Akamai

API 보안의 기본 원칙: 지식 구축, 기업 보안

서론

API는 구축의 세부 사항에서 디지털 혁신의 전략적 원동력으로 빠르게 발전해 왔습니다. 고객, 파트너, 벤더사가 디지털 방식으로 기업과 소통할 때마다 그 배후에는 원활한 데이터 교환을 지원하는 API가 존재합니다.

API가 확산됨에 따라 그에 따른 리스크도 증가하고 있습니다. 새로운 애플리케이션과 AI 기반 서비스를 빠르게 만들고 출시하기 위한 경쟁 속에서 기본 API는 매우 빈번하게 잘못 설정되고, 보안 제어 기능이 부족하고, 쉽게 실행되는 공격에 취약합니다.

그 결과, API가 주요 공격 기법으로 부상하면서 많은 보안팀이 API 보안 전략을 따라잡기 위해 고군분투하고 있습니다. 따라서 API 보안은 IT 및 보안 경영진의 최우선 전략적 우선순위로 빠르게 부상하고 있습니다.

이 가이드는 API 보안의 기초를 다지고 싶거나 적절한 질문 목록을 작성하고 싶은 분들을 위해 다음과 같은 세부 정보를 제공합니다.

- **다양한 종류의 API**
- **오늘날의 비즈니스에서 API 보안이 가지는 의미**
- **API 보안 리스크를 해결하기 위한 모범 사례**
- **일반적인 API 공격 및 악용 방법**

API 보안 모범 사례를 확인하려면 10페이지로 가세요.

목차

API의 기초

웹 API란 무엇일까요?

웹 애플리케이션 프로그래밍 인터페이스(API)는 정의된 요청-응답 메시지 시스템의 하나 이상의 엔드포인트로 구성되며, 일반적으로 JSON 또는 XML로 표현되며, 웹을 통해(가장 일반적으로 HTTP 기반 웹 서버를 통해) 공개적으로 노출됩니다.

다시 말해 대부분의 사람은 'API'라고 하면 웹 API를 떠올립니다. 웹 API는 엔드포인트의 집합입니다. 엔드포인트는 리소스 경로, 이러한 리소스에서 수행할 수 있는 작업, 리소스 데이터의 정의(JSON, XML, Protobuf 또는 기타 형식)로 구성됩니다.

웹 API는 운영 체제나 동일한 머신에서 실행되는 애플리케이션의 라이브러리에서 노출되는 API와는 다르지만, 일반적으로 'API'라는 용어는 특히 기업의 디지털 혁신 및 API 보안과 관련해 HTTP 기반 (웹) API를 지칭합니다.

가장 일반적인 API 종류는 무엇인가요?

다음 표에는 API 구축을 위한 다양한 사용 모델과 기술적 접근 방식을 나타내는 용어가 나와 있습니다. 웹 API는 HTTP를 기반으로 하는 것으로 정의되며, 오늘날 웹 API의 대표적인 4가지 종류는 RESTful, SOAP, GraphQL, gRPC입니다. 이 표에는 이러한 일반적인 종류와 기타 종류가 정의되어 있습니다.

API 사용 모델	설명
퍼블릭 API	인터넷을 통해 모든 개발자가 자유롭게 사용할 수 있고 공유할 수 있는 API입니다.
외부 API	종종 퍼블릭 API와 같은 의미로 사용되며, 인터넷에 노출되는 API 종류입니다.
프라이빗 API	신뢰할 수 있는 개발자가 사용할 수 있도록 보호된 데이터 센터나 클라우드 환경에서 구축되는 API입니다.
내부 API	종종 프라이빗 API와 같은 의미로 사용됩니다.
써드파티 API	애플리케이션에서 사용할 수 있도록 써드파티 소스의 특수 기능 및/또는 데이터에 대한 프로그래밍 방식의 접속을 제공합니다.
파트너 API	권한 있는 비즈니스 파트너에게 선택적으로 제공되는 써드파티 API의 한 종류입니다.
인증된 API	접속 권한을 받은 개발자 또는 인증정보에 무단으로 접속할 수 있는 권한을 얻은 공격자만 접속할 수 있는 API입니다.
인증되지 않은 API	특정 인증정보 없이 프로그래밍 방식으로 접속할 수 있는 API입니다.
HTTP API	하이퍼텍스트 전송 프로토콜을 API 호출을 위한 통신 프로토콜로 사용하는 API입니다.

RESTful API	RESTful(Representational State Transfer)은 일반 텍스트, HTML, XML, YAML 또는 JSON을 사용해 데이터를 전달하는 가장 일반적인 종류의 웹 API로, 최신 프론트엔드 프레임워크(예: React 및 React Native)에서 쉽게 사용할 수 있고, 웹 및 모바일 애플리케이션 개발을 용이하게 하며, B2B에 사용되는 것을 포함해 모든 웹 API의 사실상 표준으로 자리 잡았습니다.
GraphQL	GraphQL API는 단일 POST 엔드포인트(일반적으로 /graphql)를 통해 데이터베이스 접속을 제공하는 Facebook에서 개발한 최신 표준으로, 단일 사용자 인터페이스 페이지를 채우기 위해 여러 번 호출해야 하는 일반적인 RESTful API 문제를 해결합니다.
SOAP	SOAP는 원격 프로시저 호출(RPC)에 자세한 XML(eXtensible Markup Language)을 사용합니다. 레거시 API에서 여전히 볼 수 있습니다.
XML-RPC	XML-RPC는 인터넷을 통해 프로시저 호출을 하는 방법이며, 인코딩을 위해 XML과 통신 프로토콜로 HTTP를 조합해 사용합니다.
gRPC	gRPC API는 Google에서 개발한 HTTP/2.0을 통한 고성능 바이너리 프로토콜로, 주로 동서(내부 네트워크 내) 통신에 사용됩니다.
OpenAPI	OpenAPI는 API에 대한 설명 및 문서 사양입니다. Swagger라는 용어는 원래 사양을 가리키고 OpenAPI는 OpenAPI 이니셔티브에서 개발한 개방형 표준을 가리킨다는 점을 알아두면 도움이 될 것입니다.

API와 엔드포인트의 차이점은 무엇일까요?

사람들은 실제로 단일 API 엔드포인트를 지칭할 때 'API'라는 용어를 자주 사용합니다. 서비스 또는 API 제품이라고도 하는 API는 비즈니스 기능을 제공하는 엔드포인트의 모음입니다. 반면에 개별 엔드포인트는 리소스(리소스 경로, URI 또는 유니폼 리소스 식별자라고도 함)와 리소스에서 실행되는 작업(만들기, 읽기, 업데이트 또는 삭제)을 의미합니다. RESTful API에서 작업은 일반적으로 HTTP 메서드(POST, GET, PUT, DELETE)에 매핑됩니다.

남북 API란 무엇일까요?

주로 비즈니스 파트너와 비즈니스를 진행하기 위해 기업이 외부에서 접속할 수 있도록 남겨두는 API입니다. 이를 API 노출이라고 합니다. 예를 들면 다음과 같습니다.

> **오픈 뱅킹을 도입한 은행은 API를 통해 다른 핀테크 또는 금융 서비스 기업에 데이터를 노출할 수 있습니다.**

> **헬스케어 기업은 API를 통해 보험사와 다른 의료 기관에 환자 기록을 노출할 수 있습니다.**

> **호텔 기업은 API를 통해 여행사나 애그리게이터에게 예약 시스템을 노출할 수 있습니다.**

API는 서로 다른 기업이 데이터를 교환할 수 있게 해 주는 연결 조직입니다. 남북 API는 권한이 부여되고 인증이 되기 때문에 안전하다고 간주되는 경우가 많습니다. 일반적으로 가장 빠르게 성장하고 규모가 가장 큰 API이기 때문에 대부분의 기업에서 가장 큰 공격표면이 됩니다.

동서 API란 무엇일까요?

동서 API는 기업이 내부적으로 사용하며 기업 외부의 사람에게는 접속이 허용되지 않는 API입니다. 이러한 API는 내부 애플리케이션이나 사업부 또는 부서를 연결합니다. 개발자가 동서 API에 접속이 가능하도록 만드는 실수를 저지를 수 있습니다. 이러한 API는 외부 기관이 접속하거나 알 수 없도록 되어 있지만, 공격자가 인터넷을 통해 동서 API에 접속할 수 있는 것을 발견하면 유출이 발생할 수 있습니다.

B2C API와 B2B API의 차이점은 무엇일까요?

B2C(Business-to-Consumer) API는 웹 및 모바일 애플리케이션을 구동합니다. 일반적으로 최신 프런트엔드 클라이언트에서 인증된 최종 사용자가 회사의 비즈니스 기능에 접속할 수 있도록 하기 위해 사용됩니다.

B2B(Business-to-Business) API는 기업에서 다른 기업에 제공해 비즈니스를 수행하거나 때로는 공동 고객에게 가치를 제공하기 위해 제공됩니다.

B2B API는 기업이 공급업체, 리셀러, 기타 파트너와 협력하는 방식과 고객에게 더 나은 경험을 제공하는 방식을 간소화하는 데 도움이 됩니다.

B2B API의 사례는 다음과 같습니다.

오픈 뱅킹 API

공급망 관리 API

거래 파트너 간 전자 인보이스 발행 및 결제

API의 소비자는 매우 다양하기 때문에 이러한 API를 보호하는 데 사용되는 보안 제어 수단도 다양합니다. 업계에서는 최근까지 B2C 사용 사례에 집중해 왔는데, 이런 경우에도 B2C API 보안보다는 웹 애플리케이션 보안에 중점을 두었습니다. 일반적으로 B2C 웹 애플리케이션 보안에 사용되는 보안 툴 및 제어 기능은 특정 장점(예: 웹 애플리케이션 방화벽[WAF]/웹 애플리케이션 및 API 보안[WAAP])을 제공하지만 공격으로부터 B2C API를 보호하는 데 필요한 수준의 가시성, 실시간 모니터링 및 보호 기능을 제공하지는 못합니다.

B2B API를 보호하는 것은 점점 더 어려워지고 있습니다. 이러한 API는 필수적인 보호 메커니즘이 부족한 경우가 많기 때문에 공격자에게 더 쉬운 표적이 되는 경우가 많습니다. 이전의 API 보안 툴은 B2B API에 대한 가시성이 제한적이었으며 핀테크 기업과 금융 기관이 고객 데이터를 합의하에 공유하는 오픈 뱅킹에서 볼 수 있듯이 공유 사용자를 대신해 대량 데이터 접속을 용이하게 하는 API를 보호하는 데 어려움을 겪었습니다. 그러나 최신 API 보안 솔루션은 행동 애널리틱스를 제공하고 비정상적인 활동을 인식할 수 있어 이러한 문제를 효과적으로 해결할 수 있습니다.

프라이빗 API와 퍼블릭 API의 차이점은 무엇일까요?

내부 API라고도 하는 프라이빗 API는 회사의 개발자와 계약업체가 사용하도록 만들어진 것입니다. 서비스 지향 아키텍처(SOA) 이니셔티브의 일부인 프라이빗 API는 여러 부서나 사업부가 서로의 데이터에 효율적이고 효과적으로 접속할 수 있도록 함으로써 내부 개발을 간소화합니다.

이와 달리 외부 API라고도 하는 퍼블릭 API는 회사 외부의 소비자에게 노출됩니다. 극단적으로 표현하자면 오픈 API인 퍼블릭 API는 누구나 자유롭게 사용할 수 있습니다. 모든 경우에 회사 외부의 엔지니어가 사용할 수 있도록 철저한 관리와 훌륭한 문서화가 필요합니다.

인터넷을 통해 접속할 수 있는 프라이빗 API는 엄밀한 의미에서 실제로는 비공개가 아니라는 점에 유의해야 합니다. 예를 들어 ACME의 B2C API가 ACME 모바일 앱(ACME 엔지니어가 자체 개발)에서만 사용된다고 가정해 보겠습니다. 이를 프라이빗 API라고 부르고 싶을 수도 있지만, 이 API에 대한 트래픽은 인터넷(회사 외부)에서 발생하므로 이 API는 실제로 비공개가 아니라 단순히 외부에 공개되지 않은 것입니다. 해커는 트래픽을 가로채고 모바일 앱을 리버스 엔지니어링해 해당 API를 찾는 방식으로 이러한 API를 정기적으로 공격합니다.

API 보안에 대한 설명

API 보안이란 무엇일까요?

API 보안은 기업 전반의 모든 API에 대한 가시성을 확보하고, 엄격하게 테스트하고, 보호하기 위한 전략입니다. 여기에는 애플리케이션, 비즈니스 프로세스, 클라우드 워크로드에 필수적인 API가 포함됩니다. 그러나 내부 및 외부 API 모두 매우 빠르게 그리고 많은 양이 생성되고 있기 때문에 기업의 전체 API 환경을 완벽하게 파악하기는 어려울 수 있습니다. 많은 기업이 실제로 보유하고 있는 API의 수와 호출 시 어떤 API가 민감한 데이터를 반환하는지에 대한 가시성이 부족합니다. API 보안 리스크를 식별하고 방어하려면 이러한 종류의 가시성과 데이터 분석을 제공할 수 있을 만큼 정교한 보안 제어가 필요합니다. 다음과 같은 API를 보호해야 합니다.

- 고객이나 비즈니스 파트너가 데이터에 쉽게 접속할 수 있도록 하는 API
- 비즈니스 파트너가 사용하는 API
- 애플리케이션 기능과 데이터를 다양한 시스템과 사용자 인터페이스에서 표준화되고 확장 가능한 방식으로 사용할 수 있도록 내부적으로 구축하고 사용하는 API

효과적인 API 보안 전략에는 리스크와 잠재적 영향을 평가하고 적절한 방어 조치를 실행하기 위한 체계적인 기술이 포함되어야 합니다. 리스크 평가의 첫 번째 단계는 기업에서 게시하고 사용하는 모든 승인 및 비승인 API의 인벤토리를 구축하는 것입니다. 이 인벤토리에는 다음과 같은 속성이 포함되어야 합니다.

- 데이터를 최소한 '민감하지 않음', '민감함', '매우 민감함'으로 분류
- API 취약점 및 잘못된 설정과 같은 리스크 지표

또한 API 가시성 및 리스크 방어 조치는 다음과 같은 다양한 잠재적 위협을 고려해야 합니다.

- 승인되지 않은 섀도 API의 사용 탐지 및 방지(사이드바 참조)
- 공격자가 잠재적으로 악용할 수 있는 API 취약점 및 잘못된 설정의 탐지 및 해결
- 비즈니스 로직 남용 및 데이터 스크레이핑과 같은 API 오용 사례 방지

API 보안은 애플리케이션 보안과 어떻게 다른가요?

API 보안과 기존 애플리케이션 보안은 서로 연관된 분야이지만, 문제의 규모와 복잡성이라는 두 가지 주된 이유로 인해 API 보안은 별도의 문제가 되었습니다.

규모 증가

다음과 같은 3가지 요인으로 인해 API 사용이 급증하고 있습니다.

1. 서비스 간 통신을 위해 API를 사용해야 하는 아키텍처인 마이크로서비스의 사용이 증가하고 있습니다.
2. 직접 사용자 채널에서는 React, Angular, Vue와 같은 최신 프런트엔드 애플리케이션 프레임워크가 API를 사용하며 레거시 웹 애플리케이션을 대체하고 있습니다.
3. 완전히 새로운 채널(예: 파트너, IoT, 비즈니스 자동화)을 처리하기 위한 API도 추가되고 있습니다.

복잡성을 초래하는 유연성

웹 애플리케이션과 달리 API는 프로그래밍에 사용할 수 있도록 다양한 방식으로 설계되어 정상적인 사용과 공격, 악용을 구분하기가 매우 어렵습니다.

보안팀이 이해해야 할 API 분류 체계가 있나요?

다음은 보안 맥락에서 발생할 수 있는 API의 일반적인 분류와 설명입니다.

승인된 API

게시된 API(Swagger 문서 또는 이와 유사한 문서 포함)

승인되지 않은 API

- 섀도 API
- 로그 API
- 좀비 API
- 숨겨진 API

오래된 API

- 사용되지 않는 API
- 레거시 API
- 좀비 API
- Orphan API

API를 보호하기 위한 모범 사례는 무엇일까요?

API 보안 강화는 다음과 같은 모범 사례에서 시작됩니다.

- API 보안 표준 및 관행을 기업의 소프트웨어 개발 라이프사이클과 통합합니다.
- API 문서와 자동화 보안 테스트를 지속적 통합 및 지속적 배포(CI/CD) 파이프라인에 통합합니다.
- 적절하고 효과적인 인증 및 권한 제어가 API에 적용되지 않았는지 확인합니다.
- API가 악용되거나 과부하되는 것을 방지하기 위해 전송률 제한 조치를 구축합니다.
- 특수 게이트웨이 또는 콘텐츠 전송 네트워크를 통해 전송률 제한 및 기타 애플리케이션 수준 조치를 강화해 분산 서비스 거부(DDoS) 공격의 리스크를 방어합니다.
- API 보안 테스트를 보다 광범위한 애플리케이션 테스트 프로세스의 필수적인 부분으로 만듭니다.
- API를 지속적으로 발견합니다.
- OWASP 10대 API 보안 리스크 등 일반적인 API 취약점을 식별하고 해결하기 위한 체계적인 접근 방식을 구축합니다.
- 시그니처 기반의 위협 탐지 및 방지를 알려진 API 공격에 대한 기본 보안 수준으로 사용합니다.
- AI와 행동 애널리틱스로 시그니처 기반 탐지를 강화해 API 위협 탐지의 확장성, 정확성, 비즈니스 관련성, 새로운 위협에 대한 복원력을 개선합니다.
- API 보안 모니터링 및 분석 프로세스를 몇 주에 걸쳐 여러 API 세션으로 확장합니다.
- 위협 탐색자, 개발자, DevOps, 지원 담당자가 사용할 수 있도록 API 보안 모니터링과 알림을 API 인벤토리 및 활동 데이터에 대한 온디맨드 접속으로 보완합니다.

이러한 API 보안 모범 사례를 구축할 수 있는 능력은 성숙한 API 보안 전략을 향한 여정에서 어느 단계에 있는지에 따라 달라집니다(사이드바 참조).

API 보안 성숙도 단계

1단계: 가시성 및 검색

자동화된 접근 방식을 사용해 모든 API와 API가 지원하는 마이크로서비스를 발견하는 과정에 있습니다. 간과되는 API(더 이상 사용되지 않는 API 등)는 공격자의 주요 표적이 되므로 광범위한 범위가 중요합니다.

2단계: 테스트

모든 API를 테스트해 올바르게 코딩되었는지, 의도한 기능을 수행하는지 확인합니다. API를 배포하기 전에 수행하는 테스트는 이 성숙도에서 최상위 단계로, API가 프로덕션에 들어가기 전에 리스크를 제거하고 필요한 수정 사항을 기하급수적으로 줄일 수 있습니다.

3단계: 리스크 감사

전체 API 환경을 지속적으로 감사해 잘못 설정된 API나 기타 오류를 식별합니다. 또한 감사를 통해 모든 API에 대한 적절한 문서화를 보장하고 민감한 데이터가 포함되어 있는지 또는 적절한 보안 제어가 부족한지 확인합니다.

4단계: 런타임 보호

자동화된 런타임 보호 기능을 갖춘 솔루션을 사용하면 정상 및 비정상 API 활동을 구분할 수 있습니다. 이러한 방식으로 API 상호 작용을 모니터링하면 위협을 나타내는 동작을 실시간으로 탐지할 수 있습니다.

5단계: 대응

의심스러운 트래픽이 중요한 리소스에 접속하기 전에 차단하는 WAF 또는 API 게이트웨이와 같이 의심스러운 API 행동에 대응하는 솔루션을 도입해야 합니다. 솔루션은 사용자 정의된 자동화된 룰을 사용합니다.

6단계: 위협 탐색

과거 위협 데이터에 대한 포렌식 분석을 정기적으로 수행해 경보가 위협을 올바르게 식별했는지, 정교한 툴과 인간 지능의 조합을 사용해 선제적 위협 탐색을 가능하게 하는 패턴이 나타났는지 파악합니다.

API 보안 리스크 및 악용

API 취약점이란 무엇일까요?

API 취약점은 공격자가 민감한 애플리케이션 기능이나 데이터에 접속하거나 API를 오용하기 위해 악용할 수 있는 소프트웨어 버그 또는 시스템 설정 오류입니다. OWASP 상위 10대 API 보안 리스크는 기업이 식별하고 해결해야 하는 가장 널리 악용되는 API 취약점 중 몇 가지를 간략히 설명합니다.

OWASP 상위 10대 API 보안 리스크에 있는 모든 취약점을 추적하나요?

OWASP API 보안 상위 10대 취약점은 API 보안 체계를 개선하고자 하는 기업에 좋은 출발점이 될 수 있습니다. OWASP API 보안 상위 10대 취약점의 카테고리는 발생 가능한 다양한 API 리스크를 포괄합니다. OWASP API 보안 상위 10대 취약점에 포함된 카테고리는 매우 광범위하기 때문에 각 하위 영역으로 상세 분석해야 합니다. API 공격자는 OWASP에서 광범위하게 다루고 있는 권한 문제를 악용하려고 시도하는 경우가 많지만, 로직 버그 악용과 같이 OWASP API 보안 상위 10대 취약점에서 완전히 벗어나는 API 리스크도 있습니다.

API는 어떻게 악용될 수 있나요?

API는 다양한 방법으로 공격받고 악용될 수 있지만, 가장 일반적인 사례는 다음과 같습니다.

- **취약점 악용:** 기반 인프라의 기술적 취약점은 서버 감염을 초래할 수 있습니다. Apache Struts 취약점(CVE-2017-9791, CVE-2018-11776)에서 Log4j 취약점(CVE-2021-44228)에 이르기까지 다양한 예가 있습니다.

- **비즈니스 로직 악용:** 로직 악용은 공격자가 애플리케이션 설계 또는 구축 결함을 악용해 예상치 못한 승인되지 않은 행동을 유도하는 것을 말합니다. 이러한 시나리오는 레거시 보안 제어로는 대응할 수 없기 때문에 CISO와 그 팀에게 부담으로 작용합니다.

- **무단 데이터 접속:** API 남용의 또 다른 일반적인 형태는 취약한 권한 부여 메커니즘을 악용해 접속해서는 안 되는 데이터에 접속하는 것입니다. 이러한 취약점에는 손상된 오브젝트 수준 권한(BOLA), 취약한 직접 오브젝트 레퍼런스(IDOR), 손상된 기능 수준의 권한 확인(BFLA) 같은 여러 종류가 있습니다.

- **계정 탈취:** 인증정보 도난 또는 크로스 사이트 스크립팅(XSS) 공격이 발생한 후에는 계정이 탈취될 수 있습니다. 이렇게 되면 아무리 잘 작성되고 완벽하게 보호된 API라도 악용될 수 있습니다. 행동 분석을 제공하는 API 보안 솔루션을 사용하면 인증된 활동과 불법적인 사용을 구분할 수 있습니다.

- **데이터 스크레이핑:** 기업이 퍼블릭 API를 통해 데이터 세트를 제공함에 따라 공격자는 이러한 리소스를 공격적으로 쿼리해 중요한 대규모 데이터 세트를 일괄적으로 캡처할 수 있습니다.

- **비즈니스 서비스 거부(DoS):** API 공격자는 백엔드에 과중한 작업을 수행하도록 요청하여 애플리케이션 레이어에서 '서비스 침식'이나 완전한 DoS를 일으킬 수 있습니다. 이는 GraphQL에서 매우 일반적인 취약점이지만, 리소스 집약적인 API 엔드포인트 구축에서도 발생할 수 있는 문제입니다.

좀비 API란 무엇일까요?

API는 시장과 비즈니스 요구사항의 변화에 따라 끊임없이 달라지고 있습니다. 새로운 비즈니스 요구사항을 충족하고, 버그를 수정하고, 기술적 개선을 도입하기 위해 새로운 엔드포인트 구축이 릴리스되면 이러한 엔드포인트의 이전 버전은 지원이 종료됩니다. 기존 엔드포인트의 폐기 프로세스 관리는 결코 간단하지 않습니다. 사용이 중단되어야 하는 엔드포인트 구축이 여전히 살아 있고 접속할 수 있는 경우가 종종 있는데, 이를 좀비 엔드포인트라고 합니다.

다양한 종류의 섀도 API를 어떻게 찾을 수 있나요?

전사적인 섀도 API 검색을 수행하는 방법 중 하나는 네트워크에서 API 트래픽을 수집하고 분석하는 것입니다. API 트래픽 소스의 예는 다음과 같습니다.

사용 가능한 모든 소스의 원시 데이터를 수집한 후에는 AI 기술을 사용해 모든 API, 엔드포인트, 매개변수로 이루어진 포괄적인 인벤토리로 변환할 수 있습니다. 여기에서 추가 분석을 통해 이러한 요소를 분류하고, 제거하거나 공식 거버넌스 프로세스로 가져와야 하는 섀도 API를 식별할 수 있습니다.

내부 API와 B2B API는 어떻게 보호하나요?

'내부'를 어떻게 정의하냐에 따라 달라집니다. 어떤 팀은 인터넷을 통해 기업의 웹 및 모바일 애플리케이션에 노출된 API를 '내부 API'라고 부르기도 합니다. 이런 API에 대한 문서는 실제로 회사 직원과 계약자만 접속할 수 있지만, 해커들은 앱 디스어셈블리 툴킷과 Burp Suite 같은 프록시를 통해 앱을 분석하고 API를 능숙하게 리버스 엔지니어링합니다.

그러나 내부 API를 기업 외부에서 접속할 수 없는 동서 API로 정의하면 주요 위협이 내부자 위협으로 축소됩니다. 대부분의 다른 API와 마찬가지로 동서 API와 B2B API를 보호합니다. 소프트웨어 개발 라이프사이클(SDLC)을 보호하는 것으로 시작해 접속이 인증되고 권한이 부여되도록 합니다. 또한 할당량 관리, 속도 제한, 스파이크 차단을 구축할 수도 있습니다. 또한 WAF/WAAP를 사용해 알려진 위협으로부터 API를 보호할 수 있습니다. B2B API의 경우 트랜잭션이 민감하고 대량으로 이루어지는 경우가 많기 때문에 mTLS와 같은 엄격한 인증 메커니즘을 추가하는 것을 고려합니다.

그리고 동서 및 B2B API 모두에 대해 행동 애널리틱스를 사용하는 것이 좋습니다. 특히 많은 주체가 관련되어 있어 정상적인 행동과 비정상적인 행동을 구분하는 과정이 어려울 경우 더욱 그렇습니다.

> **특정 사용자의 API 인증정보가 감염되었는지 어떻게 확인하나요?**

> **파트너가 계정 데이터를 훔치기 위해 인보이스 번호를 열거하는 방식으로 인보이스 발행 API가 악용되고 있는지 어떻게 확인하나요?**

B2B API와 동서 API를 보호하려면 IP 주소나 API 토큰 같은 기술적 요소만 분석해서는 얻을 수 없는 비즈니스 맥락이 필요합니다. 머신 러닝과 행동 애널리틱스를 사용해 비즈니스 관련 개체에 대한 가시성을 확보하는 것이 리스크를 효과적으로 이해하고 관리할 수 있는 유일한 방법입니다. 사용자나 파트너, 심지어 비즈니스 프로세스 주체(인보이스, 결제, 주문 등)와 같은 특정 주체의 정상적인 API 사용에 대한 비즈니스 맥락과 과거 벤치마크를 통해 다른 방법으로는 탐지되지 않는 비정상을 파악할 수 있습니다.

API 게이트웨이는 충분한 리스크 보호 기능을 제공하나요?

API에 대한 전략적 접근 방식을 취하는 많은 기업이 API 게이트웨이를 사용합니다. 대부분의 API 게이트웨이에는 기업이 활용해야 할 풍부한 통합 보안 기능이 있으며, 그중 첫 번째는 인증(OpenID Connect를 활용할 수 있는 경우 권한 부여도 포함)입니다. 하지만 API 게이트웨이에서 인증, 권한 확인, 할당량 관리만 하는 것은 여러 가지 이유로 충분하지 않습니다.

API 게이트웨이의 검색 격차: API 게이트웨이는 관리하도록 설정된 API에 대한 가시성과 제어 기능만 가지고 있기 때문에 섀도 API와 엔드포인트를 탐지하는 데 효과적이지 않습니다.

API 게이트웨이의 보안 격차: API 게이트웨이는 인증과 어느 정도 권한 부여 체계를 시행할 수 있지만, WAF와 WAAP처럼 페이로드를 검사하거나 남용을 탐지하기 위해 행동을 프로파일링하지는 않습니다.

가장 일반적인 API 설정 오류는 무엇일까요?

API가 사용되는 다양한 방법을 고려할 때 API가 잘못 설정될 수 있는 가능성은 거의 무한합니다. 그러나 잘못된 설정에는 몇 가지 공통된 주제가 있습니다.

인증이 손상되었거나 없는 경우

인증은 API를 통해 제공되는 민감한 데이터를 보호하기 위한 기본적인 요소입니다. 첫 번째 단계는 민감한 데이터를 전달하는 모든 API가 처음에 인증을 받았는지 확인하는 것입니다. 하지만 속도 제한을 통해 무차별 대입 공격, 크리덴셜 스터핑, 도난당한 인증 토큰의 사용으로부터 인증 메커니즘을 보호하는 것도 중요합니다. API 소비자가 인증 메커니즘을 우회할 수 있는 설정 오류는 때때로 토큰 관리와 관련해 발생할 수 있습니다(예: 일부 악명 높은 JWT 유효성 검사 문제 또는 토큰 범위를 확인하지 않는 경우).

손상된 권한

API의 가장 일반적인 용도 중 하나는 민감한 정보를 포함한 데이터나 콘텐츠에 대한 접속을 제공하는 것입니다. 권한 확인은 API 소비자에게 데이터를 제공하기 전에 API 소비자가 접속하려는 데이터에 접속할 수 있는 자격이 있는지 확인하는 프로세스입니다. 이는 오브젝트나 리소스 수준(예: 내 주문에는 접속할 수 있지만 다른 사람의 주문에는 접속할 수 없음)에서 수행하거나 기능 수준(관리 기능의 경우처럼)에서 수행할 수 있습니다. 엣지 케이스와 조건의 수가 많고 마이크로서비스 간에 API 호출이 발생할 수 있는 흐름이 다양하기 때문에 권한을 올바르게 확인하기는 쉽지 않습니다. 중앙 집중식 권한 부여 엔진이 없는 경우, API 구축에 BOLA 및 BFLA와 같은 취약점 중 일부가 포함되어 있을 가능성이 높습니다.

보안 설정 오류

위에서 언급한 인증 및 권한 확인 문제 외에도 안전하지 않은 통신(예: SSL/TLS 사용 실패 또는 취약한 암호 모음 사용), 보호되지 않는 클라우드 스토리지, 지나치게 허용적인 교차 오리진 리소스 공유 정책 등 여러 가지 종류의 보안 설정이 잘못될 수 있습니다.

리소스 부족 및 전송률 제한

API 소비자가 호출할 수 있는 횟수에 제한 없이 API를 구축하면 공격자가 시스템 리소스를 압도해 서비스 성능 저하나 전면적인 DoS로 이어질 수 있습니다. 인증 엔드포인트가 매우 중요한 경우, 최소한 인증되지 않은 엔드포인트에 대한 접속에 대해 속도 제한을 적용해야 하며 그렇지 않으면 무차별 대입 공격, 크리덴셜 스터핑, 인증정보 확인 공격이 발생할 수밖에 없습니다.

API 공격이란 무엇일까요?

API 공격은 악의적이거나 승인되지 않은 목적으로 API를 사용하려는 시도를 말합니다. API 공격은 다음과 같이 다양한 형태로 나타납니다.

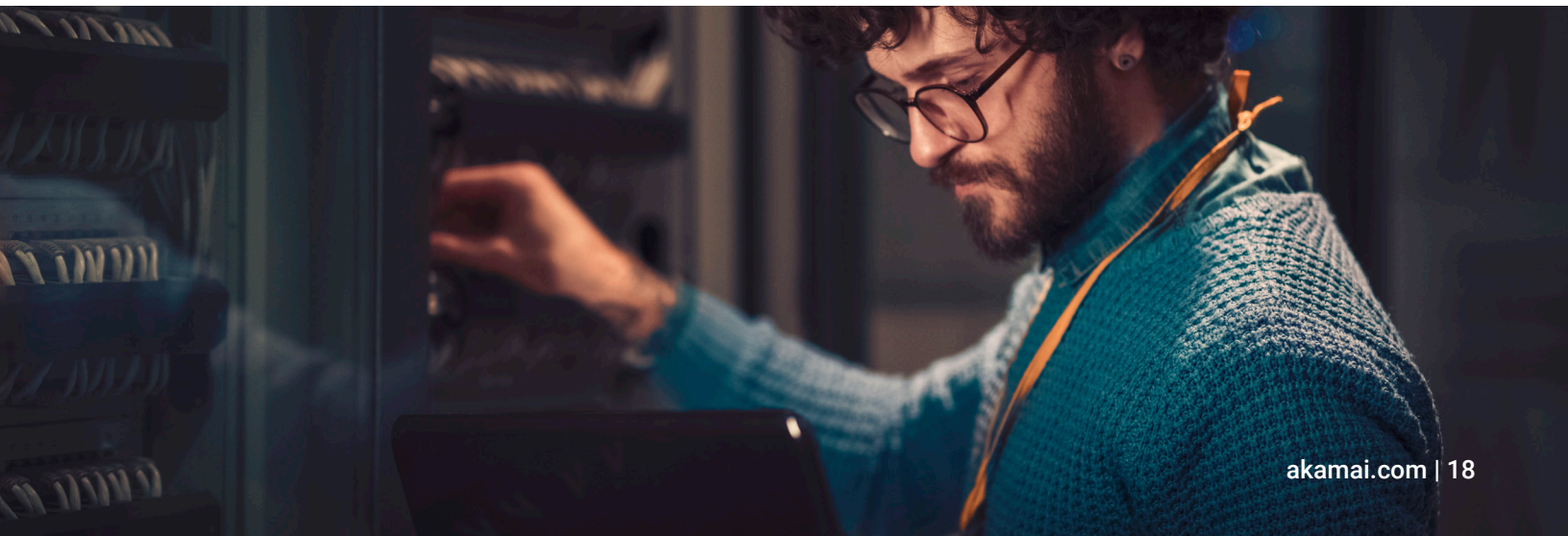
- API 구축의 기술적 취약점 악용
- 도난당한 인증정보 및 기타 계정 탈취 기술을 이용해 정상 사용자로 가장
- 예상치 못한 방식으로 API를 사용하기 위한 비즈니스 로직 악용

API에 대한 크리덴셜 스터핑이란 무엇일까요?

웹사이트와 SaaS(Software as a Service) 플랫폼에서 사용자 ID와 비밀번호 정보가 유출되는 사고가 빈번하게 발생하고 있습니다. 이러한 인시던트로 인해 대량의 인증정보 세트가 온라인에서 널리 공유되는 경우가 많습니다. 크리덴셜 스터핑은 이전에 침해된 웹사이트에서 유출된 인증정보를 사용해 다른 웹사이트에 자동 로그인을 시도하는 행위입니다. 이 방법은 일정 비율의 사용자가 여러 사이트에서 동일한 인증정보를 사용한다는 전제를 바탕으로 합니다. 점점 더 많은 공격자들이 API를 직접 공격하고 인증 메커니즘을 표적으로 삼고 있습니다. API는 쉽게 사용할 수 있도록 만들어졌기 때문에 공격자가 공격을 더 쉽게 자동화할 수 있습니다.

API를 통한 데이터 유출이란 무엇일까요?

데이터 유출은 성공적인 API 공격 및 악용의 결과로 발생하는 경우가 많습니다. 경우에 따라서는 공격자가 API 공격을 통해 훔친 매우 민감한 비공개 정보를 의미하기도 합니다. 그러나 집계적인 형태로 가치 있는 대규모 데이터 세트를 모으기 위해 공개적으로 사용 가능한 데이터를 공격적으로 스크레이핑하는 등 덜 심각한 종류의 API 악용에도 적용될 수 있습니다.



API 보안 솔루션 및 트렌드

API 보안의 최신 트렌드는 무엇일까요?

다음은 보안 담당 임원이 API 보안 전략을 개발할 때 고려해야 할 주요 트렌드입니다.

행동 애널리틱스 및 비정상 탐지: 가능한 공격을 예측하고 리스크를 방어하기 위해 시그니처 기반 탐지 및 사전 정의된 정책(예: WAF)에만 의존하는 대신 머신 러닝 및 행동 애널리틱스를 추가해 비즈니스 맥락에서 API 활동을 보고 비정상을 탐지하는 기업이 점점 더 많아지고 있습니다.

온프레미스에서 SaaS로 전환: 많은 1세대 API 보안 제품이 온프레미스에 배포되었지만 속도와 배포 용이성, 대규모로 머신 러닝의 힘을 활용할 수 있는 능력으로 인해 SaaS 기반 접근 방식이 인기를 얻고 있습니다.

장기간에 걸친 분석: 개별 API 호출이나 단기 세션 활동만 분석하는 API 보안 접근 방식은 기본적인 자동화된 WAF 정책 최적화 완료부터 행동 애널리틱스 수행과 비정상 탐지에 이르기까지 며칠, 때로는 몇 주에 걸쳐 API 활동을 분석하는 플랫폼으로 대체되고 있습니다.

DevSecOps - 비보안 이해관계자 포용: API 리스크를 줄이는 가장 좋은 방법 중 하나는 API 보안 전략 및 툴과 API를 생성, 구축, 설정하는 데 관여하는 개발자 및 시스템 간의 연결성을 강화하는 것입니다.

API 기반의 API 보안: 현재 진행 중인 API 공격과 악용 사례를 탐지하고 방어하는 것이 중요하지만, 미래 지향적인 기업은 API 보안 데이터와 인사이트에 대한 온디맨드 접속을 사용해 위협 탐색, 인시던트 대응, API 개발 관행을 개선하는 방법을 찾고 있습니다.

시그니처 기반의 API 보안이란 무엇일까요?

시그니처 기반의 API 보안 기술은 알려진 공격 특성과 패턴을 모니터링하고, 일치하는 것이 관찰되면 보안 알림과 기타 자동화된 대응을 생성합니다. 이는 WAF의 전형적인 방식입니다. 장점은 기업이 감염되었거나 비정상적으로 동작하는 수신 API 트래픽에 대한 알림을 받으면 시그니처 기반 API 보안을 사용해 즉시 차단할 수 있다는 것입니다.

공격 시그니처 패턴을 학습하는 머신 러닝을 통해 고급 탐지 기능을 제공하고 규모에 맞게 민첩성을 유지할 수 있는 대규모 WAAP 솔루션의 일부인 WAF를 찾아야 합니다. 행동 애널리틱스와 맞춤형 대응을 제공하는 API 보안 솔루션과 통합된 WAAP를 찾아 두 가지 장점을 모두 활용합니다. 이러한 솔루션을 함께 사용하면 내부 및 외부에서 완벽한 API 가시성, 탐지, 대응을 제공합니다.

API 탐지 및 대응이란 무엇일까요?

API 탐지 및 대응은 기록 데이터에 대한 심층 분석에 초점을 맞춘 새로운 API 보안 카테고리이며 다음을 지원합니다.

- 모든 API 소비자의 행동에 대한 기준 결정
- API 악용과 오용 가능성을 나타내는 공격 및 비정상 탐지

대규모의 효과적인 API 탐지 및 대응은 리소스 집약적인 AI 및 머신 러닝 기술이 필요한 대규모 데이터 세트와 관련이 있기 때문에 SaaS 모델에서만 제공할 수 있습니다.

최신 API 위협 방어란 무엇일까요?

최신 API 위협 방어는 행동 애널리틱스와 위협 탐색을 결합한 SaaS 기반의 API 보안 접근 방식이며 다음을 지원합니다.

- 섀도 또는 좀비 API를 비롯한 기업에서 사용 중인 모든 API 검색
- 머신 러닝을 적용해 API 사용 및 악용 실태에 대한 비즈니스 맥락을 오버레이
- API 및 API 활동 데이터에 대한 행동 분석 및 위협 탐색

API 보안 플랫폼이란 무엇일까요?

API 보안 플랫폼은 특별히 다음과 같은 용도로 설계된 SaaS 기반 제품입니다.

- 전사적으로 사용 중인 모든 API에 대한 지속적으로 업데이트된(제재 여부와 관계없이) 인벤토리 생성
- API와 그 사용 현황을 분석해 비즈니스 맥락을 파악하고 예상되는 행동의 기준선 결정
- 비정상적인 API 사용을 탐지하고, 필요한 경우 보안 정보 및 이벤트 관리(SIEM) 및 보안 오케스트레이션, 자동화 및 대응(SOAR) 워크플로우에 알림과 지원 데이터 제공
- 보안 및 비보안 이해관계자 모두에게 API 인벤토리, 활동, 위협 정보에 대한 온디맨드 접속 제공

API 보안 회사란 무엇일까요?

API를 보다 전략적으로 사용하는 IT 및 보안 리더는 전문 API 파트너와 협력해야 할 수도 있습니다. API 회사의 가장 일반적인 세 가지 종류는 다음과 같습니다.

- 중앙에서 API 호출을 수락하고 적절한 백엔드 리소스와 마이크로서비스로 라우팅하는 기술을 제공하는 API 게이트웨이 회사
- 기업이 모든 활성 API와 잠재적 리스크를 인지하고, 공격 및 악용 사례를 탐지하도록 하고, 포괄적인 보안 테스트를 지원하고, API 사용 방식에 대한 풍부한 데이터를 제공할 수 있는 API 보안 플랫폼 기업
- API 트래픽 데이터를 원활하게 전송하는 동시에 플랫폼 안팎에서 API를 검색할 수 있는 기능을 제공해 벤더사 통합과 디지털 격차 해소에 이상적인 WAAP 및 API 보안 플랫폼 회사

API에서 위협 탐색이란 무엇일까요?

위협 탐지는 알려지지 않았거나 이전에 탐지되지 않은 위협을 적극적으로 찾는 작업입니다. 이러한 선제적인 접근 방식은 이전에는 볼 수 없었던 새로운 위협을 식별하고 심각한 피해를 입기 전에 방어하는 데 매우 중요합니다. 위협 탐지에 사용되는 핵심 기술 중 하나는 행동 분석입니다. 여기에는 의심스럽거나 비정상적인 활동을 식별하기 위해 API 행동을 분석하는 것이 포함됩니다. 예를 들어, API가 단기간에 갑자기 수천 개의 레코드를 요청하는 경우 API 비즈니스 로직이 감염되었음을 나타낼 수 있습니다. 최신 API 보안 솔루션은 보안팀이 가능한 위협을 조기에 식별하고 대응 조치를 취할 수 있도록 특정 위협 탐색 기능을 제공합니다.

WAAP란 무엇일까요?

웹 애플리케이션 및 API 보안(WAAP)은 리서치 기관인 Gartner가 업계에서 새롭게 등장하는 웹 및 API 보안 솔루션에 사용하는 카테고리입니다. 이는 API 보안의 전략적 중요성이 커지고 WAF 플랫폼이 매니지드 SaaS로 클라우드로 이동함에 따라 WAF 시장의 범위가 처음보다 확장된 데 따른 것입니다.

API 문서화의 사례는 무엇일까요?

웹 API의 가장 일반적인 종류인 RESTful API에 대한 가장 일반적인 형태의 API 문서는 OpenAPI 사양에 기반한 Swagger 파일 모음입니다. API 문서는 API를 설계하거나 구축할 때 개발자가 작성하는 것이 이상적입니다. 그러나 현실에서는 API 문서가 오래된 경우가 많아 실제 API 사용과 문서가 일치하지 않는 경우가 많습니다. 이 문제를 해결하기 위해 일부 API 보안 플랫폼은 실제 API 활동에서 Swagger 파일을 생성해 문서화된 내용과 실제 배포된 내용 사이의 차이를 강조할 수 있으며, 이는 모든 API 리스크 평가에서 필수적인 구성요소입니다.

기업이 확인해야 할 API 보안 체크리스트가 있나요?

효과적인 API 보안을 위해서는 기업에 맞는 많은 세부 단계와 지속적인 관행이 필요합니다. 다음은 보안팀이 API 보안을 개선할 때 출발점으로 삼을 수 있는 API 체크리스트입니다.

- API 보안 접근 방식에 전사적으로 지속적인 API 검색을 위한 메커니즘이 포함되어 있나요?
- API 체계 관리가 기업의 광범위한 보안 및 리스크 관리 관행에 통합되어 있나요?
- 특정 데이터 센터 또는 클라우드 인프라 모델에 종속되지 않는 범용 API 보안 접근 방식을 구축하고 있나요?
- 이러한 접근 방식이 관측되는 API 활동과 발생 가능한 리스크를 제대로 이해하는 데 필요한 비즈니스 맥락을 제공하나요?
- API 보안 플랫폼과 SIEM/SOAR, 위협 탐색, 문서화, DevOps 툴 등과 같은 기타 관련 비즈니스 프로세스 간의 양방향 자동화를 위한 전략이 있나요?
- 개발자 같은 비보안 이해관계자를 API 보안 툴과 프로세스에 참여시키기 위한 조치를 취하고 있나요?

Akamai 보안은 성능이나 고객 경험에 영향을 주지 않으면서 모든 상호 작용 지점에서 비즈니스의 원동력이 되는 애플리케이션을 보호합니다. 글로벌 플랫폼의 규모와 위협에 대한 가시성을 활용해 고객과 협력함으로써 위협을 예방, 탐지, 방어하고 브랜드 신뢰를 쌓고 비전을 실현할 수 있도록 지원합니다. Akamai의 클라우드 컴퓨팅, 보안, 콘텐츠 전송 솔루션에 관한 자세한 정보를 보려면 akamai.com과 akamai.com/blog를 방문하거나 X(기존의 Twitter)와 LinkedIn에서 Akamai Technologies를 팔로우하시기 바랍니다. 2024년 9월 발행.