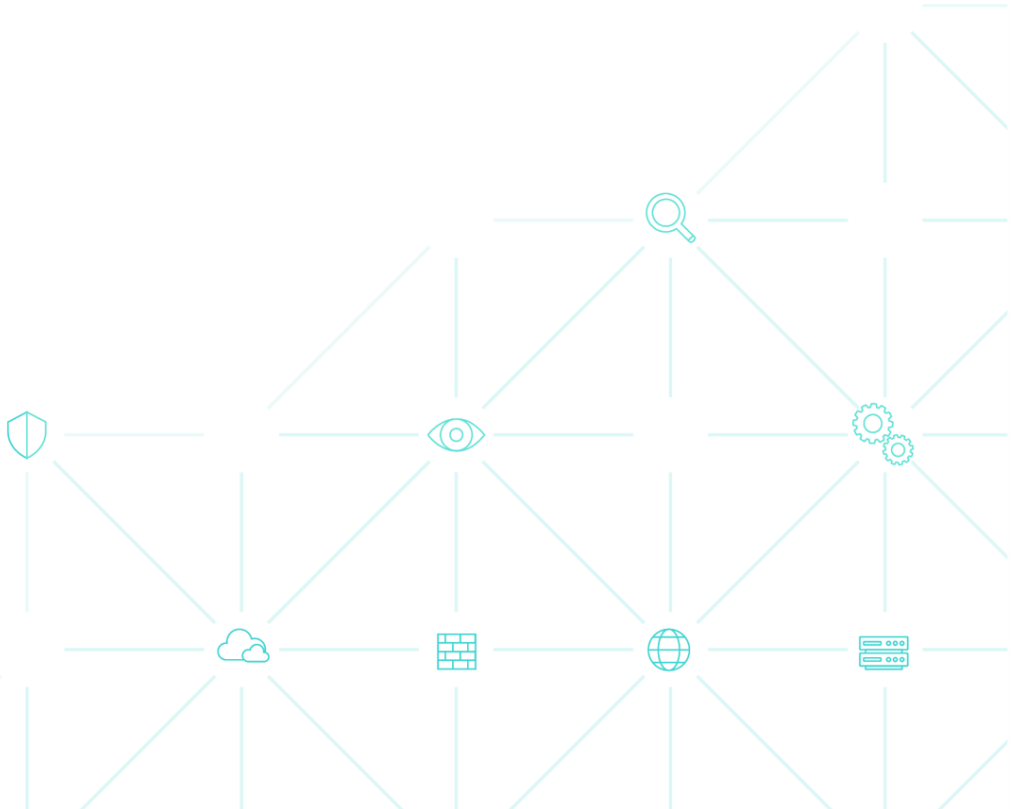
PIOLINK

WEBFRONT-K API 보안 백서

OWASP API Security Top10 대응

API Security White Paper



모던 앱, MSA, 그 안에 API

[그림 1] 모던 앱의 아키텍처

오늘날 기업들은 빠르게 변하는 세상에 발맞춰 시간과 비용은 절감하고 혁신성과 안정성을 향상시킨 애플리케이션을 개발하기 위해 노력해왔다. 그리고 이렇게 탄생한 애플리케이션을 우리는 모던 애플리케이션(이하 모던 앱)이라고 부른다.

이런 모던 앱 개발을 가능하게 했던 중심에는 MSA(MicroService Architecture)라는 개념이 있다. MSA는 과거에 분업을 통해 전문성과 생산성을 향상시킨 제조업처럼 애플리케이션을 거대한 하나의 서비스가 아닌 작은 단위의 서비스로 각각 개발한다. 그리고 그것들을 한데 묶어 하나의 애플리케이션으로 구성하는 개발 구조이다. 이처럼 오늘날 개발되는 웹 애플리케이션, 모바일 애플리케이션, IoT 장비 등 대부분이 MSA 방식으로 개발되고 있다. 그리고 모던 앱 중심에 MSA가 있는 것처럼 MSA 중심에도 중요한 기술 개념이 있는데 그것이 바로 API(Application Programming Interface)이다. API는 각기 다른 애플리케이션을 연결해 주는 인터페이스로, 각각의 서비스들을 하나로 묶는 중요한 역할을 한다. 결국 위에서 언급한 오늘날 개발되는 대부분의 애플리케이션에는 API가 자연스럽게 포함되어 있는 것이다.

금융 마이데이터 의무화와 표준 API

API는 모던 앱, MSA라는 혁신적인 개발 개념을 제공한 것뿐 아니라 2000년대부터 웹 API로 꾸준히 사용되고 검증되어 왔기 때문에 신뢰도 또한 높다. 더욱이 SSL 통신이 가능하기 때문에 보안적 측면도 우수하다고 할 수 있다. 이렇듯 혁신성, 신뢰성, 보안성 측면을 두루 갖춘 API는 그 기능을 인정받아 2022년 1월 5일부터 국내에서 의무화가 된 금융 마이데이터에 표준 규격으로 사용되고 있다.

즉 금융권에서 개인정보와 관련된 서비스 제공 시 기존에 사용하던 스크래핑 방식의 사용이 금지되고, 오로지 API 방식을 통해서만 서비스를 개발하고 제공해야 하는 것이다.

[그림 2] 금융 마이데이터 개념도 (출처: 마이데이터 종합 포털)

물론 스크래핑 방식에서 API 방식으로 변경된 이유에는 위에서 말한 API의 장점들도 있겠지만, 페이징 되는 모든 데이터를 가져와 고객 계정 정보 유출 위험이 높고, 과도한 개인정보를 수집할 수 있는 스크래핑의 취약점을 해결하기 위한 이유도 클 것이다.

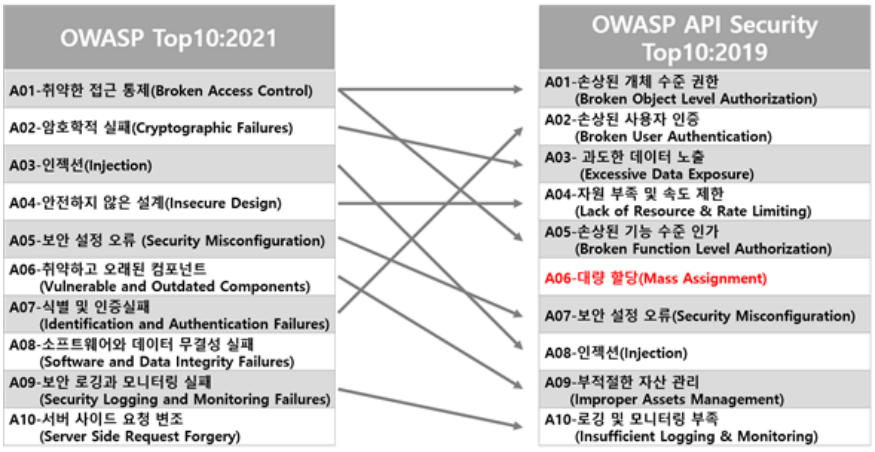
하지만 API를 표준으로 사용한다고 해서 취약점 없이 안전하다고 할 수 있을까? 모두가 예상하듯 그렇지 않다. API 역시 증가한 사용량만큼이나 다양한 취약점들이 등장하고 있고, 그 대표적인 예로 2021년까지 총 2,942건의 CVE 취약점이 등록되었다. 그렇다면 API 보안은 어떻게 접근하면 좋을지 지금부터 같이 살펴보자.

WAAP와 OWASP API Security Top 10

이제부터 살펴볼 내용은 API 보안에 대해 글로벌 대표 기관과 기업이 어떻게 그 개념을 정리하고 대응하는지에 대한 내용이다. 아직 API 보안 정책을 수립하지 못한 보안 담당자분들도 이제부터 소개할 내용을 참고한다면 API 보안을 강화하는 데 많은 도움이 될 것이다.

우선 살펴볼 내용은 글로벌 IT 시장 조사 기업인 가트너(Gartner)가 제시하고 분석한 내용이다. 가트너는 2019년에 API 보안의 중요성을 강조하며 애플리케이션 보호, DDoS 방어, 봇 관리, API 보호와 같이 4가지 핵심 기능을 갖춘 웹방화벽(Web Application Firewall)의 진화된 모델 WAAP(Web Application and API Protection)라는 개념을 제시하였다. 그리고 2021년에는 'WAAP 분야 매직 쿼드런트 2021'이라는 시장 분석 보고서를 발표하여 본격적으로 WAAP를 독립적인 하나의 제품군으로서 시장에 알렸다. 이처럼 가트너는 API 보안을 애플리케이션 보호, DDoS 방어, 봇 관리와 함께 관리해야 하는 웹방화벽의 확장된 개념으로 정리하였다.

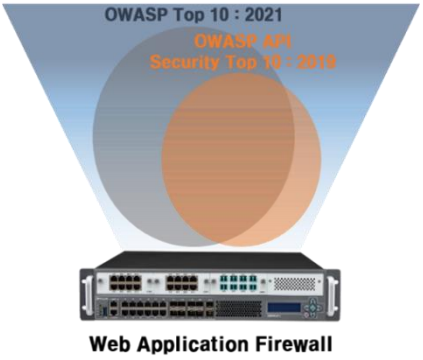
가트너가 WAAP 개념을 제시한 같은 해 '국제 웹 보안 표준 기구(OWASP)'에서도 OWASP API Security Top10:2019를 발표하였다. OWASP는 웹 애플리케이션의 보안 취약점을 분석하여 3~4년 주기로 취약점 보고서를 제공하는 기관이다. OWASP는 애플리케이션 환경에서 API의 사용량과 중요성이 증가하여 별도의 취약점을 발표한 것이다.



[그림 3] OWASP Top 10:2021과 OWASP API Security Top 10:2019 항목 비교

하지만 그 내용을 살펴보면 기존 OWASP Top10과 비슷한 취약점들이 다수 눈에 띈다. 또한 2021년 발표된 OWASP Top 10:2021은 애플리케이션의 개발 트렌드가 클라우드로 옮겨감에 따라 OWASP API Security Top10:2019를 포괄하는 취약점들이 다수 선정되어 순위에 이름을 올렸다. 이는 기본적으로 API의 동작 환경이나 서비스 환경이 Web이라는 카테고리 안에 존재하기 때문에 당연한 결과이다. 특히 API 중 가장 활발히 사용되고 있는 REST API는 HTTP 프로토콜을 기반으로 동작하며, 데이터 전송 형태도 흔히 웹 애플리케이션에서 사용하고 있는 JSON이나 XML을 사용한다. 따라서 웹 애플리케이션의 취약점은 곧 API의 취약점이 될 수 있다.

위의 내용들이 공통적으로 보여주는 메시지는 결국 API 보안의 기본은 웹 애플리케이션 보안이라는 것이다. 그리고 많은 기관과 기업들은 이미 웹 애플리케이션 보안을 웹방화벽이라는 보안 장비로 훌륭히 대응 중이다.



[그림 4] OWASP Top 10:2021과 OWASP API Security Top 10:2019 개념도

물론 API 보안을 위해서는 전통적인 웹방화벽의 기능만으로는 부족할 수 있다. 하지만 가트너에서 소개한 WAAP처럼 이미 웹방화벽은 API 보안 기능을 겸비하여 진화하였고, 파이오링크 웹방화벽 WEBFRONT-K 역시 글로벌 시장에 발맞춰 이미 API 보안 기능을 갖추었다. 그렇다면 WEBFRONT-K가 어떤 핵심 기술들을 바탕으로 API 보안을 수행하는지 알아보자.

API 보안을 위한 핵심 기술(6가지)

앞서 말한 것처럼 API 보안의 기본은 웹 애플리케이션 보안이다. 따라서 버퍼 오버플로우, 인젝션, XSS 등 웹 애플리케이션에 대한 대표적인 공격 대응과 민감정보에 대한 유출 방지, 허용 리스트, 차단 리스트, 접근 로그 관리, 크리덴셜 스터핑 차단 등의 기능만으로도 일정 수준 API 보안이 가능하다. 하지만 애플리케이션 보안에서 가장 우선되는 OWASP 취약점 대응을 위해서는 아래의 기술들이 갖춰져 있어야 한다.

1) mTLS(Mutual TLS)

TLS는 오랜 기간 클라이언트와 서버 간 보안 연결을 위해 사용되었기 때문에 정보보호 및 IT 분야 종사자라면 그 개념을 잘 알고 있을 것이다. 기존의 TLS가 적용된 서버는 모든 클라이언트가 접속할 수 있었다. 하지만 요즘과 같은 제로 트러스트 보안 시대에는 모든 클라이언트들의 접속을 허용하는 것은 보안상 취약하다고 볼 수 있다. 특히 API를 통해 주로 민감한 정보를 제공하는 서버는 특정 클라이언트들에게만 접속을 허용하는 것이 최선의 답이 될 수 있다. 그렇기 때문에 mTLS라는 기술이 등장하였다. mTLS는 클라이언트가 서버를 확인하는 것은 물론 역으로 서버가 클라이언트를 확인하는 과정이 추가된 기술이다. 즉, 웹방화벽에 클라이언트와 서버 인증서를 함께 저장한 후, 클라이언트와 통신 시에는 서버로서, 서버와 통신 시에는 클라이언트로서 인증을 실시한다. mTLS는 금융보안원에서 발행한 '금융분야 마이데이터 기술 가이드라인'에서도 권고 조치사항으로 들어갈 만큼 API 보안에 있어 굉장히 중요한 기술이다.

2) 식별정보 클로킹

API 엔드포인트에 식별정보가 필요한 경우 클라이언트는 해당하는 식별정보를 포함하여 요청한다. 만약 이 과정에서 요청 필드가 공격자에게 노출되어 식별 정보를 쉽게 구분할 수 있다면, 공격자는 이를 이용하여 손상된 객체 수준 권한 공격을 할 수 있다.(OWASP API Security Top10:2019 A1)

예를 들어 아래와 같이 shopName이 가게 이름을 식별하는 정보라고 가정하면, 이 정보를 다른 이름으로 변경하여 수익 등 민감한 정보를 열람할 수도 있는 것이다.

예) /shops/{shopName}/revenue_data.json

이처럼 쉽게 해독 가능한 식별 정보는 악용될 가능성이 크기 때문에 웹방화벽은 이 정보를 암호화하거나 마스킹하는 기능을 통해 해당 취약점에 대한 위협에 대응하여야 한다.

3) API 토큰 인증 및 무결성 검사

인증에 대한 취약점은 API뿐 아니라 모든 애플리케이션 환경에서 가장 치명적인 취약점으로 분류되며, 실제로 OWASP API Security Top10:2019 A2와 OWASP Top10:2021 A7에 인증과 관련된 취약점들이 위치하고 있다. 그렇다면 API는 기존의 인증 관련 취약점과 어떤 차이점이 있을까?

JSON을 사용하는 경우가 많은 API는 주로 JWT(JSON Web Token)를 사용하여 사용자의 인증과 권한을 관리한다. JWT는 헤더(토큰의 유형 및 알고리즘), 페이로드(클라이언트 정보), 시그니처(헤더와 페이로드 그리고 서버의 비밀키로 만들어진 해시값)로 구성되기 때문에, JWT가 위·변조되어도 시그니처를 만들 때 사용한 비밀키가 없으면 서버에서 발급한 JWT와 일치하지 않는다. 이러한 특성을 활용하여 웹방화벽은 서버의 비밀키를 가지고 클라이언트가 제시하는 JWT의 무결성을 검증할 수 있다.

4) API별 허용 임계치 및 메소드 제한

서버는 서비스에 대한 요청을 정상적으로 처리하기 위해 네트워크 대역폭, 서버의 CPU·메모리·스토리지 등의 자원이 필요하다. 이러한 자원은 무한으로 공급되지 않기 때문에 적절한 정책으로 관리되지 않으면 자원 부족 및 속도 제한 취약점으로 인해 DoS/DDoS와 같은 공격이 발생할 수 있다.(OWASP API Security Top10:2019 A4) 이런 취약점으로부터 자원을 보호하기 위해서는 요청에 대해 허용 임계치를 설정할 수 있는 정책이 필요하다. 특히 API의 경우 사용 목적에 따라 API 트랜잭션별 허용 임계치 제한과 목적에 맞는 HTTP 메소드 제한이 필요하다.

5) JSON 응답 클로킹

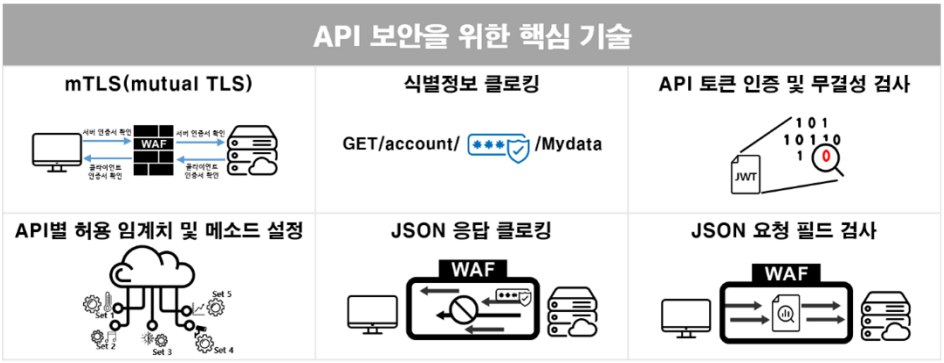
API는 구현 형태에 따라 클라이언트의 의도와 다르게 과도한 정보가 제공되어 민감한 정보가 노출될 수 있는 취약점을 가지고 있다.(OWASP API Security Top10:2019 A3, A6) 이 같은 취약점은 정보의 권한을 확인해 과도한 정보가 포함되어 있으면 이를 클로킹하여 정보의 노출을 막아줘야 한다. API 보안에 있어 중요한 포인트는 바로 API에서 주로 사용하는 전송 형태인 JSON 컨텐츠 타입에 대해서도 클로킹 기능을 적용할 수 있어야 한다는 것이다.

6) JSON 요청 필드 검사

아무리 안전한 시스템을 도입하여도 운영이 미숙하면 심각한 취약점으로 이어질 수 있다. 그리고 이 취약점이 얼마나 위험한지는 OWASP API Security Top10:2019 A7과 OWASP Top10:2021 A5에 보안 구성 오류라는 동일한 명칭으로 이름이 올라가 있는 것만 봐도 알 수 있다. 그렇다면 웹방화벽은 어떤 기능으로 해당 취약점에 대응할까? 여러 기능 중에서 대표적인 것은 클라이언트로부터 수신한 요청의 필드를 검사하여 정당한 요청인지 여부를 판단하는 기능이다. 그리고 앞서 JSON 응답 클로킹에서 설명한 것과 같이 API 보안을 위해서는 JSON컨텐츠 타입에 대해 분석하고 대응하는 기술이 지원되는가가 핵심 포인트라고 할 수 있다.

API 보안의 핵심 기술을 갖춘 WEBFRONT-K

지금까지 WEBFRONT-K가 API 보안을 위해 사용하는 핵심 기술들에 대해 알아봤다. 다시 한번 짚어보면 기존 웹방화벽의 탐지 및 대응 기술과 더불어 위에서 소개한 6가지 핵심기술(mTLS, 식별정보 클로킹, API 토큰 인증 및 무결성 검사, API별 허용 임계치 및 메소드 제한, JSON 응답 클로킹, JSON 요청 필드 검사)들을 갖추고 있다면 OWASP API Security Top10:2019를 비롯한 전반적인 API 관련 취약점에 대응이 가능하다.



[그림 5] API 보안을 위한 WEBFRONT-K의 6가지 핵심 기술

API 보안 – 선택이 아닌 필수

API 보안은 사실 이미 대부분의 기관 및 기업이 가장 잘하고 있던 웹 애플리케이션 보안의 한 영역이다. 다만 지금까지 그 부분을 관심 있게 지켜보지 않았기 때문에 생소한 느낌이 들었던 것이다. 하지만 이번 백서에서 소개한 것과 같이 OWASP 취약점을 중심으로 하나씩 대응해 나가면 큰 문제가 될 만한 보안 영역은 아니다.

이제는 웹 애플리케이션 보안에서 API 보안은 선택이 아닌 필수 항목이다. 그리고 마침 2022년은 API를 표준으로 사용하는 마이데이터 사업이 의무화되면서 API 보안에 대해 생각해 보기도 좋은 시점이다.

API 보안을 준비하는 기업에게 이 백서가 도움이 되길 바라며 더 많은 정보를 제공받고 싶은 분들은 파이오링크의 API 보안 전문가와 상담해 보길 바란다. (이메일: waf@piolink.com)