CLOUDFLARE

2024년 API 보안 및 관리 보고서

목차

섹션을 클릭하여 해당 페이지로 건너뛰기

핵심 요약

인터넷은 컴퓨터 사이에서 끝없이 이어지는 대화의 흐름입니다. 이러한 대화는 새로운 방식으로 소프트웨어 및 앱과 상호작용할 수 있게 해주는 애플리케이션 프로그래밍 인터페이스(API)를 사용하여 이루어지는 경우가 많습니다. 예를 들어 OpenAI의 ChatGPT API를 사용하여 Slack은 채팅 기반 워크플로우를 간소화할 수 있고 Booking.com은 더욱 맞춤화된 여행 계획 경험을 제공할 수 있습니다.

오늘날 API는 다른 인터넷 트래픽을 앞질러, 작년에는 Cloudflare[1]에서 처리한 **동적 인터넷 트래픽의 절반 이상(57%)을 구성**했습니다.

하지만 본 **2024년 API 보안 및 관리 보고서**에서 다룬 것과 같이, API를 관리하고 남용으로부터 보호하는 것은 점점 더 복잡해지고 있습니다.

예를 들어 자체의 API에 대한 정확한 정보가 부족한 조직이 많습니다. Cloudflare에서는 머신 러닝 기반 검색을 통해 조직에서 자체적으로 보고한 것보다 API 엔드포인트를 30.7% 더 많이 발견했습니다.[2]

30.7%

만큼 API 엔드포인트가 더 많음

안타깝지만, 조직에서는 보이지 않는 것을 적절히 방어할 수는 없습니다.

API 환경을 정확하게 실시간으로 파악하지 않고 API 보안을 구현하는 조직에서는 **의도치 않게 정상 트래픽을 차단할 수 있습니다.**

"요청이 너무 많음"(429) 오류 코드는 2023년에 Cloudflare에서 완화한 **1위 API 클라이언트 오류 범주**입니다. 429 코드가 곧 *공격자의 요청이 너무 많다*는 의미는 아닙니다. 예를 들어서 오류를 일으킨 레이트 리미팅을 원래는 분산 서비스 거부(DDoS) 공격 때문에 설정한 경우, 레이트 리미팅을 지나치게 광범위하게 시행하면 정상적인 사용자를 차단할 수도 있습니다(Cloudflare 고객의 **1순위 API 완화 방법은 DDoS 방어**라는 점을 참고하세요).

이 보고서의 목표는 조직에서 **조직 API 엔드포인트 관리 상태를 전체적으로 평가**할 수 있는 유용한 벤치마크를 제공하는 것입니다. 결국 가시성, 성능, 위험을 관리하려면 API 보안에서 데이터를 통합해야 하니까요.

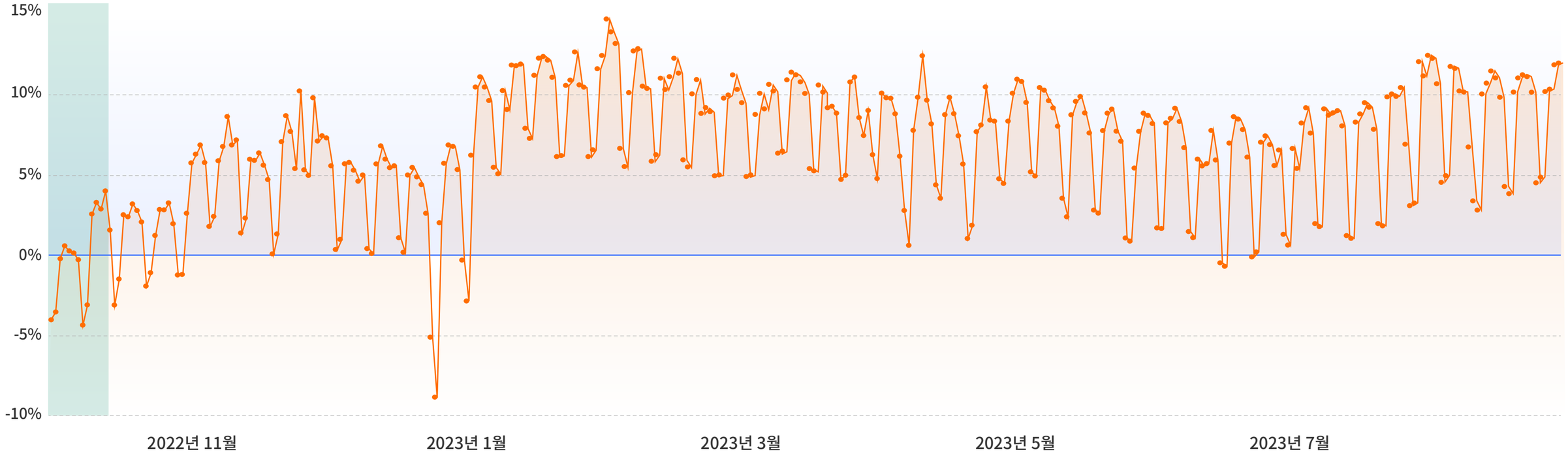
연구 방법

이 보고서의 결과는 2022년 10월 1일부터 2023년 8월 31일 사이에 Cloudflare의 전역 네트워크(Cloudflare의 웹 애플리케이션 방화벽, DDoS 방어, 봇 관리, API 게이트웨이 서비스 포함)에서 관찰한 트래픽 패턴을 집계한 내용을 바탕으로 합니다. Cloudflare에서는 초당 평균 5,000만 건이 넘는 HTTP 요청을 처리하며 하루 평균 1,700억 건의 사이버 위협을 차단합니다.

스냅샷: 전 세계 API 관련 트래픽

시간에 따른 전 세계 API 트래픽의 증가

기준치는 강조 표시되어 있으며, 200 응답 코드 및 동적 캐시만 해당합니다



2022년 10월 1일부터 2023년 8월 31일 사이에 성공적으로 응답한(상태 코드 200) API 트래픽은 Cloudflare의 동적 HTTP 트래픽 중 53.1%~60.1%를 차지했습니다.
동적 콘텐츠는 방문 시간, 위치, 장치 등 사용자에 특정한 요인에 따라 변경되는 콘텐츠를 말합니다.

핵심 결과

다른 인터넷 트래픽을 앞지르는 API

성공적인 API 요청은 Cloudflare에서 처리한 **인터넷 트래픽 중 57%**(동적 HTTP 트래픽)를 차지합니다.[1]

1순위 완화 방법: DDoS 방어

API 완화 중 1/3**(33%)은 분산 서비스 거부(DDoS) 공격 차단으로 이루어졌습니다.**[4]

알려지지 않은 공격면

머신 러닝 모델 발견 결과에 따르면 조직에서 자체 보고한 것보다 약 **1/3(30.7%) 이상의 API 엔드포인트**가 있습니다.[2]

산업별 차이

산업 중 **API 트래픽 비율이 가장 높은 산업에는 IoT 플랫폼, 철도/버스/택시, 법률 서비스, 멀티미디어/게임, 물류/공급망**이 포함됩니다.[5]

1순위 오류: 요청이 너무 많음

절반 이상**(51.6%)의 API 오류 비율은 "요청이 너무 많음"**(429 오류)으로 구성되어 있습니다.[3]

지역별 차이

API 트래픽 비율이 가장 높은 곳은 **아프리카와 아시아**입니다. API 트래픽은 **중동에서 가장 다양하게 나타납니다.**[6]

목차

숨겨진 공격면

기업의 경우, API는 경쟁 우위의 원동력입니다. 비즈니스 인텔리전스가 많아지고, 클라우드 배포가 더 빨라지며, 새로운 AI 기능을 통합할 수 있습니다. 하지만 API를 최적화하기 위한 첫 번째 단계는 인터넷에 노출된 호스트 이름과 모든 API 엔드포인트의 전체 목록을 갖추는 것입니다.

존재하는지도 모른다면 조직에서 API를 관리하거나 보호할 수는 없습니다. 그리고, **전체 API 목록을 갖추지 못한 조직이 많은 것으로 드러났습니다**.

- **Cloudflare에서 머신 러닝을 통해 발견한 API REST 엔드포인트는** 고객이 제공한 세션 식별자로 발견한 것보다 **약 31% *더* 많았습니다**.[2]
- Cloudflare를 사용하는 ***15,000개 이상의 계정***에서는 머신 러닝 방법을 통해서만 API 엔드포인트가 드러났습니다.[7]

사용하는 조직에서 관리하지 않거나 보호하지 않는 API, 즉 **'섀도우' API**는 개발자나 개인 사용자가 특정한 비즈니스 기능을 실행하기 위해 도입하는 경우가 많습니다.

본래부터 악의적인 것은 아니지만, **섀도우 API는 본질적으로 보호되지 않은 공격면으로서 새로운 위험을 초래합니다**.

섀도우 API가 악용되면 데이터 노출, 패치되지 않은 취약점, 데이터 규제 준수 위반, 내부망 이동, 기타 위협을 초래할 수 있습니다.

가시성 확인

지금 API를 어떻게 발견하여 분류하고 있나요?

조직의 API 목록이나 개발자의 API 목록은 유효한 API 요청 및 응답 규격을 정의하는 메타데이터인 API 스키마를 통해 파악되고 있습니다. 이러한 API 스키마(OpenAPI 규격으로 문서화 되는 경우가 많음)에는 API 호스트, HTTP 메서드, 경로, 기타 개발자가 수립한 요구 사항(예: 경로, 쿼리 변수)이 포함됩니다.

목차

섀도우 API의 위험

Cloudflare에서 보면 API 관리 여정의 초기 단계에 있는 조직은 "이메일 및 요청" 접근 방식을 사용하는 경우가 많습니다. 그러면 다음 코드 릴리스에 따라 변경될 가능성이 있는 시점별 목록이 만들어집니다. 하지만 이렇게 수동적으로 접근하는 방식은 일반적으로 조직 내 지식에 의존하며, 수동 방식으로 인한 오류가 생기기 쉽습니다.

API로 인해 벤더가 특정 시스템에 액세스할 수 있다는 것을 의료 조직의 IT 팀에서는 모르고 있다고 가정해 보겠습니다. 벤더 손상이 이루어질 경우 공격자는 API를 남용하여 환자 데이터를 빼낼 수도 있습니다.

예를 들어 2019년에 벌어졌던 Quest Diagnostics 데이터 유출에서는 청구 벤더에게 정보를 전송하는 API에 권한 없는 사용자가 액세스할 수 있게 되면서 약 1,200만 명의 환자 데이터가 노출되었습니다.

2022년에는 호주 통신 공급자 Optus에 대한 침해가 이루어졌고, 이는 공격자가 인증되지 않은 API를 통해 고객 데이터베이스에 액세스했기 때문이라고 보고되었습니다.

API 경제가 성장하면서 API 개발, 관리, 보안 측면에서 손실의 문제, 관리, 복잡성 문제도 늘어나고 있습니다.

보안 확인

어떤 API가 '쓰기' 액세스를 허용하는지 어떻게 모니터링하고 있나요?

Cloudflare에서 모든 계정 API에 걸쳐 집계한 결과, **조직 중 59.2%는 최소 절반의 API에 '쓰기' 액세스를 허용하고 있었습니다.**[8]

'읽기 전용'(GET) 액세스 API는 시스템에서 정보를 가져오고 수집합니다. 하지만 '쓰기'(POST, PUT, DELETE) API를 사용하면 사용자와 다른 앱에서 업데이트(변경)를 시스템에 푸시할 수도 있습니다.

많은 API 침해는 사용자에게 너무 많은 권한을 부여하거나, 다른 사용자의 데이터에 액세스할 수 있게 하는 등 허용 권한으로 인해 발생합니다. API가 잘못된 사람에게 '쓰기' 액세스 권한을 제공할 경우 본 보고서에서 설명한 것과 같은 공격이 초래될 수 있습니다.

일반적인 API 오류

조직에서 API 엔드포인트를 정확하게 발견했다면(그런 다음 저장하거나 삭제했다면) 무엇이 잘 작동하고 무엇이 그렇지 않은지 파악해야 합니다. API 오류는 결과적으로 정상적인 비즈니스에 방해가 되는 사이버 공격이나 앱 성능 문제를 알리는 신호일 수 있습니다.

HTTP 상태 코드는 앱이 잘 작동하는지, 또는 오류가 있는지 나타낼 때 가장 자주 사용되는 3자리 코드입니다.

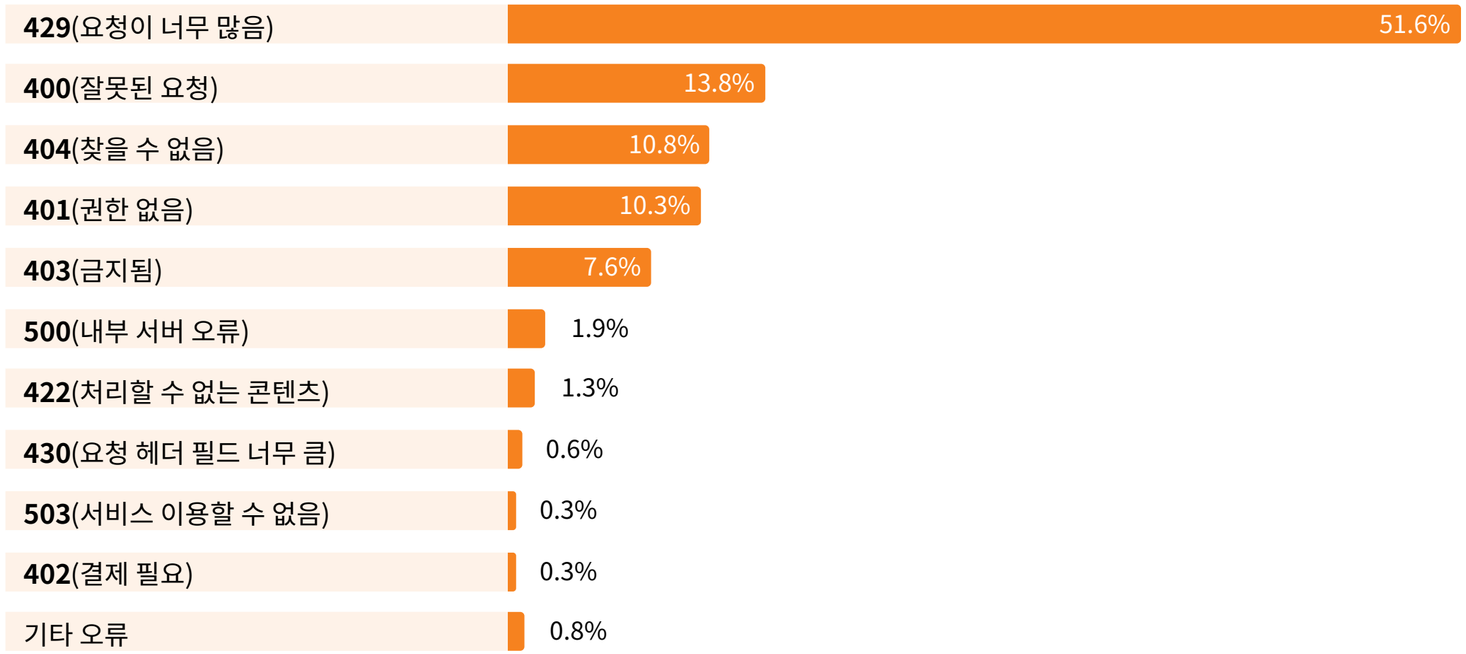
API 및 기타 HTTP 요청의 경우 '2'로 시작하는 상태 코드(2xx 성공 코드)는 클라이언트 동작을 수신했고 이해했으며 수락했음을 나타냅니다(즉, 성공함).

하지만 앱 방문자가 의도한 대상에 도달하지 못할 경우, 그 대신 리디렉션 되거나(3xx 리디렉션), 4xx 클라이언트 오류 또는 5xx 서버측 오류를 겪게 될 수 있습니다.

Cloudflare에서는 API 원본에서 1조 개의 트래픽 오류를 관찰했는데, **절반이 넘게(51.6%) '429' 코드, 즉 "요청이 너무 많음"으로 구성되어 있었습니다.**[3]

"레이트 리미팅"이라고도 하는 429 오류는 서버에 따라 특정 시간 동안 클라이언트가 요청을 너무 많이 전송했을 때 발생합니다.

일반적인 API 오류



오류 설명은 부록을 참조하세요

API 오류 오진단 위험

429 오류(위에서 언급한 것처럼 가장 자주 발생하는 API 오류)는 특정 동작이 일어났을 때(예: 특정 IP 주소에서 /login 엔드포인트에 분당 특정 요청 수를 초과함) 서버에서 API 트래픽을 자동으로 조절했다는 의미입니다.

하지만 조직에서 수동으로 설정한 레이트 리미팅을 사용하는 경우(적응형 레이트 리미팅 대신) 빠르게 상황에 뒤떨어질 수 있습니다. 마케팅 캠페인이 성공하면서 /login 엔드포인트에 평균보다 높은 트래픽이 발생했고, 공격 상황이 아니라면 어떻게 될까요? 이 상황에서는 수동 레이트 리미팅으로 인해 정당한 트랜잭션이 차단될 수도 있습니다.

원인이 종종 '오진단'되는 다른 오류 예시는 **401 "권한 없음" 오류(Cloudflare가 API 트래픽에서 관찰한 네 번째로 흔한 오류)**입니다.

401은 사용자의 자격 증명이 존재하지 않거나 요청된 소스에 적절한 수준의 액세스가 없음을 의미합니다. 하지만 다른 HTTP 오류 코드와 마찬가지로 이 코드는 위협 때문일 수도 있고(예: 전체 계정 탈취로 이어질 수 있는 취약한 개체 수준 권한 부여 공격 시도), 그냥 정당한 사용자가 실수로 자격 증명을 잘못 입력했기 때문일 수도 있습니다.

API 트래픽 '오진단'의 한 가지 예는 2023년 초 Google의 경고인데, Google은 웹 사이트 소유자와 일부 콘텐츠 전송 네트워크에 (정당한) Googlebot의 크롤링 속도를 제한하기 위해 잘못된 상태 오류를 사용하는 것을 조심하라고 경고했습니다.

Google에서는 사용자에게 *"클라이언트 오류는 바로 클라이언트 오류를 의미합니다 … 클라이언트 요청이 어떤 방식으로 잘못되었다는 의미일 뿐입니다"*라고 알렸습니다.

성능 확인

API 오류를 어떻게 모니터링하고 평가하나요?

모든 API 오류가 공격으로 인해 생기는 것은 아닙니다. API 오류의 근본 원인(및 문제 이면의 동향)을 이해하려면 API 트래픽을 지속적으로 기록하고 시간에 따른 동향을 분석해야 합니다.

레이트 리미팅이 적용되는 API 트래픽이 얼마나 되는지 아시나요? (잘못된 권한 부여로 인해) 얼마나 많이 금지되고 있나요? 오류가 공격 때문에 발생했는지, 만료된(또는 잘못 입력한) 사용자 자격 증명 때문에 발생했는지 확인하고 있나요?

최상위 API 보안 취약점

API는 남용으로부터 보호하기 어렵습니다. 다른 웹 애플리케이션 보안 서비스와 비교하면 더 심층적인 비즈니스 맥락, 검색 방법, 액세스 인증 제어가 필요합니다.

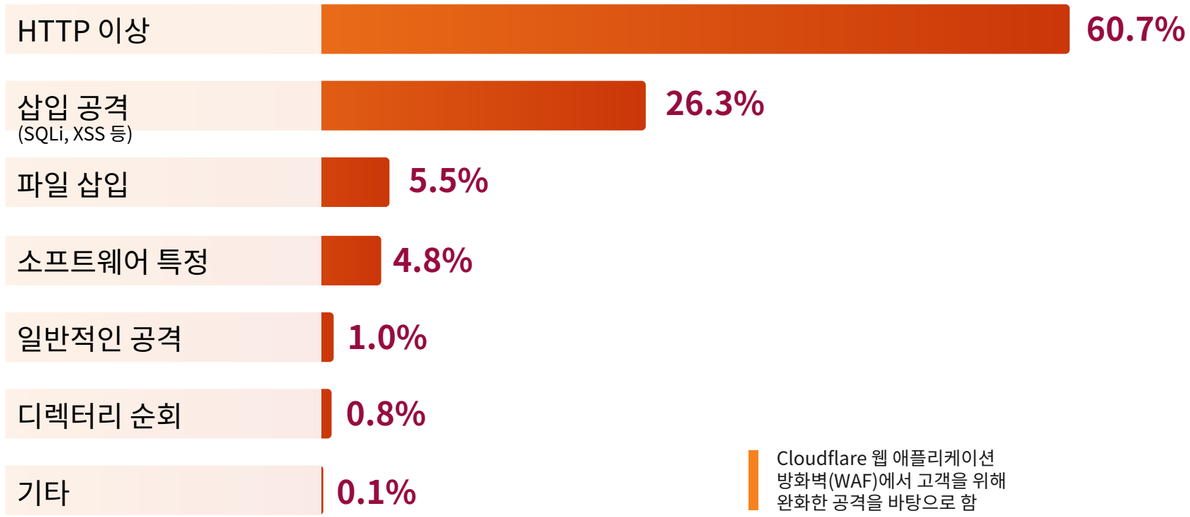
예를 들어서 다음을 생각해 보세요.

웹 애플리케이션	API
최종 사용자가 **볼 수 있습니다**	앱 사용자가 **볼 수 없습니다**
웹 브라우저를 통해 **최종 사용자가 액세스**합니다	**시스템 및 앱이** 데이터를 교환할 수 있습니다
다양한 사용자 상호 작용을 통해 백엔드(HTML, CSS, JavaScript 사용)의 **데이터를 시각화합니다**.	정의된 형식(가장 흔하게는 RESTful JSON, gRPC, XML, GraphQL)을 사용해 서버 및 애플리케이션에 액세스하여 **데이터를 전송합니다**.
일반적으로 알려진 악의적 트래픽을 차단하는 **'소극적 보안'** 모델을 통해 보호합니다.	검증 및 인증된 트래픽만을 허용하는 **'적극적 보안'** 모델을 통해 더 효과적으로 보호합니다.

이러한 이유(및 다른 이유)로 인해 보안 위협을 즉각 파악하고 해결하려면 API를 정기적이고 자동적으로 모니터링하는 것이 중요합니다.

아래는 API를 대상으로 한 위협 중 2023년에 Cloudflare에서 고객 대신 가장 자주 완화한 위협 스냅샷입니다[9].

상위 API 위협



이러한 공격 유형에 대한 자세한 설명은 부록에서 확인할 수 있습니다.

한 MDM 공격에서 API 취약점의 역할

모바일 장치 관리(MDM)를 사용하면 조직에서 지리적으로 분산된 장치를 하나의 플랫폼에서 모두 관리할 수 있습니다. MDM을 사용하면 IT 팀에서 장치의 기본 제공 API를 통해 관리되는 장치에 앱을 배포하고 제어할 수 있습니다.

하지만 MDM 시스템의 간편함과 편리함은 위험과 함께 따져보아야 합니다. 공격자는 MDM 시스템을 사용해 모바일 장치 수천 대에 높은 수준으로 액세스할 수 있으므로, MDM 시스템은 매력적인 대상입니다.

2023년 8월, 사이버보안 및 인프라 보안국(Cybersecurity and Infrastructure Security Agency, CISA)과 노르웨이 국립 사이버 보안 센터(Norwegian National Cyber Security Centre, NCSC-NO)에서는 공동 사이버 보안 권고를 발행하여 **이전에 MobileIron Core로 알려졌던 Ivanti EPMM(Endpoint Manager Mobile)을 악용하기 위해** 공격자들이 두 가지 취약점을 연결하고 있음을 경고했습니다.

공격자는 다음 차트에 요약되어 있는 MITRE ATT&CK® 기법 등 다양한 방법을 사용했습니다. MITRE ATT&CK 및 OWASP 상위 10위 API 보안 등의 프레임워크를 준수하면 더욱 강력한 API 방어를 포함해 더 탄력성 있는 사이버 보안을 위한 강력한 기반을 마련하는 데 도움이 됩니다.

기법 예시 *(전체 목록은 여기)*	**사용**
공개된 애플리케이션 악용	최소 2023년 4월부터 공격자는 공개된 Ivanti EPMM 장비에서 CVE-2023-35078을 악용했습니다.
명령 및 스크립팅 인터프리터	공격자는 EPMM 장치에 웹셸을 업로드하고 명령을 실행하기 위해 CVE-2023-35081을 악용했을 수 있습니다.
계정 검색: 도메인 계정	공격자는 EPMM 장치 사용자 및 관리자를 수집하기 위해 CVE-2023-35078을 악용했습니다. **이 상황에서 공격자는 API 경로** /mifs/aad/api/v2/authorized/users**를 이용하여 EPMM 장치에서 사용자와 관리자를 확인했습니다**.
원격 시스템 검색	공격자는 LDAP 엔드포인트를 검색했습니다.
서버 소프트웨어 구성 요소: 웹셸	공격자는 손상된 인프라에 웹셸을 심었습니다.
프록시	공격자는 손상된 SOHO 라우터를 활용해 인프라를 프록시하고 손상을 입혔습니다.

일반적인 API 취약점을 완화하는 두 가지 방법

1. 스키마 유효성 검사

누락된 사용자 에이전트(최종 사용자를 위해 인터넷 콘텐츠를 검색하는 소프트웨어), 잘못된 형태의 메서드 이름, 비표준 포트 등 HTTP 이상은 악의적 요청을 나타내는 일반적인 신호입니다. 앞서 언급한 바와 같이 이러한 HTTP 이상 유형은 Cloudflare에서 완화한 API 위협의 대부분을 이룹니다.

스키마 유효성 검사는 API 서버로 향하는 각 API에 "클린" 트래픽만 허용하기 위해 HTTP 이상을 구분하기 좋은 방법입니다. API 스키마는 대상 엔드포인트, 경로 또는 쿼리 변수 형식, HTTP 메서드 등 여러 가지 요청 속성에 따라 어떤 API 요청이 유효한지 정의합니다.

2. 인증 허점 처리하기

API와 관련된 과거의 데이터 유출 뉴스 헤드라인에서 볼 수 있듯 퍼블릭 API의 인증 부족(또는 결함)은 또 다른 심각한 문제입니다.

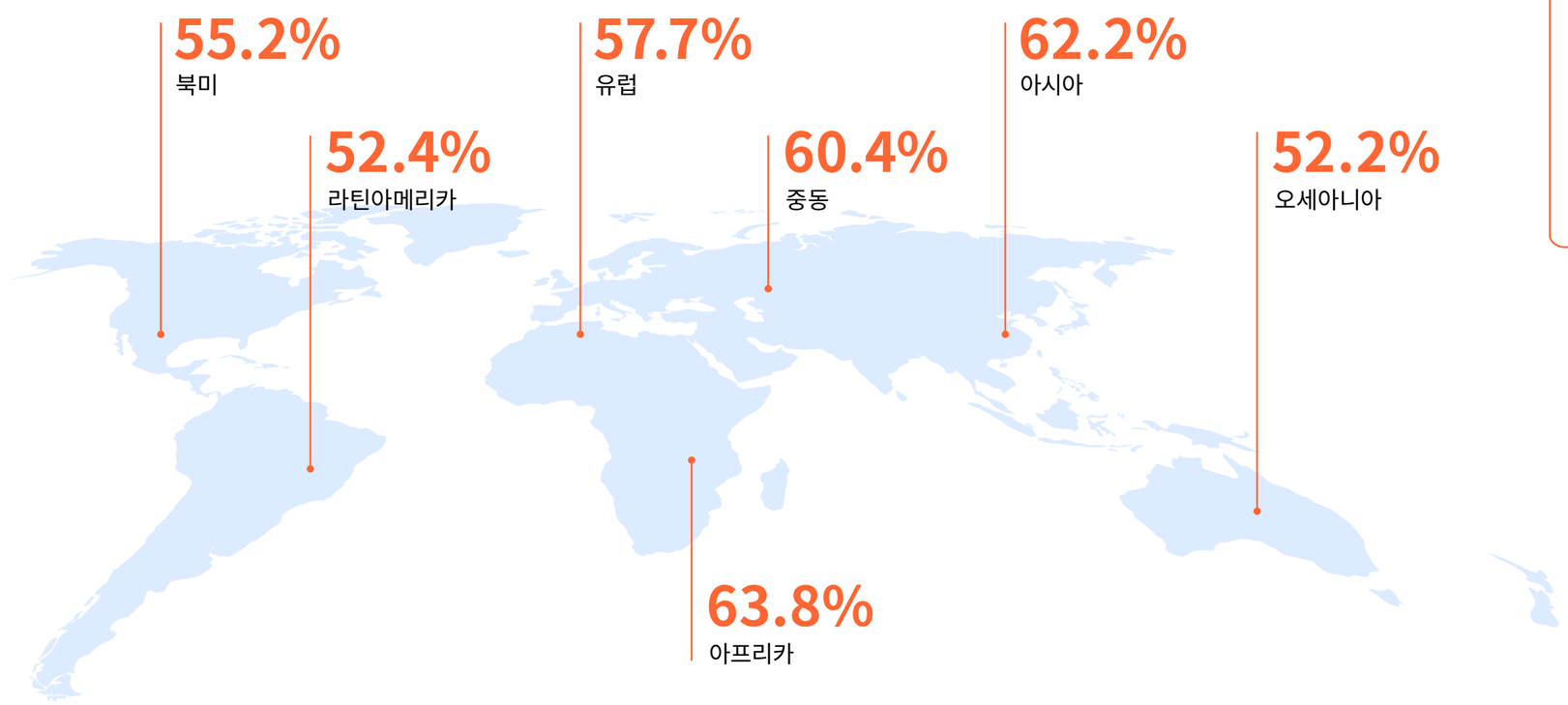
API를 통해 중요한 데이터를 노출시킬 수도 있는 인증 허점을 처리하는 데는 네 가지 방법이 있습니다.

- 먼저, 기업에서 예외를 승인하지 않은 한 공개적으로 액세스할 수 있는 API마다 인증을 시행합니다
- 서버에 대한 API 요청 속도를 제한하여 잠재적인 공격자의 속도를 늦춥니다
- 중요한 데이터가 비정상적인 양으로 유출되는 것을 차단합니다
- 공격자가 정당한 API 요청 시퀀스를 건너뛰지 못하게 차단합니다

API 중심 세상

지역 동향

Cloudflare에서 보호하는 각 지역에서 API 트래픽은 해당 지역의 동적 HTTP 트래픽 중 절반 이상을 차지했습니다[10].



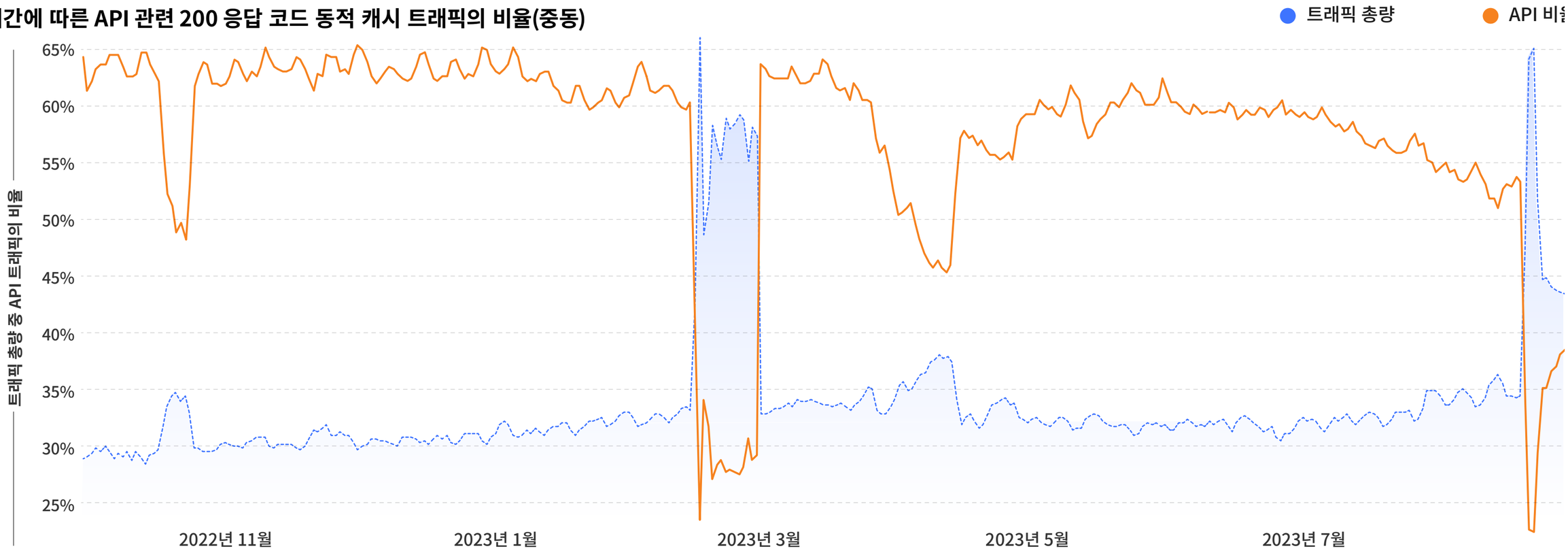
전체적으로 API 트래픽 총량은 2023년 내내 꾸준히 늘었습니다. 하지만 다음 지역에서는 눈에 띄는 변동이 있었습니다.

- **라틴아메리카**에서 API 트래픽은 동적 HTTP 트래픽 중 **46.1%~58.6%**를 차지했습니다
- **오세아니아**에서 API 트래픽은 동적 HTTP 트래픽 중 **44.1%~57.4%**를 차지했습니다
- 그리고 **중동**에서 **API 트래픽은 가장 다양하게 나타났는데**, 다음 섹션에서 이를 살펴봅니다.

중동 트래픽 급증

중동에서 API 트래픽이 크게 변화한 것은 **익명성 도구에 대한 전체 트래픽**이 갑자기 일시적으로 급증한 시기와 맞물립니다(이 도구는 네트워크 제한을 우회하는 데 도움이 된다고 알려져 있음). Cloudflare에서 관찰한 결과, 2023년에 익명성 도구에 대한 트래픽이 급증한 시기는 정부 지시 인터넷 셧다운 직후입니다.

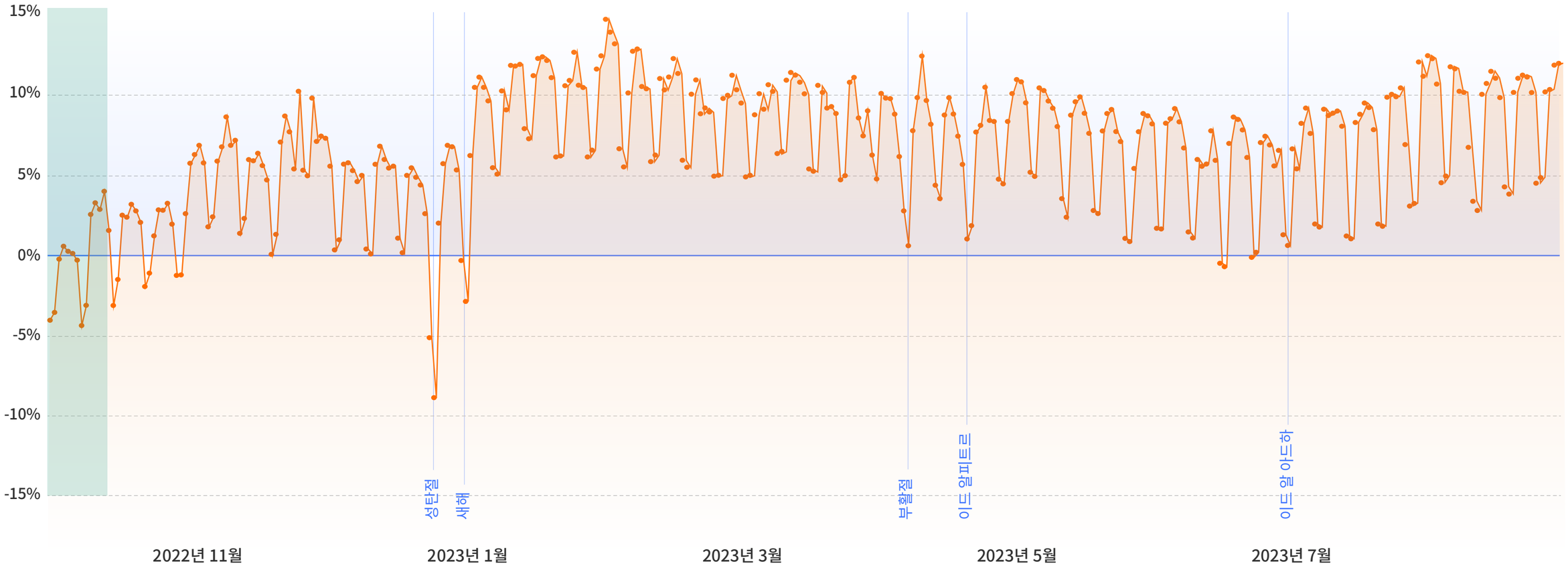
시간에 따른 API 관련 200 응답 코드 동적 캐시 트래픽의 비율(중동)



API 트래픽은 느려지는가?

API 트래픽을 봇 사이의 대화로 생각하는 경우가 많지만, Cloudflare의 데이터를 보면 한 해 동안, 특히 중요한 휴일 즈음에 API 트래픽이 급증하고 급락하는 것이 명확하게 나타납니다.

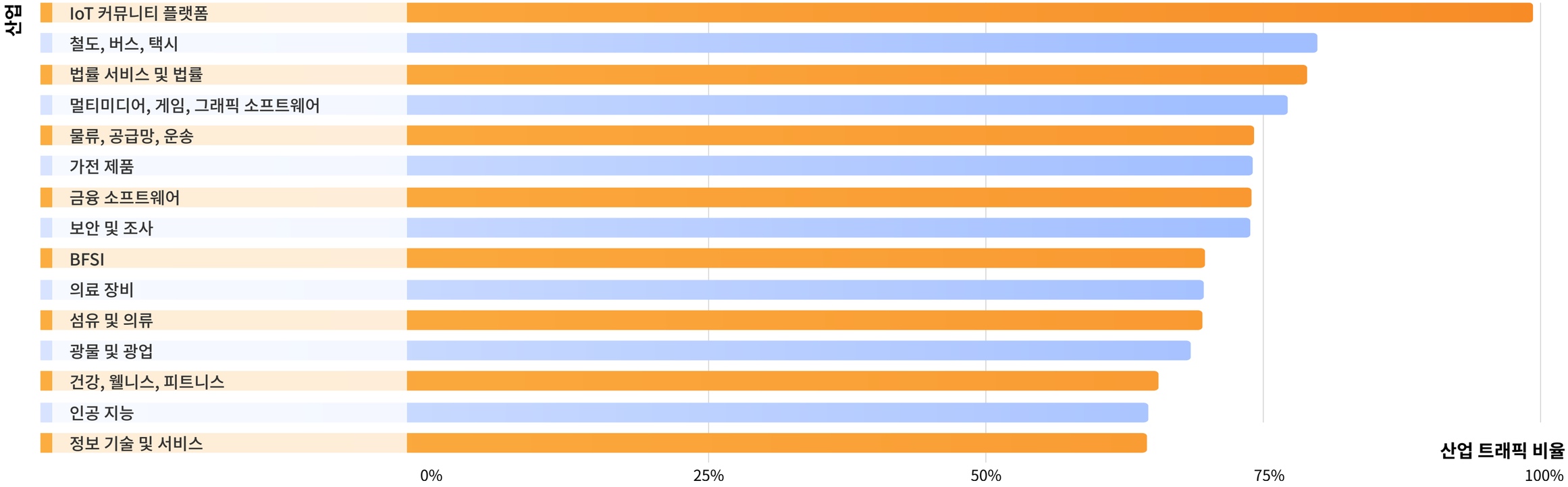
사람들이 오프라인일 가능성이 더 높은 시기, 예를 들어 **12월 25일(크리스마스), 4월 9일(부활절), 4월 22일(이드 알피트르)과 동시에 API 트래픽은 눈에 띄게 줄어듭니다.**[11]



산업 전반의 API 트래픽

지리적 차이 이외에도 **특정 산업은 다른 산업에 비해 API 트래픽 비율이 더 높았습니다.**

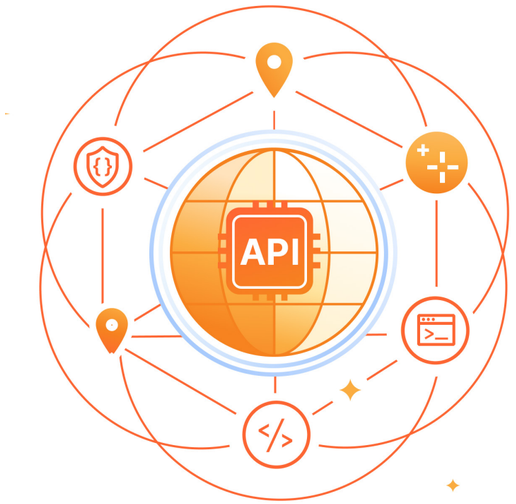
Cloudflare에서 관찰한 결과 API 기반 트래픽 볼륨이 더 높았던(산업 전체 동적 HTTP 트래픽과 비교)[12]
상위 15개 산업은 다음과 같았습니다.



산업 벤치마크

애플리케이션, 웹 사이트, 모바일 앱에서는 처음부터 새로운 기능을 마련하는 대신 API를 통해 기능을 추가하여 사용자 경험을 풍부하게 만들 수 있습니다.

예를 들어 차량 공유 앱에서는 처음부터 자체 결제 서비스를 마련하는 대신 결제 회사의 API를 통해 결제를 추가할 수 있습니다. 소매 API는 가상 피팅룸, 제품 추천, 주문 상태를 통해 고객 경험을 맞춤화하는 데 유용합니다.



API는 모든 산업, 모든 곳에서 유용합니다.

특정 지역에서 API 트래픽 비율이 가장 높은 산업[12]:

아프리카

1. **시설 서비스**
2. 광물 및 광업
3. 자본 시장
4. 모금
5. 신용 카드 및 거래 처리

아시아

1. **IoT 커뮤니티 플랫폼**
2. 광물 및 광업
3. 섬유 및 의류
4. 은행, 금융 서비스, 보험
5. 인공 지능

유럽

1. **멀티미디어, 게임, 그래픽 소프트웨어**
2. 콘텐츠 및 협업 소프트웨어
3. 의료 장비
4. 섬유 및 의류
5. 법률 서비스

남미

1. **광물 및 광업**
2. 금융 소프트웨어
3. 멀티미디어, 게임, 그래픽 소프트웨어
4. 자본 시장
5. 법률 업무

중동

1. **모금**
2. 법률 서비스
3. 무선
4. 자본 시장
5. 운송/트럭/철도

중동

1. **법률 서비스**
2. 철도, 버스, 택시
3. 소비자 가전
4. 보안 및 조사
5. 물류, 공급망, 운송

북미

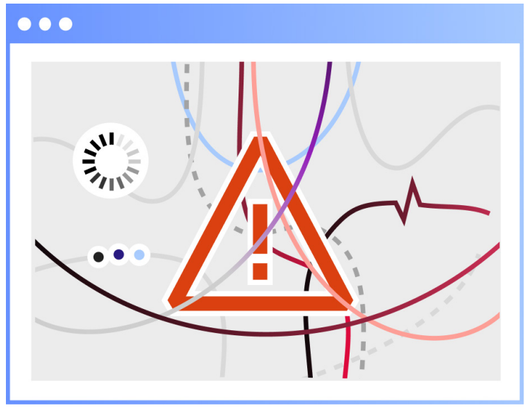
1. **광물 및 광업**
2. 섬유 및 의류
3. 자본 시장
4. 보안 및 조사
5. 제약 및 생명 공학, 건강

2024년 및 그 이후의 예측

소비자와 최종 사용자는 더욱 빠르고 동적인 웹 경험과 모바일 경험을 기대하므로, 더 많은 API를 배포하고 유지해야 한다는 압박이 개발 및 API 팀에 더욱 강해질 것입니다. 이러한 선의의 앱 개발자는 계속 API를 신속하게 배포할 텐데, 다른 IT 및 보안 이해관계자와 상의하지 않는 경우도 있습니다.

이와 같이 응집력 있게 접근하지 못하는 기업은 다음의 어려움을 겪으며 까다로운 상황에 처하게 됩니다.

1 제어 능력 상실 및 복잡성 증가



IT 의사 결정자들은 IT 및 보안 환경 제어 능력을 상실하는 데 가장 크게 기여하는 요소는 "전체 애플리케이션 수"이며, 그 다음이 "애플리케이션 위치의 증가"라고 답했습니다.

하지만 대부분의 조직에서 이러한 팀의 사일로 상태는 여전합니다.

73% 보안 팀에서 사용하라고 요구한 작업 또는 도구가 "생산성 및 혁신에 지장을 준다"고 답한 개발자 비율.

87% 소프트웨어 엔지니어 및 개발자가 "새로운 제품 및 서비스를 더 빨리 시장에 출시하기 위해 보안 정책을 양보한다"고 생각하는 CIO 비율.

<50% 개발자가 개발 및 워크플로우 도구의 보안 위험을 "매우 잘 알고 있다"고 느끼는 CISO 비율.

수천 개의 API 지원 자산과 관련된 방대한 공격면을 보호할 책임은 IT, 보안, 앱 개발 팀 모두에 있습니다.

자동화된 API 보호로 IT, 보안, 앱 개발 간의 단절을 해결하지 않는 **기업에서는 API 위험 및 관리 복잡성이 높아질 수 있습니다**.

2 AI에 더 쉽게 접근하며 증가하는 API 위험

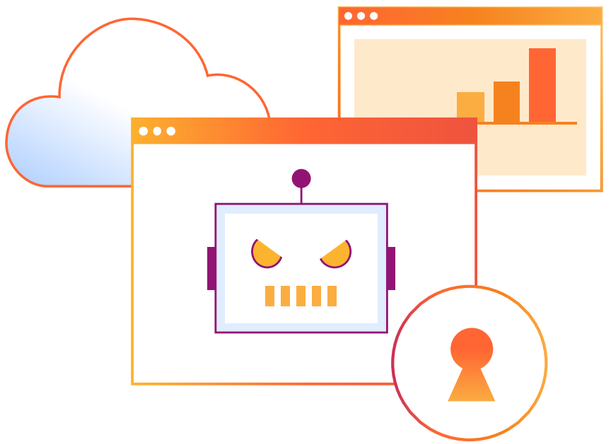
분석가 예측에 따르면 2026년까지 프로덕션 환경에서 ChatGPT와 같은 생성형 인공 지능(GenAI) API 또는 모델을 사용하거나 GenAI 지원 애플리케이션을 배포하게 될 기업은 80%가 넘습니다.

일반적으로 GenAI 모델은(웹 앱 프론트엔드가 없는 경우) 내부 기능으로 직접 액세스하거나 다른 앱 및 사용자가 OpenAI의 ChatGPT, Whisper API와 같은 공개 API를 통해 액세스합니다. **GenAI 사용 시 API 사용이 급격하게 늘어나므로 GenAI 서비스의 API에 대한 API 관련 공격 역시 늘어납니다**.

예를 들어 경쟁사나 공격자가 제품 API를 수백만 번 '호출'하여 데이터를 스크래핑하고 훔치는 경우 피해자의 인프라 요금에 대한 직접적 비용은 상대적으로 미미한 수준일 수 있습니다. 하지만 공격자가 API를 통해 대상 피해자의 생성 모델을 활용하면 호출당 수 센트로, 비용이 훨씬 많이 듭니다. 공격자가 AI 앱의 API로 악성 호출을 수백만 번 실행하면 즉시 금전적 손실이 초래됩니다.

그리고 좋은 의도로 GenAI를 활용하더라도 많은 개발자에게 이는 여전히 미지의(즉, 위험한) 영역입니다. Forrester에서 예측한 바에 따르면, 적절한 보호 조치가 없을 경우 2024년에 "최소 세 건의 데이터 유출 사건에서, 생성된 코드 자체의 보안 결함이나 AI 제안 종속성의 취약점으로 인해 안전하지 않은 AI 생성 코드가 공개적으로 비판을 받을 것"입니다.

3 비즈니스 로직 기반 사기 공격 증가



2020년대에 봇 운영자는 복잡한 브라우저 기반 봇을 만들기 위해 계측 버전의 웹 브라우저를 사용하는 웹 앱을 대상으로 삼았습니다. 동시에, 대부분의 최신 앱에서는 계정 생성, 로그인, 양식 작성, 금전 거래 워크플로우 등 사용자 동작을 완료하기 위해 배후에서 API를 사용하고 있습니다.

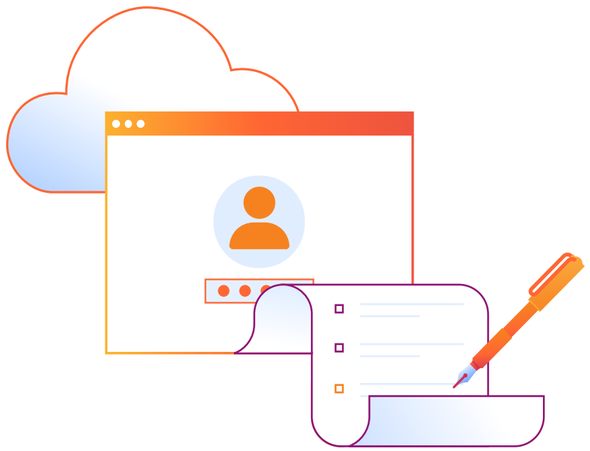
2024년에는 봇 운영자가 해당 워크플로우 배후의 API를 직접 공격하는 경우가 늘어날 것으로 예상됩니다. 이러한 공격이 더 효율적이며(API는 웹 앱 UI보다 더 적게 변경되는 편임) 보호 조치도 적기 때문입니다(웹 앱과 비교할 때).

스포츠 베팅 및 판타지 리그에서 가짜 계정을 생성하는 예시를 들어보겠습니다. 한 사람이 다양한 베팅 및 팀 라인업을 위해 계정을 여러 개 갖고 있으면 승리할 확률이 높아지는데, 승리는 금전적 보상으로 이어지는 경우가 많습니다. 그러므로 신규 계정 생성을 대규모로 자동화하려는 동기가 있으면 수익이 더 커집니다.

자격 증명 스터핑 공격(다단계 인증이 없거나 회피하기 쉬운 경우) 및 공급이 제한된 품목을 부정하게 구매하는 경우에도 이와 유사한 동기가 생깁니다.

이러한 경우, 조직의 API 보안 도구에는 비즈니스 로직에 기반한 인텔리전스가 필요합니다. 예를 들어 공격자가 사기를 빨리 시도하려고 사용한 비정상적인 시퀀스를 구별하는 것입니다. 그리고 API의 기본 트랜잭션 볼륨보다 더 빠르게 트랜잭션을 완료하려고 하는 등, API 호출에 비정상적인 동작 특성이 있는 경우 이를 구별하는 것입니다.

4 규제 및 거버넌스 증가



조직에서는 **API 관련 보안 및 개인정보 보호를 규제하기 위한 더 큰 규모의 거버넌스 및 활동** 역시 예상해야 합니다.

예를 들어 결제 카드 보안 표준(PCI DSS)은 카드 소지자 거래 및 결제 인증 데이터를 관리하는 프로세스에 대해 기업에 지침을 제시하기 위한 프레임워크입니다. 2024년 3월 31일에 **새로운 PCI DSS v4.0 요건(API 보안을 명시적으로 다루는 최초의 버전)이 시행될 예정입니다**.

PCI DSS v4.0이 발표되면 카드 결제를 전송하거나 처리하는 모든 조직에서는 API 취약점을 해결하고 적절한 API 인증을 보장하는 등 여러 조치를 취해야 합니다. PCI DSS 요건을 준수하지 못하면 엄청난 벌금과 기타 처벌이 따를 수 있습니다.

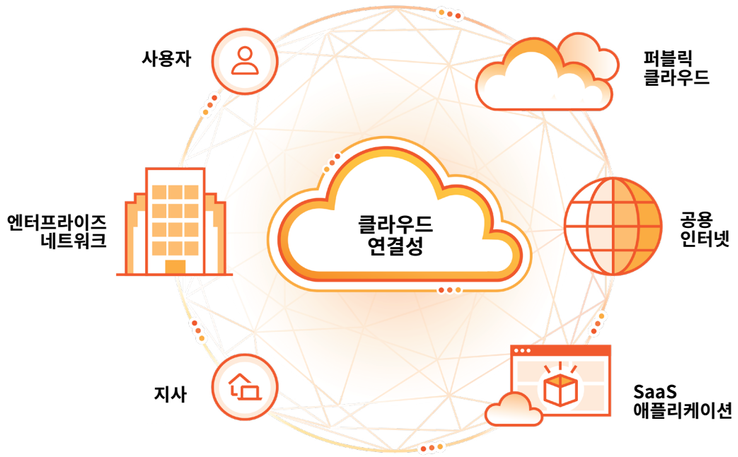
의료 분야는 시스템 간에 전자적으로 보호된 건강 정보(ePHI)를 전송할 수 있는 능력을 고려할 때, API와 관련하여 더욱 철저한 검토가 예상되는 규제가 엄격한 산업 중 하나입니다.

2023년 7월 미국 연방거래위원회 및 보건복지부 시민권 사무국(Office for Civil Rights, OCR)에서는 의료 앱의 개인정보 보호 위험에 대한 검토를 강화하며, 개인 건강 정보 유출을 밝히지 못할 경우 재정적 처벌이 있음을 경고했습니다.

권장 사항

다른 소프트웨어와 마찬가지로 API 취약점도 생겨날 것입니다. 그 누구도 애플리케이션 및 API를 무너뜨리려고 끊임없이 새로운 전술을 시도하는 공격자를 막을 수는 없지만, 조직에서는 다음의 모범 사례를 통합한 총체적인 접근 방식으로 API를 파악, 보호, 관리할 수 있습니다.

1 클라우드 연결성으로 앱 개발, 가시성, 성능, 보안 관리를 통합하세요



전용 인프라, 고유한 규제 준수 요구, 준호환성 프로세스 및 구성으로 인해 SaaS 앱, 웹 앱, 기타 IT 인프라를 연결하기 어려워하는 기업이 많습니다. 간단히 말하자면 이들 도메인은 쉽고 안전하게 함께 작동하도록 구축되지 않았습니다.

클라우드 연결성은 회사에서 디지털 환경을 보호하고 연결하는 데 필요한 여러 서비스를 제공하는 새로운 접근 방식입니다. 프로그래밍 가능한 클라우드 네이티브 서비스를 제공하는 인텔리전트 플랫폼으로, 네트워크, 클라우드 환경, 앱, 사용자 사이에서 무제한 연결이 가능합니다.

클라우드 연결성은 앱 배포 및 API 심층 방어 사이에서 다음과 같은 연결 기능을 제공합니다.

- **자동화된 API 검색 및 가시성**으로 조직에 명확한 API 자산 목록을 제공합니다
- **최신 인증 및 권한 부여** 프로세스를 처음부터 기본으로 제공합니다
- **API 엔드포인트 관리**로 API 기반 도메인의 대기 시간, 오류 및 오류율, 응답 크기 등의 지표를 모니터링할 수 있습니다
- **API** 계층 7(L7) **보호**를 제공하며 고급 레이트 리미팅 및 DDoS 방어가 여기에 포함됩니다(서비스 거부 공격, 무차별 암호 대입 로그인 시도, 기타 API 남용 방어)
- **zero-day**(패치 또는 수정되지 않은 소프트웨어 내 새로운 취약점)를 zero-day 공격 이전에 감지합니다

2 API 게이트웨이로 '적극적 보안' 모델로 나아가세요

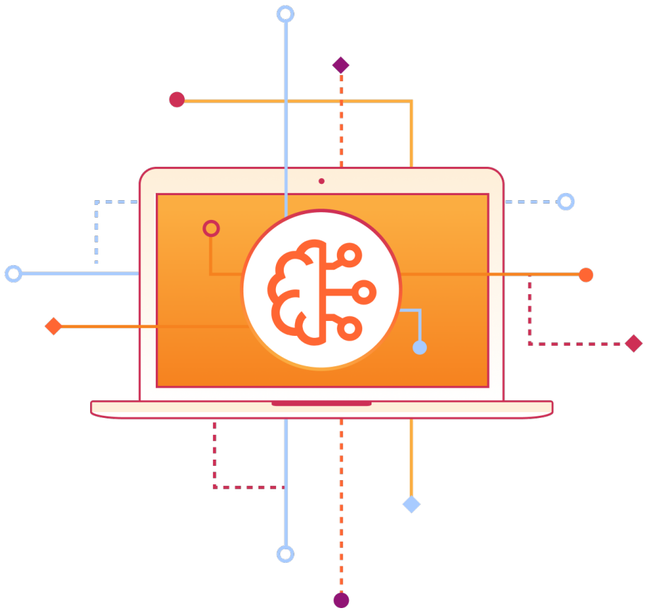
추정에 따르면 2억 개의 공개 및 비공개 API가 사용되고 있고(늘어나고 있음) IT 및 보안 리더는 각 API의 성능, 동작, 위험 노출을 현실적으로 '따라가지' 못하고 있습니다.

전통적으로, 웹 애플리케이션을 보호하는 것은 문제 있는 IP, ASN, 국가의 요청이나 서명에 문제가 있는 요청(SQLi 시도 등)을 제외하면 모든 것을 허용하는 웹 애플리케이션 방화벽(WAF)으로 시행되는 '소극적 보안' 모델이었습니다. 사용자가 다양한 방식으로 웹 앱에 액세스하고 웹 앱과 상호작용할 수 있기 때문입니다. 이 소극적 모델에서 WAF는 다른 모든 트래픽을 허용하면서 "악성이라고 알려진" 트래픽을 차단합니다.

대조적으로 API에는 “적극적 보안” 모델이 더 적절합니다. API에는 상호작용할 수 있도록 구조화된 형식이 있기 때문입니다. 소극적 보안 접근 방식과 반대로 **적극적 보안 모델에서는 “괜찮다고 알려진” 동작 및 ID(API 스키마에서 “괜찮다”고 정의한 것)**만을 허용하고 나머지는 모두 거부합니다.

적극적 보안 모델을 사용하는 조직에서는 스키마에 부합하는 트래픽만 받아서 API를 보호합니다. 그러면 자격 증명 스터핑 공격, 자동화된 스캔 도구 등 잘못된 형식의 요청 및 HTTP 이상을 더 효과적으로 차단할 수 있습니다.

3 머신 러닝 기술을 사용하여 리소스를 확보하고 비용을 줄이세요

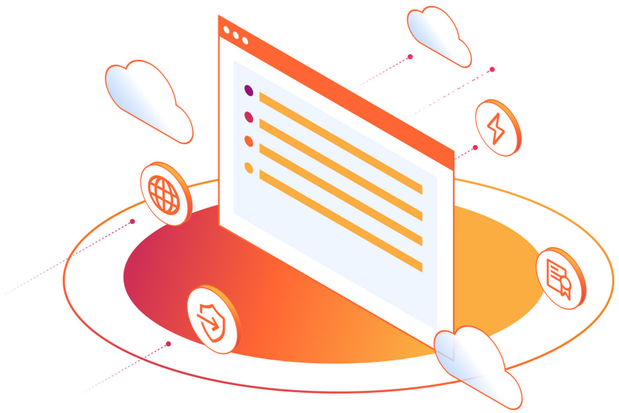


목적에 맞게 구축된 자동 API 도구가 없다면 IT 및 보안 이해관계자가 API 팀을 따라갈 수 없습니다.

하지만 조직에서 머신 러닝 기반 보안 서비스를 사용한다면 API 가시성 및 보안 관리의 효율성을 높일 수 있습니다. 예를 들어, 머신 러닝으로 다음 작업을 더 빠르게 할 수 있습니다.

- 세션 식별자 기반 데이터와 무관하게 도메인에 대한 모든 API 트래픽 **파악**(인증되지 않은 API까지 포함)
- API 상의 RCE, XSS, SQLi 공격에서 공격 변형 **감지**
- 다양한 트래픽 유형 및 API 공격 벡터를 구분하도록 분류기 **학습**
- 정상적인 앱 사용자 트래픽 급증과 잠재적으로 악의적인 봇 트래픽으로 인한 급증 **구별**

4 조직의 API 성숙 수준을 시간에 따라 측정하고 개선하세요



가장 포괄적으로 API를 보호할 수 있는 접근 방식은 총체적인 웹 애플리케이션 및 API 보호(WAAP) 플랫폼을 도입하는 것입니다. 하지만 이제 막 API 노출을 인지하기 시작한 조직에서 하룻밤 사이 이를 실현할 수는 없을 것입니다.

하지만 모든 발전에는 시작점이 있습니다. 조직에서 무엇을 보호해야 하는지 이해했다면 포괄적인 API 관리 및 보안을 향해 나아갈 수 있습니다.

수준 1: 가시성

회사에서는 먼저 섀도우 API를 포함한 모든 API 엔드포인트를 추적하고 공식적으로 관리해야 합니다. 하지만 개발자가 API를 만드는 것만큼 빠르게 API를 찾을 수 없는 조직이 많습니다. 그리고 API를 찾는다 해도 잠재적으로 수백 개에 달하는 API 엔드포인트 각각에 고유한 스키마를 정확하게 구축하기란 어렵습니다.

조직에서 API 가시성 서비스를 사용하면 API 엔드포인트를 자동으로 탐색할 수 있으며 API를 누가 소유했는지, 해당 API를 어떻게 사용하는지 식별할 수 있습니다.

수준 2: 일반 웹 공격 보호

웹 애플리케이션과 API는 함께 작동하는 경우가 많습니다(예: 결제를 처리하기 위해 API를 사용하는 전자 상거래 웹 사이트). 하지만 인터넷의 글로벌 특성으로 인해 웹 사이트와 기타 애플리케이션은 여러 위치에서 다양한 수준의 규모와 복잡성을 가진 공격에 노출됩니다.

다음은 웹 애플리케이션과 그 이면의 API를 Dos 및 DDoS 공격, 자격 증명 스터핑, zero-day 취약점, 기타 위협 유형에서 직접 보호하는 '필수' 서비스(여기에서 더 자세히 다룸)의 예시입니다.

- **DDoS 완화 서비스**는 서버와 공용 인터넷 사이에 위치하여 급증하는 악성 트래픽으로 서버가 압도되지 않도록 합니다
- **웹 애플리케이션 방화벽(WAF)**은 웹 애플리케이션 취약점을 이용하는 것으로 알려진(또는 그렇게 의심되는) 트래픽을 필터링합니다
- **암호화 인증 관리**는 SSL/TLS 암호화 프로세스의 핵심 요소를 관리하는 데 유용합니다
- **고급 레이트 리미팅**은 정상적인 사용자에게는 불이익을 주지 않으면서 DoS 공격, 무차별 암호 대입 로그인 시도, 기타 API 트래픽 급증으로부터 엔드포인트를 보호합니다.

수준 3 — API 특정 공격 보호

WAF, DDoS 등의 도구는 웹 보안과 (인간) 앱 사용자의 경험에 중요하지만, 이들 서비스는 구체적으로는 API가 아니라 애플리케이션을 보호하기 위해 고안되었습니다.

조직에서는 API를 통해 서비스가 더욱 많이 노출되므로, 목적에 맞게 구축된 API 보안 및 관리를 통해 웹 앱 보안을 강화해야 합니다.

비지도 머신 러닝을 사용하는 고급 API 보안은 각각의 API마다 별도의 기준을 개발하고 API 요청이 생긴 의도(정상적인 의도인지, 악의적인 의도인지)를 예측할 수 있습니다.

조직에서는 팀의 많은 구성원에게 API 보안이 새로운 내용이라는 점을 잘 알고 있습니다. 보안은 그 자체를 위해서가 아니라, 비즈니스 결과를 개선하고 높이려는 목적으로 달성해야 합니다. 제품 제공이 빨라지고, 게시된 API의 보안 허점이 줄어들며, 보안 팀의 효율이 높아지고, 궁극적으로는 개발자 및 API 팀의 생산성이 높아진다는 것이 보안 달성의 이점입니다.

비즈니스를 구동하는 API 보호

Cloudflare 클라우드 연결성을 기반으로 한 Cloudflare 웹 애플리케이션 및 API 보호(WAAP) 포트폴리오에는 애플리케이션 및 API의 보안과 생산성을 유지하고 DDoS 공격을 저지하며 봇을 차단하는 등 선도적인 애플리케이션 보안 기능이 통합되어 있습니다.

자세히 알아보기

Cloudflare API 검색, OWASP API 상위 10가지 보호, mutual TLS, 혁신을 저해하지 않는 API 보호 등에 대해 알아보세요.

목차

API 보안 용어

API 호출 또는 API 요청: 서버로 전송되는 메시지로 서비스 또는 정보를 제공하도록 API에 요청합니다.

API 검색: API 검색은 조직 내에서 사용되는 모든 내부 및 타사 API를 카탈로그화하는 프로세스입니다.

API 엔드포인트: API 요청(API 호출이라고도 함)이 이루어지는 장소입니다. API 엔드포인트는 거의 항상 서버에서 호스팅됩니다.

API 트래픽: 응답 콘텐츠 유형이 XML, JSON, gRPC 등인 모든 HTTP 요청입니다. 완화된 요청과 같이 응답 콘텐츠 유형을 알 수 없는 경우 대신 그에 준하는 Accept 콘텐츠 유형(사용자 에이전트가 지정)을 사용합니다. 후자의 경우, API 트래픽이 완전히 반영되지는 않지만, 인사이트 목적으로는 대표성이 있습니다.

봇 트래픽/자동화된 트래픽: Cloudflare의 봇 관리 시스템에서 봇이 생성한 것으로 확인한 모든 HTTP 요청입니다.

취약한 개체 수준 권한 부여(BOLA): BOLA는 요청 내의 개체 ID를 조작해 중요한 데이터에 무단 액세스 권한을 얻는 것입니다. 공격자는 BOLA를 이용해 ID를 바꾸는 것만으로도 액세스 권한이 없는 개체(데이터)에 액세스할 수 있습니다.

취약한 사용자 인증: 인증이 잘못 구현된 경우, 공격자는 API 사용자를 가장하여 기밀 데이터에 액세스할 수 있습니다.

클라이언트: HTTP 요청을 수행한 당사자입니다. 보통 브라우저에서 사이트에 액세스하는 최종 사용자에 해당하지만, API 클라이언트 또는 사이트에서 리소스를 요청하는 모든 당사자도 클라이언트에 해당할 수 있습니다.

디렉터리 순회: 경로 순회 공격이라고도 하는 디렉터리 순회의 목표는 웹 루트 폴더 외부에 저장된 파일 및 디렉터리에 액세스하는 것입니다.

분산 서비스 거부(DDoS) 공격: DDoS 공격은 공격 대상이나 주변 인프라를 인터넷 트래픽의 폭주로 압도하여 표적 서버, 서비스, 네트워크의 정상적인 트래픽을 방해하려는 악의적인 시도입니다.

파일 삽입: 공격자는 이 취약점을 이용하여 대상 애플리케이션에 파일을 삽입할 수 있습니다. 적절한 유효성 검사 없이 사용자가 제공한 입력을 사용할 때 이 취약점이 발생합니다.

HTTP 이상: 잘못된 형식의 메서드 이름, 헤더의 null 바이트 문자, 비표준 포트, POST 요청이 있는 콘텐츠에서 길이가 0인 경우 등의 HTTP 이상은 Cloudflare의 관리형 WAF 규칙으로 완화되는 공격을 일반적으로 나타내는 지표입니다. 예시 HTTP 이상 규칙에 대한 자세한 설명은 이곳 Cloudflare 블로그에서 확인할 수 있습니다.

삽입 공격 유형 예시:

- **명령 삽입**: 공격자가 취약한 애플리케이션을 통해 호스트 운영 시스템에서 임의의 명령을 실행하는 경우입니다.
- **교차 사이트 스크립팅(XSS):** XSS는 취약점으로, 이를 통해 공격자는 클라이언트 측 스크립트를 웹 앱에 삽입하여 중요한 정보에 직접 액세스하거나 사용자를 가장하거나 사용자가 중요한 정보를 공개하도록 유도할 수 있습니다.
- **SQL 삽입(SQLi):** 데이터베이스가 검색 쿼리를 실행하는 방식의 취약점을 공격자가 악용할 수 있는 방법입니다. 공격자는 SQLi를 이용하여 비인가 정보에 액세스하거나, 새 사용자 권한을 수정하거나 생성하거나, 기타 방식으로 중요한 데이터를 조작하거나 파괴합니다.

HTTP 요청: 웹 브라우저 및 앱과 같은 인터넷 통신 플랫폼에서 리소스를 로드하는 데 필요한 정보를 요청하는 방법입니다.

완화된 트래픽: Cloudflare 플랫폼에서 적용한 "종료" 동작이 있는 대략적 HTTP* 요청. 여기에는 BLOCK, CHALLENGE(예: 캡차, JavaScript 기반 질문) 등이 포함됩니다. LOG, SKIP, ALLOW 동작이 적용된 요청은 포함되지 않습니다.

레이트 리미팅: 요청 처리 속도를 제어하기 위해 컴퓨터 시스템에서 사용되는 기법입니다. API 공격을 방지하거나 원본 서버의 리소스 사용을 제한하기 위한 보안 조치로 사용할 수 있습니다.

원격 코드 실행(RCE): 공격자가 조직의 컴퓨터나 네트워크에서 악성 코드를 실행하는 공격입니다. 공격자가 제어하는 코드를 실행하는 기능은 추가 맬웨어를 배포하거나 중요한 데이터를 탈취하는 등 다양한 목적으로 사용될 수 있습니다.

스키마 유효성 검사: API 요청이 API 스키마를 준수하지 않는 경우, 예를 들면 기밀 데이터를 노출하는 등 API가 예상과 다른 방식으로 반응할 수 있습니다. 스키마 유효성 검사를 통해 API는 이러한 요청을 삭제할 수 있습니다.

zero-day 취약점: 애플리케이션 제작자에게 알려지지 않은 취약점이며, 따라서 해결 방법이 없습니다. 공격자는 최대한 빠르게 이러한 취약점을 악용하려고 합니다.

HTTP 상태 코드 설명

아래의 상태 코드 예시(섹션 8에서 설명한 가장 흔한 API 오류)에서는 Cloudflare에서 HTTP 응답 코드에 대한 인터넷 표준 추적 프로토콜을 어떻게 해석하는지 자세히 설명합니다. 표준화 상태와 이 프로토콜의 상태는 최신 버전의 '인터넷 공식 프로토콜 표준(STD 1)'을 참조하십시오.

429는 **요청이 너무 많다**는 의미입니다. 서버에 따르면 지정된 시간 내에 클라이언트에서 너무 많은 요청이 전송된 것입니다. 흔히 "레이트 리미팅"이라고 합니다. 서버에서는 요청자가 지정된 시간 이후에 다시 시도할 수 있다는 정보로 응답할 수도 있습니다.

400은 **잘못된 요청**을 의미합니다. 클라이언트에서 서버에 올바른 요청을 전송하지 않은 것입니다. 이는 잘못된 형식의 요청 구문, 유효하지 않은 요청, 메시지 프레임, 사기성 요청 라우팅 등의 클라이언트 오류입니다.

404는 **찾을 수 없다**는 의미입니다. 원본 서버에서 요청 리소스를 찾을 수 없거나 찾으려 하지 않는 것입니다. 일반적으로 호스트 서버가 API URL을 인식할 수 없다는 의미이며 발생 이유는 다양할 수 있습니다.

401은 **권한이 없다**는 의미입니다. 사용자 자격 증명이 존재하지 않거나, 요청 리소스에 적절한 액세스 수준이 없는 것입니다.

403은 **금지되었다**는 의미입니다. 오렌지 구름 형태의 모든 Cloudflare 도메인에 활성화된 기본 WAF 관리형 규칙이나 특정 영역에 활성화된 WAF 관리형 규칙이 요청에서 위반된 경우, Cloudflare에서는 403 응답을 제공합니다. Cloudflare 브랜드 없이 403 오류가 나타났다면 항상 Cloudflare가 아닌 원본 웹 서버에서 직접 반환된 것이며, 일반적으로 사용자 서버의 권한 규칙과 관련됩니다.

500은 **내부 서버 오류**를 의미합니다. 서버 측의 예기치 않은 오류를 나타내는 일반 오류 메시지입니다.

422는 **처리할 수 없는 콘텐츠**를 의미합니다. 요청에 의미 오류가 있는 경우입니다.

503은 **서비스를 이용할 수 없다**는 의미입니다. 서버는 유지보수로 인해 다운되거나 원본 웹 서버 과부하 시 다운될 수 있습니다.

430은 **요청 헤더 필드가 너무 크다**는 의미입니다. 공식 오류 코드는 아니지만, 악의적 요청일 수 있어 요청이 수락되지 않았다는 의미로 Shopify에서 사용됩니다. Shopify는 발생할 수 있는 공격으로부터 앱을 보호하기 위해 이러한 요청을 거부하는 것으로 응답합니다.

402는 **결제가 필요하다**는 의미입니다. 널리 사용되지는 않지만, 일일 한도가 초과되었거나 결제에 문제가 있을 때 일부 플랫폼에서 사용됩니다.

미주

1. Cloudflare의 전역 네트워크에서는 초당 평균 5,000만 건의 HTTP 요청이 처리되며, 최고일 때는 초당 7,000만 건 이상의 HTTP 요청이 처리되기도 합니다. 2022년 10월 1일부터 2023년 8월 31일 사이에 성공적으로 응답한(상태 코드 200) API 트래픽은 Cloudflare의 동적 HTTP 트래픽 중 53.1%~60.1%를 차지했습니다. 동적 콘텐츠는 방문 시간, 위치, 장치 등 사용자에 특정한 요인에 따라 변경되는 콘텐츠를 말합니다.
2. REST API 엔드포인트의 경우, 모든 고객의 도메인/영역에서 고객이 제공한 세션 식별자로 발견한 것보다 Cloudflare API 검색이 머신 러닝을 이용하여 발견한 엔드포인트가 계정당 중앙값 30.7%만큼 더 많았습니다(260 대 199).
3. 2022년 10월 1일부터 2023년 8월 31일까지, API에 대한 모든 HTTP 오류(동적 캐시 상태) 중 가장 일반적인 2xx 이외의 HTTP 상태 코드(4xx 및 5xx 오류 포함) 백분율을 바탕으로 합니다.
4. Cloudflare에서는 완화된 API 트래픽을 계산하기 위해 Cloudflare 제품 소스별 일일 API 트래픽 완화 비율과 더불어 웹 애플리케이션 방화벽(WAF) 규칙 범주별로 관리형 규칙을 통해 완화된 일일 트래픽 비율을 계산했습니다.
5. 전체 산업의 동적 HTTP 트래픽 중 API 트래픽이 70%를 초과하는 상위 산업(조직의 Salesforce 산업 범주별)입니다.
6. Cloudflare 네트워크에서 처리된 모든 동적 HTTP 트래픽 중 성공적인(200) 응답 코드를 반환한 북미, 유럽, 라틴 아메리카, 오세아니아, 아시아, 아프리카, 중동 지역 API 트래픽의 비율을 바탕으로 합니다.
7. Cloudflare의 API 검색 방법은 두 가지입니다. 세션 식별자가 포함된 트래픽을 살펴보는 방법과 세션 식별자가 필요하지 않은 ML 기반 검색 엔진을 사용하는 방법입니다. ML로만 엔드포인트가 검색된 계정은 15,431개였습니다.
8. 계정별로 집계된 API 수를 바탕으로 하여, 쓰기 액세스(PUT, POST, PATCH, DELETE)가 있는 엔드포인트와 '정보 전용'(GET) 액세스가 있는 엔드포인트로 분류했습니다. 본 보고서의 목적을 위해 Cloudflare에서는 각 고객의 총 API 수에서 최소 50% 이상 GET API로 구성된 계정의 비율을 계산했습니다.
9. Cloudflare WAF 관리형 규칙 범주별로 고객을 위해 완화한 API 트래픽을 바탕으로 합니다
10. (동적 캐시가 있는 전체 200 응답 코드 트래픽 중) 클라이언트 국가 지역 내에서 200 응답 코드 및 동적 캐시를 반환한 API 요청 수로 계산한 일일 API 비율 중앙값을 기준으로 합니다.
11. 전 세계 일일 API 트래픽 기준치(평균)와 비교한 일별 API 트래픽 변화율을 바탕으로 합니다.
12. 다른 산업과 비교한 해당 산업의 동적 HTTP 트래픽 총량을 바탕으로 합니다. 여기에서 '산업'은 고객 계정의 Salesforce 산업 범주로 정의됩니다.

전화번호: 007-9814-2030-192
이메일: enterprise@cloudflare.com
방문: www.cloudflare.com/ko-kr/

REV:BDES-5122.2024JAN04