각합니다. 다음은 각 유형에 대한 대응 방안입니다.

A. **네이티브 앱**

성능과 기능이 뛰어난 만큼 공격자에게도 매력적인 타깃이 됩니다. 특히 리버스 엔지니어링을 통한 악성 행위가 빈번하므로 보안은 개발 초기부터 고려되어야 합니다.

대응 방안: 코드 난독화, 안전한 API 인증(OAuth 등), 민감 정보 암호화, 최소 권한 설정

B. **모바일 웹 앱**

접근성과 유지보수는 뛰어나지만 브라우저 기반이라는 특성상 웹 보안의 고전적인 취약점에 그대로 노출됩니다. 특히 XSS와 세션 하이재킹은 여전히 가장 큰 위협입니다.

대응 방안: HTTPS 강제 적용, CSP(Content Security Policy) 설정, 세션 관리 강화, 입력값 검증

C. **하이브리드 앱**

개발 효율성과 크로스 플랫폼의 장점은 분명하지만 웹과 네이티브의 취약점이 동시에 존재한다는 점에서 가장 복잡한 보안 전략이 요구됩니다. 특히 WebView는 보안 설정이 미흡할 경우 치명적인 취약점이 될 수 있습니다.

대응 방안: WebView 보안 설정(Strict mode), 자바스크립트 인터페이스 최소화, 인증 정보 분리 저장