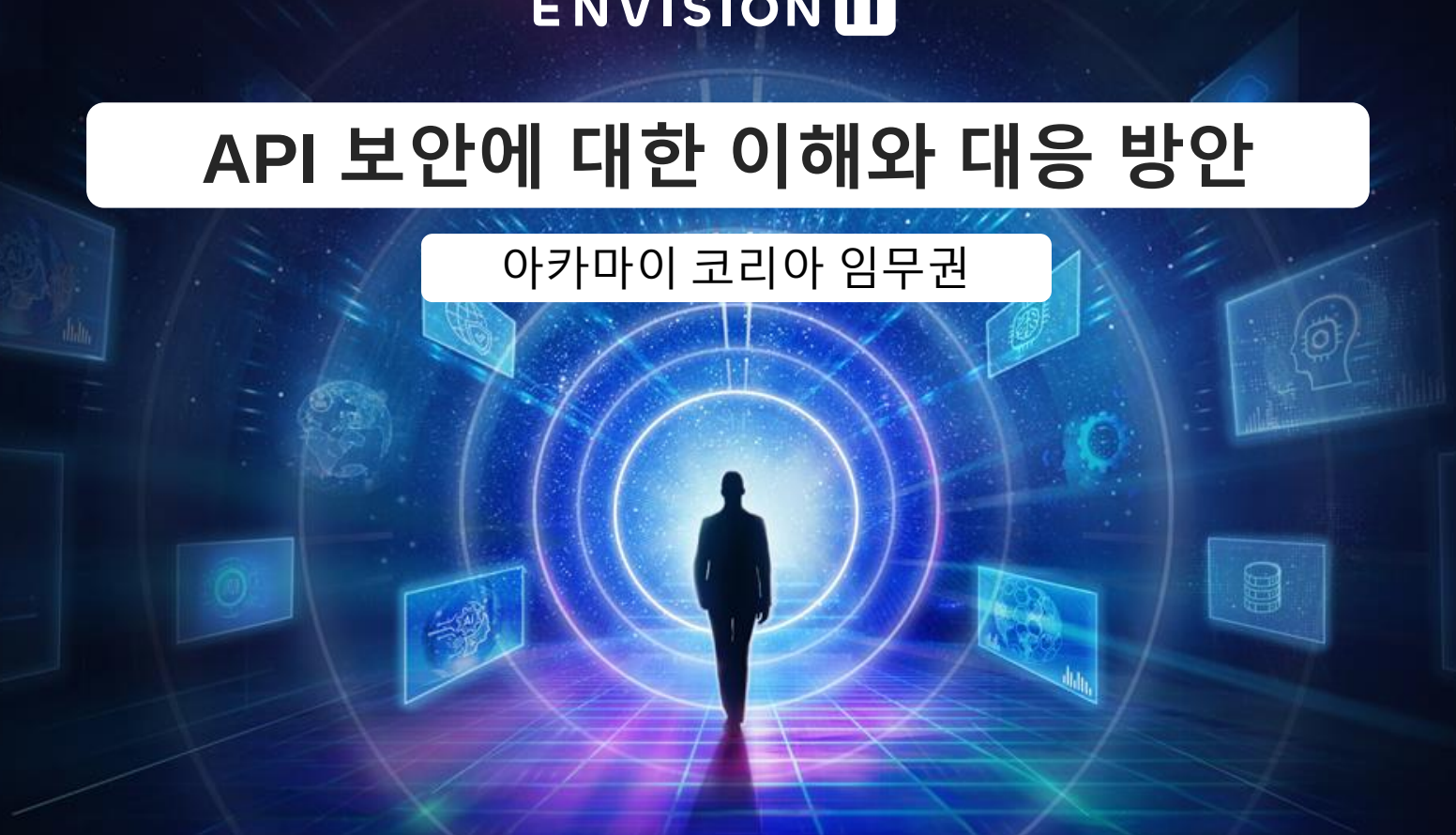
CLOUDSEC 2023

ENVISION IT

API 보안에 대한 이해와 대응 방안

아카마이 코리아 임무권



Hosted by

TREND MICRO™

API 알아 보기

API란 무엇인가요?

- 프로그래밍 가능한 **인터페이스**
- 애플리케이션간 통신을 위한 **프로그래밍 집합**
- 요청을 보내는 App를 **클라이언트**, 응답을 보내는 App을 **서버**

API(Application Programming Interface)란?

> *시스템에 공개적으로 노출된 하나이상의 엔드 포인트로 구성된 프로그래밍 인터페이스*

API

- **JSON, XML 등으로 표현**
- **HTTP 기반 웹을 통해 노출**
- **엔드 포인트**
  - 리소스 경로(URI)
  - 수행 작업(읽기, 삭제, 생성)
  - 데이터 정의(JSON, XML 등)

- **대표적 웹 API**
  - RESTful API
  - SOAP API
  - GraphQL API
  - gRPC API
  - OpenAPI

- **사용 용도 별 API**
  - 공개 / 외부 API
  - 비공개 / 내부 API
  - 타사 / 파트너 API
  - 인증 / 비 인증 API

API에 대한 활용

API 확산

API 서비스 소비량

출처: Smarbear 소프트웨어 품질 현황

중요 요소

API 신뢰성, 보안 및 안정성이 가장 중요

개발자 사고 방식

노력의 대부분이 API 작업에 투입

출처: Postman 2022 API 현황

API만의 고유한 고민들

Security

90%

웹 애플리케이션 공격이 API와 관련되어 있으며, 가장 일반적인 공격 벡터

- 애플리케이션 오리진 공격
- 컨텐츠 갈취 / 사용자 데이터 노출

Performance

<50ms

조직의 90%가 예상하는 응답시간

- 사용자 이탈
- 매출 손실
- 고객 손실 / 충성도 하락

Reliability

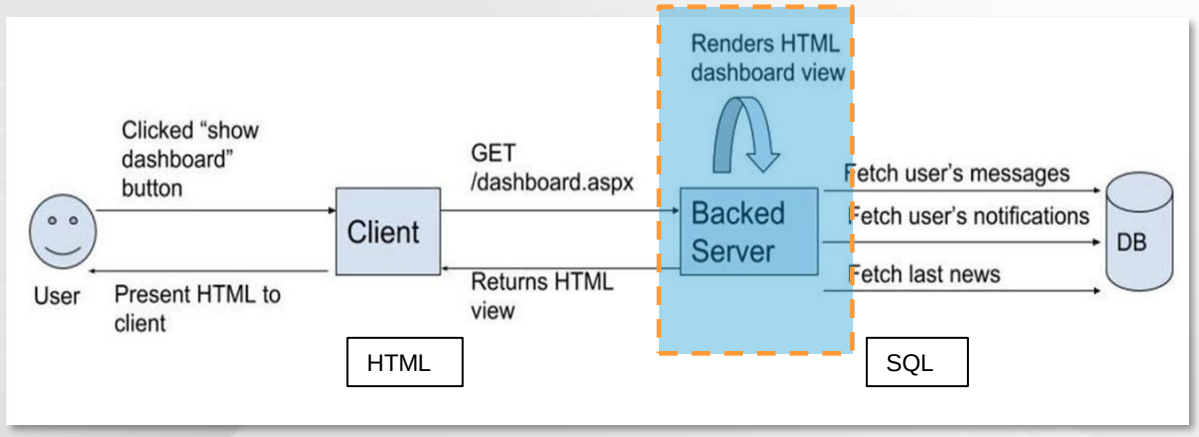
77%

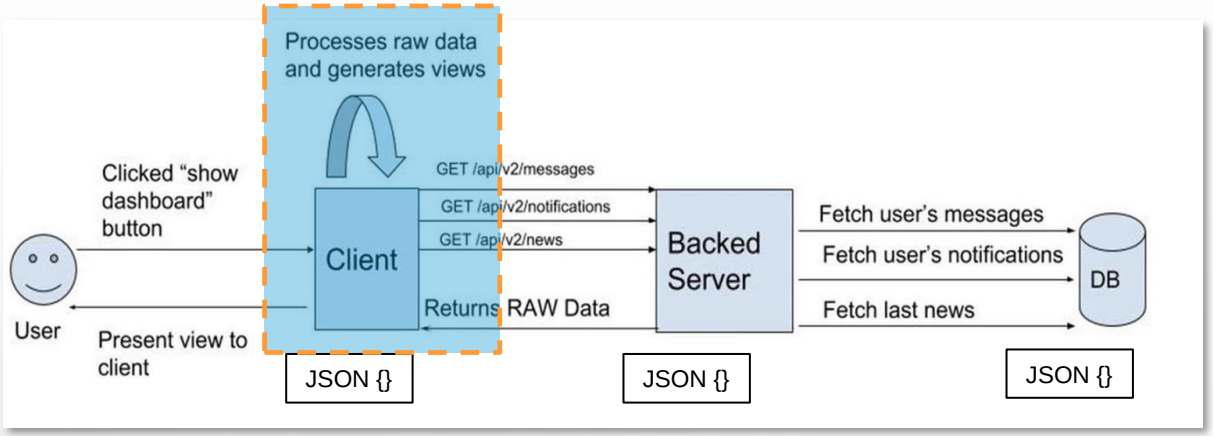
API 안정성을 최우선 과제로 취급

- 수요에 맞춰 애플리케이션 확장 어려움
- 서비스 가용성의 불일치

Traditional vs Modern Application

> *기존 앱과 최신 앱의 주요 차이점 중 하나는* ***뷰가 렌더링 되는 위치***





- 백엔드 서버가 대부분의 뷰 생성을 담당
- 클라이언트-서버는 HTML
- 데이터 소스는 데이터 쿼리가 있는 DB이고 출력은 SQL

- 뷰는 일반적으로 **클라이언트 자체에서 렌더링**
- 데이터 소스는 앱 서버이고 **데이터 쿼리는 API 호출**
- 데이터는 **모두 JSON**
- 클라이언트에서 DB까지 공유 언어 사용

API를 활용하는 사례

Interactive Apps (social, Weather, etc,.)

Gambling / Auction Apps

OTT Video workflows / application

Telemetry data

Digital Wallets / Payment

Console Game

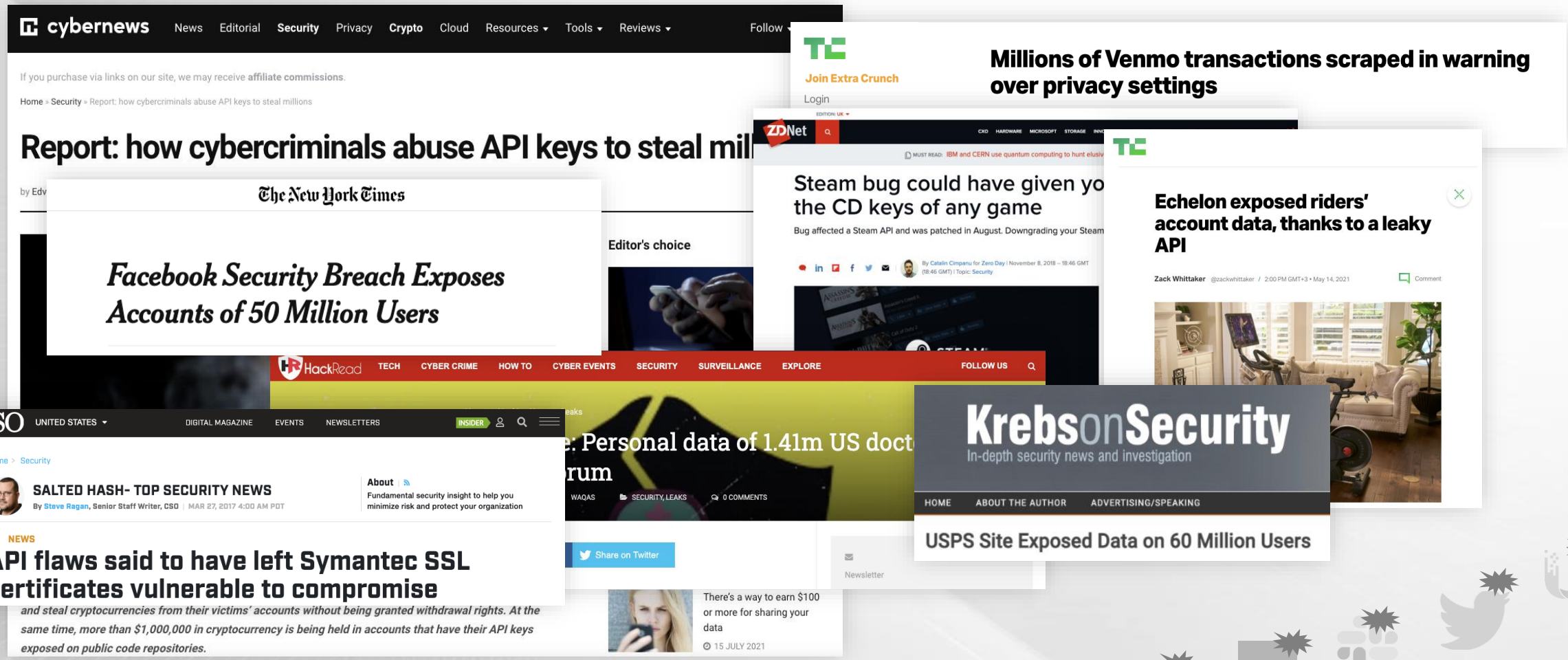
Ad Tech

Location Services / Mapping

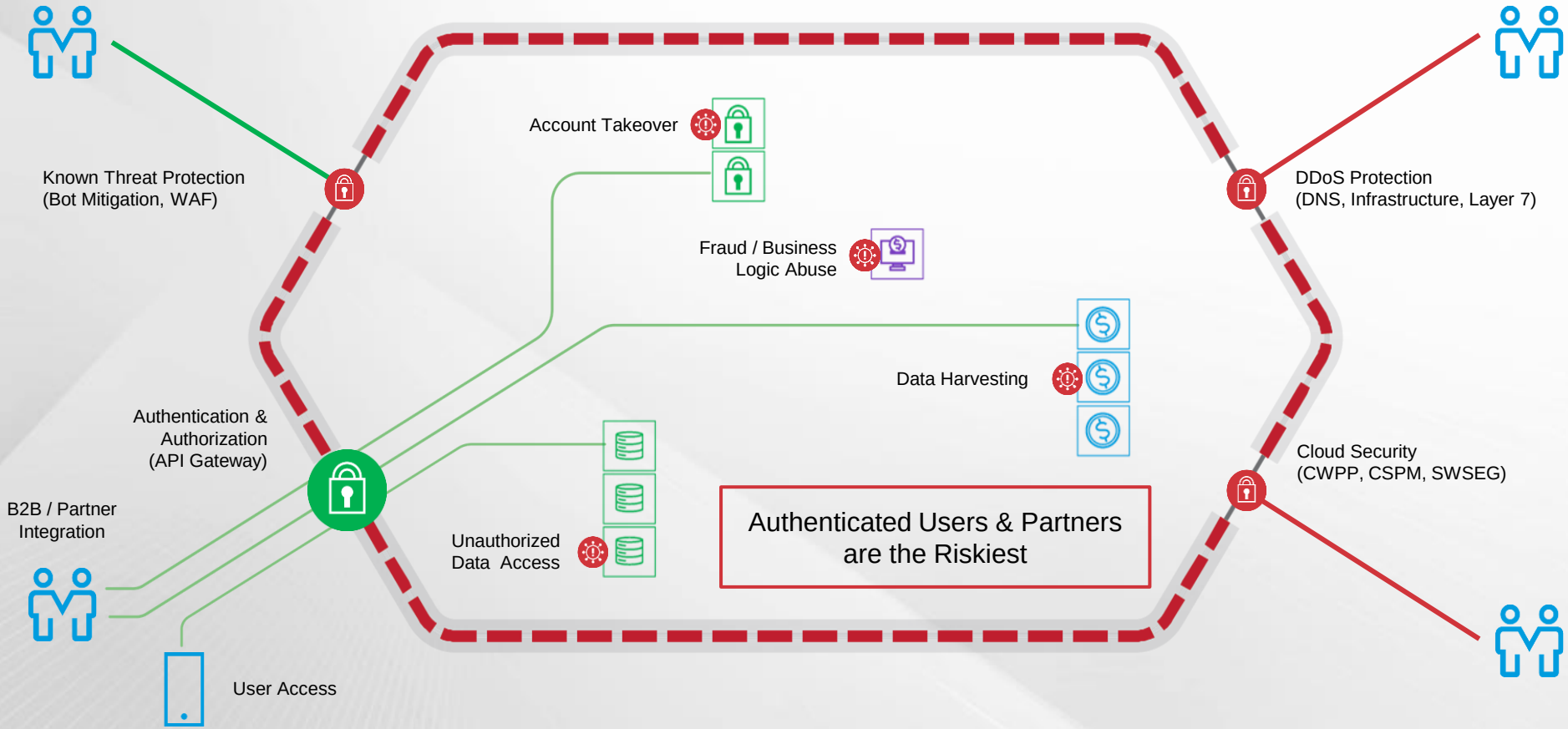
“By 2022, APIs will become the #1 attack vector.”

Gartner, How to Build an Effective API Security Strategy

데이터 유출을 초래하는 API 위협



API 공격 표면

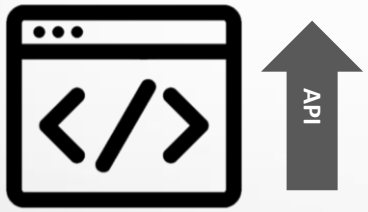

Known Threat Protection
(Bot Mitigation, WAF)
Account Takeover
Fraud / Business
Logic Abuse
Data Harvesting
Authentication &
Authorization
(API Gateway)
B2B / Partner
Integration
Unauthorized
Data Access
Authenticated Users & Partners
are the Riskiest
DDoS Protection
(DNS, Infrastructure, Layer 7)
Cloud Security
(CWPP, CSPM, SWSEG)
User Access

API Security?

증가되는 API와 잠재 위협

“ *API 설계 특성 때문에 **합법적인 사용과 공격 및 악용을 구분하기 어려움***

Microservice Architecture



Front-end Application

IoT & Business Automation

By 2022, API abuses will move from an infrequent to the most frequent attack vector, resulting in data breaches for enterprise web applications. - Gartner -

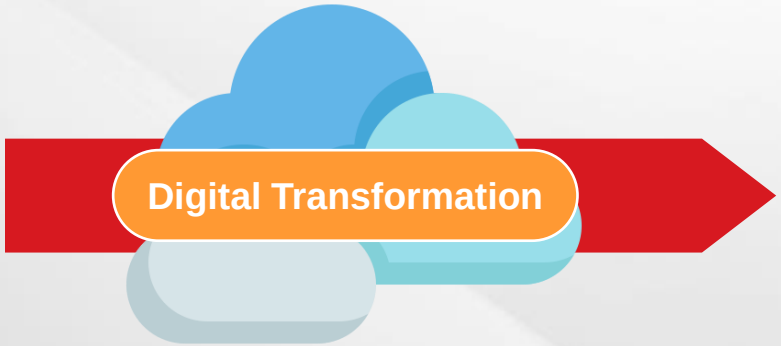
API 공격 변화

> *복잡하고 빠르게 진화하는 위협 환경에 대응하기 위한 **정교한 보안 제어 필요***
>
> *API 보안에 영향을 미치는 소프트웨어 개발 및 문서화 같은 **'비 보안'** 영역까지 확장*

Yesterday's Attacks

목표
– 데이터 센터에 중요 시스템을 식별

방법
- 네트워크 침투 후 측면 이동



Today's Attacks

목표
– 노출된 API에 비즈니스 로직과 데이터 식별

방법
- 설계상 API를 통해 노출된 데이터 및 트랜잭션
- API 키와 자격증명을 손상시킴

인적 오류를 조심해야 합니다.

DevOps 역시 고민이 있습니다.

Dev Team

Sec Team

OWASP API Top 10

OWASP API Top 10, 2019

1	손상된 오브젝트 수준 권한(BOLA)
2	취약한 인증
3	과도한 데이터 노출
4	리소스 부족 및 속도 제한
5	손상된 기능 수준의 권한 부여(BFLA)
6	대량 할당
7	잘못된 보안 설정
8	인젝션
9	부적절한 자산관리
10	충분하지 않은 로깅 및 모니터링

OWASP API Top 10, 2023

손상된 오브젝트 수준 권한(BOLA)	1
취약한 인증	2
손상된 오브젝트 프로퍼티 수준 권한	3
무제한 리소스사용	4
손상된 기능 수준의 권한 부여(BFLA)	5
서버 측 요청 위조	6
잘못된 보안 설정	7
자동화된 위협 방어 기능 부족	8
부적절한 자산관리	9
안전하지 않은 API 사용	10

OWASP API ToP 10 2023 - BOLA

- 잘못된 addDriver API 호출
- 사용자의 UUID를 유출
- 호출을 반복하여 사용자의 UUID 수집

1

```
POST /marketplace/\_rpc?rpc=getConsentScreenDetails HTTP/1.1
Host: [redacted].com
Connection: close
Content-Length: 67
Accept: application/json
Origin: [https://[redacted]com](https://[redacted]com)
x-csrf-token: xxxx
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_3) AppleW
DNT: 1
Content-Type: application/json
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: xxxxx
{"language":"en","userUuid":"xxxx-776-4xxxx1bd-861a-837xxx604ce"}
```

- UUID로 getConsentScreenDetails 호출
- 모든 사용자 정보를 유출
- 계정을 탈취를 위한 정보 획득

2


```
{
    "status":"success",
    "data":{
        "data":{
            "language":"en",
            "userUuid":"xxxxxx1e"
        },
        "getUser":{
            "uuid":"cxxxxxc5f7371e",
            "firstname":"Maxxxx",
            "lastname":"XXXX",
            "role":"PARTNER",
            "languageId":1,
            "countryId":77,
            "mobile":null,
            "mobileToken":1234,
            "mobileCountryId":77,
            "mobileCountryCode":"+91",
            "hasAmbiguousMobileCountry":false,
            "lastConfirmedMobileCountryId":77,
            "email":"xxxx@gmail.com",
            "emailToken":"xxxxxxxx",
```


손상된 오브젝트 수준 권한 (BOLA) 1

취약한 인증 2

손상된 오브젝트 프로퍼티 수준 권한 3

무제한 리소스사용 4

손상된 기능 레벨 권한 부여 (BFLA) 5

서버 측 요청 위조 6

잘못된 보안 설정 7

자동화된 위협을 방어할 기능 부족 8

부적절한 자산 관리 9

안전하지 않은 API 사용 10

OWASP API ToP 10 2023 - BFLA

DELETE
/api/v1.1/user/12358/posts?id=32

POST
/api/v1.1/user/12358/posts?id=32

"Launched my new startup today…yeahh" #sucess #apisec

1. 손상된 오브젝트 수준 권한 (BOLA)
2. 취약한 인증
3. 손상된 오브젝트 프로퍼티 수준 권한
4. 무제한 리소스사용
5. **손상된 기능 레벨 권한 부여(BFLA)**
6. 서버 측 요청 위조
7. 잘못된 보안 설정
8. 자동화된 위협을 방어할 기능 부족
9. 부적절한 자산 관리
10. 안전하지 않은 API 사용

API 보호를 위한 모범 사례

- 소프트웨어 개발 **라이프 사이클과 통합**
- 보안 테스트를 지속적으로 **CI/CD에 통합**
- 인증 및 권한 부여 **통제**
- **속도 제한**
- **DDoS 공격 위험 완화**
- API 보안과 애플리케이션 **테스트 프로세스의 통합**
- API의 **지속적인 검색**
- 일반적인 **API 취약점을 식별 및 조치**
- **시그니처 기반 API 공격**에 대한 보호
- 새로운 위협에 대한 **복원력 증가**
- API 보안 분석이 **History로 확장**
- API 활동에 대한 모니터링 및 알림

OWASP API Top 10, 2023

손상된 오브젝트 수준 권한(BOLA)	1
취약한 인증	2
손상된 오브젝트 프로퍼티 수준 권한	3
무제한 리소스사용	4
손상된 기능 레벨 권한 부여(BFLA)	5
서버 측 요청 위조	6
잘못된 보안 설정	7
자동화된 위협을 방어할 기능 부족	8
부적절한 자산 관리	9
안전하지 않은 API 사용	10

모든 API 탐지가 중요합니다.

“ *모든 API 취약점이 **OWAS API Top 10** 안에서 식별 될까요?*

좀비 API란?

오래된 엔드포인트가 폐기되지 않고
더이상 사용되지 않는 API가 접속 가능
상태로 존재함

쉐도우 API란?

공격자, 레거시, 관리자, 좀비 등
비즈니스 요구와 관련 없는 모든 API

잠재적 위험

취약점 악용

인프라의 기술적
취약점으로 인해 서버
손상 가능
CVE-2017-9791
CVE-2018-11776
CVE-2021-44228

비즈니스 로직 악용

악의적인 공격자가
애플리케이션 설계
또는 구현 결함을
악용하여 승인되지
않은 동작을 유도하는
행위

무단 데이터 접속

API 남용 또는 취약한
인증 메커니즘을
악용하여 접근하는
행위
BOLA, IDOR, BFLA

계정 탈취

인증정보 도난 또는
XSS 공격으로 계정
탈취하여 정상적인
행동으로 간주되어
탐지 어려움

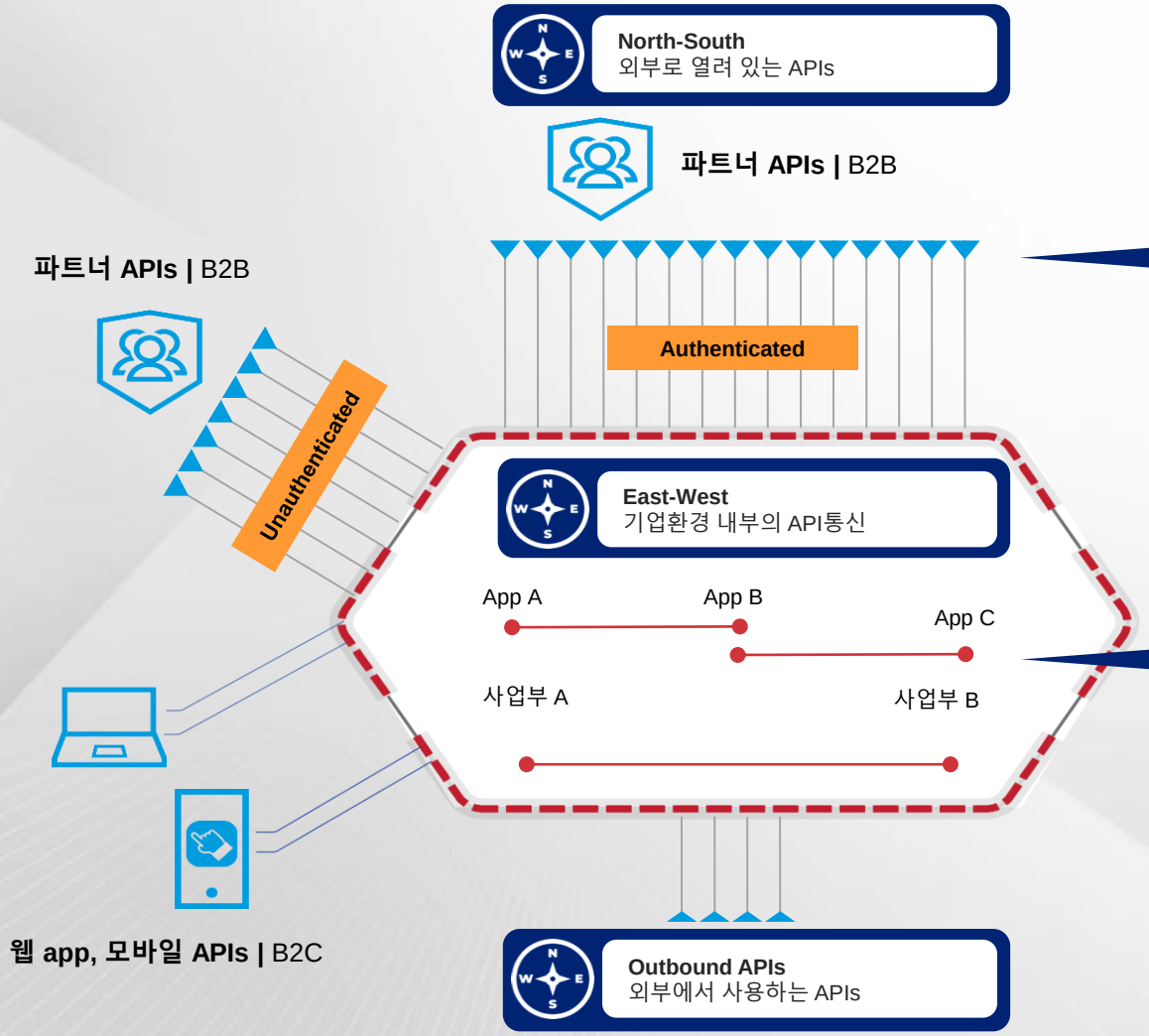
데이터 스크래핑

공개 API를 통해
데이터 제공을
악용하여 이러한
리소스를 공격적으로
쿼리하여 데이터 수집

서비스 거부

백엔드에 과중한
작업을 요청하여
서비스 지연 및 거부를
유발

API 다른 시각으로 보면



North – South API란?

- 외부, 주로 파트너 API
- 은행과 다른 핀테크
- 의료 기관과 보험사
- 호텔과 여행사

빠르게 성장하며 ***가장 큰 공격 표면***

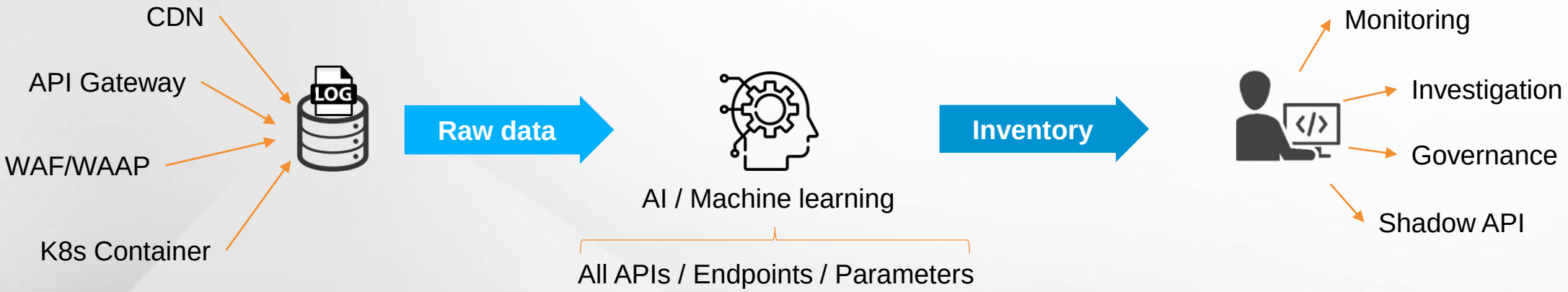
East – West API란?

- 내부 API
- 애플리케이션간 연결
- 사업부서간 연결
- 외부에서 접속할 수 없는 API

안전하다 간주되며 ***가시성 부족***

쉐도우 API는 어떻게 찾을까?

> *가장 광범위한 범위의 소스에서 API 활동 로그를 수집하고 분석*



광범위한 로그 수집

- Data Center(On-prem)
- Cloud(Pub, Pri)

지능화된 분석

- Security Platform
  - Smart Analysis
  - Historical Analysis
  - Faster Deployment
  - Business Context

위험 제거

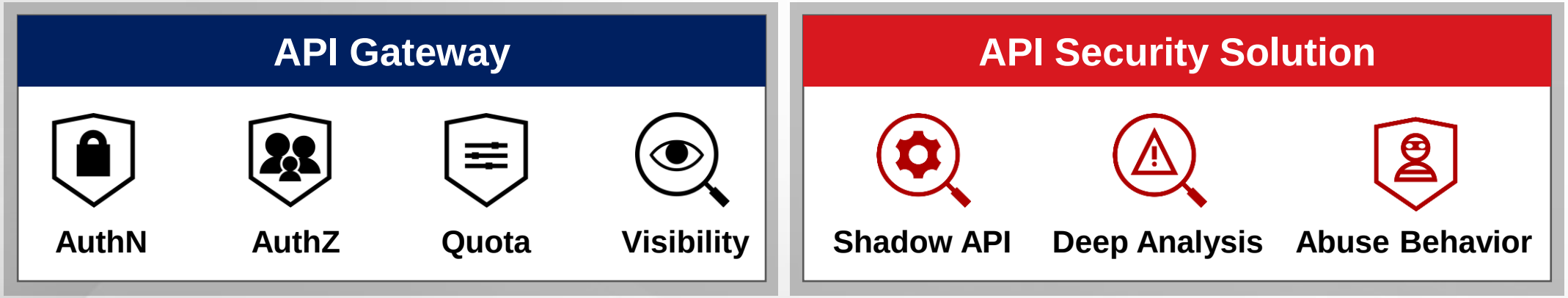
- API Visibility
- Timeline Investigation
- Signature / Behavior Detection
- Flexible Response

API 보안 솔루션 동향

TREND MICRO

API Gateway로 보안을 충족할 수 있나요?

"*API Gateway는 **통합 보안 기능과 인증**을 포함하고 있지만…*



탐지하기 까다로운 API 공격

- API 구현의 **기술적 취약점 악용**하는 행위
- 탈취된 계정을 통해 **합법적인 사용자로 위장**하는 행위
- 예기치 않은 방식으로 **비즈니스 오남용**하는 행위

API 자격 증명 스터핑

- ID / 비밀번호 유출사고 빈번히 발생

API를 통한 데이터 유출

- 성공적인 API 공격과 빈번한 Abuse
- 매우 민감한 비공개 정보 유출

API 보안 동향

“ API 보안전략을 개발할 때 고려해야할 주요 동향

행동 분석 및 이상징후 탐지	컨텍스트 확보로 모든 API활동을 AI 및 행위분석 기술로 이상행위 탐지
서비스형 소프트웨어로 전환	배포가 빠르고 쉬우며 AI 및 머신러닝의 힘을 활용
장기간 흐름 분석	단기간이 아닌 장기간에 걸친 History 행동 분석 및 이상징후 탐지
데브섹옵스 (비보안 이해관계자 수용)	API 보안 전략 및 도구, API의 구현 및 구성에 관련된 개발자와 시스템 간의 연계성을 높이는 것
API 보안과 활용	위협 헌팅, 사고 대응, API 개발 관행 개선

아카마이 API 보안 프레임워크

Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
Access to Logs	API Discovery	Risk Audit	Behavioral Detection	Response	Investigate & Threat Hunt
• API 환경에 대한 로그 • 로그에 접속 • 충분한 로그	• 모든 마이크로서비스 식별 • 모든 API 식별	• 위험 상태 - Misconfigured - Errors - Documented - Sensitive data	• 비즈니스 오남용 감지 • API 엔티티를 식별	• 자동화된 대응 • 대응 방법 지정	• 과거 데이터에서 위협 식별 • 위협을 헌팅

아카마이의 API Coverage

178 billion
매일 트리거되는 보안정책

+300 Trillion
2020년에 유입된 응답 수

+50%
전년대비 요청 수 증가

14%
알려진 악성행위 요청

+4,500
API를 제공하는 고객

83%
전체 요청의 API량

아카마이 API 보안

API Detection & Response

- 데이터에 대한 심층 분석에 초점
- 모든 API 사용자의 **행동 기준 설정**
- API **오남용 가능성 분석**
- 공격 및 이상징후 탐지

Advanced API Protection

- 행동 분석과 위협 헌팅 결합
- 쉐도우 API 및 **모든 API 식별**
- 비즈니스 컨텍스트의 오버레이
- **장기간 API 활동 분석** 및 헌팅

API Security Platform

- 모든 **API 인벤토리 지속 업데이트**
- **AI와 머신러닝** 기술 탑재
- SIEM, SOAR 연동 및 인시던트 대응
- API 활동 및 위협 온디맨드 접속 제공

Akamai

TREND MICRO

aalim@akamai.com

Thank you