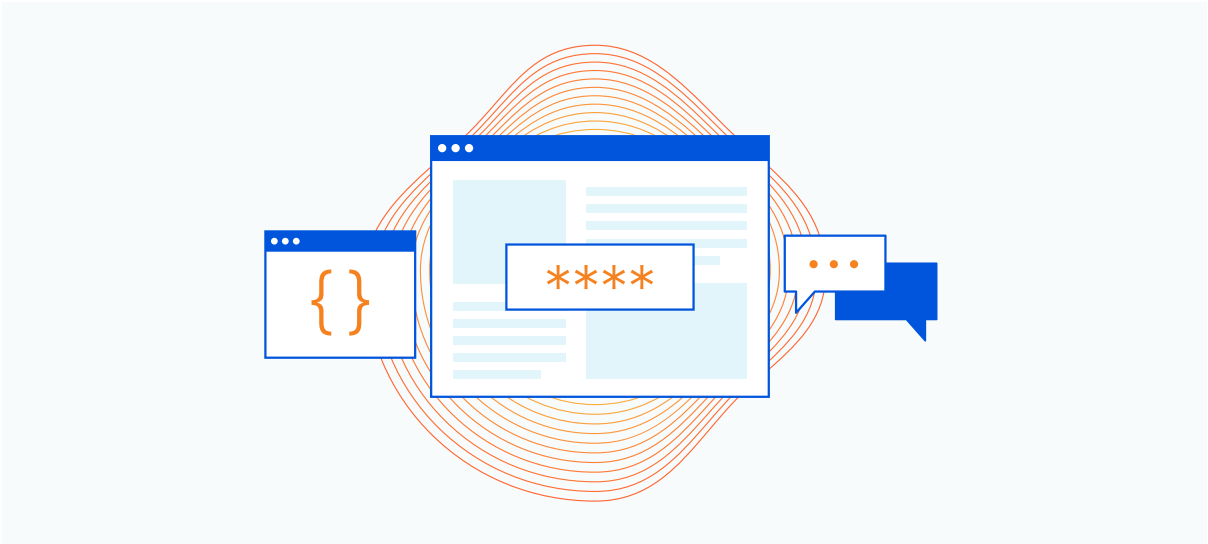
ホワイトペーパー

CLOUDFLARE

APIセキュリティのためのガイド

03-4510-1893 | enterprise@cloudflare.com | www.cloudflare.com

APIは（アプリの）世界を動かす



API（アプリケーションプログラミングインターフェース）が世界を動かしていることは、もう誰もが知っていることでしょう。より正確には、APIによって異なる最新のアプリケーションが互いに通信することができます。モバイルやWebアプリケーションは、データが保存され処理されるバックエンドにアクセスすることができます。APIには、異なる企業のアプリケーションが通信できるようにするパブリックなものと、ビジネス目標を達成するために内部のアプリケーションを統合する一般的なプライベートなものがあります。

その結果、より堅牢で本格的なアプリケーション、Webサイトおよびモバイルアプリを、幅広い機能と多様なデータで提供することができます。

例えば、ライドシェアリング企業は、自社で決済サービスを一から作るのではなく、決済会社のAPIを介して追加することができます。もう一つの例は、フライト集約サイトです。フライト時間、料金、目的地、その他航空券について知る必要のあるすべての情報を表示するために、航空会社のデータベースとAPIコールで接続し、適切なデータを引き出して集約検索結果のページに表示します。

APIの重要性はますます高まり、「API ファースト」を自称する企業も増えています。場合によっては、企業の実際の製品がAPIであり、それを消費することを中心としたビジネスモデルになっていることもあります。例えば、気象データを提供する企業がAPIを製品としている場合、気象情報を求める他の企業はAPIアクセスのための月額料金を支払うことになります。他の多くの例では、アプリケーションが他のアプリケーションと相互作用することが期待されるため、APIは実際の製品の機能を提供するコードと一緒に、あるいはそれ以前に設計され、開発の最後に追加されることはありません。

企業は、APIがどのように適切なデータを公開し、収益とビジネスモデルの基礎となるかを考慮し、慎重にAPIアプローチを作成するために時間と労力を費やしています。

しかし、完璧なAPIを作るのは大変です。他のソフトウェアと同じように、脆弱性が発生し、本書で説明するセキュリティの課題につながるからです。

APIはあらゆるところに存在し、今後数年の間にますますその勢いを増していくでしょうし、保護されなければなりません。このため、組織のAPIセキュリティについて考える際には、API攻撃と多層防御の側面について検討することにします。

数字で見るAPIの勢い

50%

のCloudflareを経由するトラフィックはAPIトラフィックです

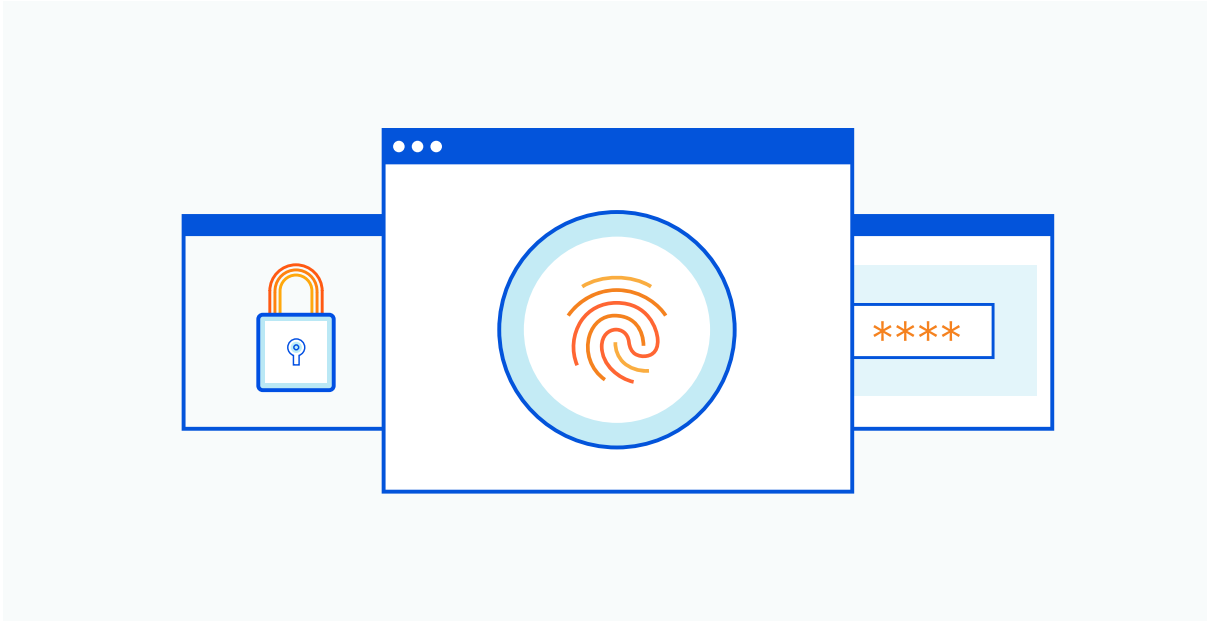
61%

APIトラフィックが前年同期比で増加

プログラム可能なWeb[1]は、24,000以上公開されている有名なAPIに気を配っています。しかし、ほとんどのAPIはプライベートなものであり、内部のアプリケーション同士をリンクしていることが分かります。プライベートAPIの数は数百万にのぼると推定されています。

そして、APIが勢いを増していると言えば、Cloudflareはその成長をじかに目の当たりにした証人です。2021年上半期のCloudflare Radarのデータによると、Cloudflareネットワーク上のトラフィックのおよそ半分がAPI関連でした。その上、2020年から2021年にかけて61%も増加しています。

重要なデータが公開されることを考えると、当社が保護しなければならない新たな攻撃対象領域がいかに巨大であるかということが見え始めてきます。なぜこのことが分かるのでしょうか。近年、APIを標的とした多くの顕著な攻撃が行われているのです。



[1]https://www.programmableweb.com/apis/directory

拡大する攻撃対象領域

当社は、APIがどこにでもあり、現代のビジネスの成功の基本であることを承知しています。APIは、アプリケーションのロジックを公開し、他のアプリケーションと機密データを共有することができます。

しかし、誰も驚きはしませんが、攻撃者はこのことを知っていて、企業内で拡大する攻撃対象領域を悪用しようと考えていることが判明しました。

2022年までに、APIの不正利用は「まれな攻撃ベクトルから最も頻繁な攻撃ベクトルに移行し、企業のWebアプリケーションのデータ漏えいを引き起こす」と断言したGartnerの言葉は、正しかったのかもしれません[2]。

攻撃ベクトルとして最も頻繁に狙われるのがAPIになるかどうかはまだ分かりませんが、APIが今後も攻撃者の標的になり続けることは明らかです。

当社が「なり続ける」と言ったのは、脆弱なAPIセキュリティ、あるいはセキュリティを考慮しないAPI開発によって、すでにいくつか侵害が起きているからです。

T-Mobileは2017年にAPI攻撃の被害に遭い、新しい端末を購入したり、T-Mobileのアカウントを申請したりした1500万人の顧客の情報が流出しました。データには、名前、住所、生年月日、社会保障番号、運転免許証番号、パスポート番号などが含まれていました。攻撃は、API呼び出しの電話番号パラメータを調整することで行われた、とViceは報じました。どのユーザーの電話番号でも問い合わせることができ、T-MobileのAPIは、電話番号が問い合わせられた人の機密アカウントデータを含むレスポンスを送信します。

別のAPI関連の侵害がUSPSを襲いました。リアルタイムのパッケージ追跡をサポートするAPIに基本的な認可が欠けていることが判明したのです。あるユーザーがログインすると、ワイルドカード検索パラメータを使用して、他のユーザーのアカウント情報を照会し、データセットの全レコードが引き出されたのです。このため、6000万人のUSPSアカウント所有者のデータが危険に晒されることになりました。

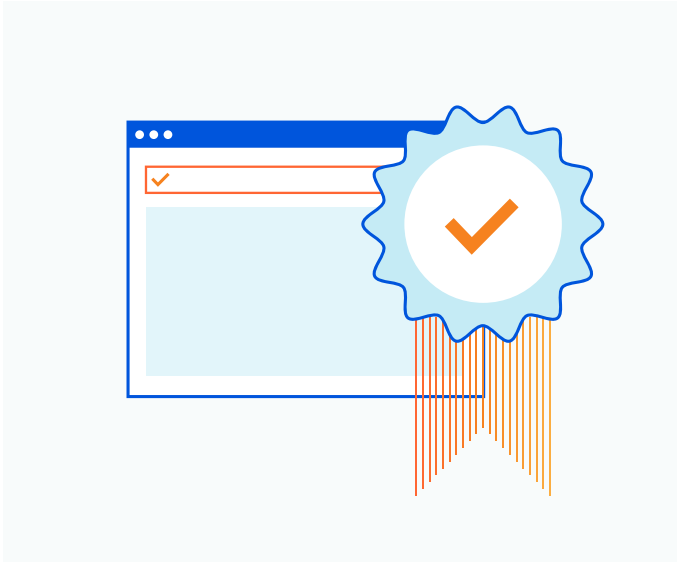
2019年、インドの大手ローカル検索エンジンJustDialは、事実上、新しいバージョンに置き換えられた古いバージョンのAPIを残していた際すべての顧客データを流出させてしまいました。さらに悪いことに、事実上、認証が行われていなかったため、誰でもAPIを呼び出し、本番サーバーからデータをスクレイピングすることができたのです。言い換えれば、ユーザーデータにアクセスするために高度な技術は必要なかったのです。

Facebookもまた、GraphQLでリーダーシップを発揮しているにもかかわらず、自分のAPIによって複数の侵害に遭っています。例えば2019年末、Facebookはデータベースをスクレイピングされ、2億6000万人以上のユーザー名、電話番号、ユーザーIDが危険に晒されました。

APIのセキュリティ確保に悩む組織は枚挙にいとまがありません。理由はいくつもあります。まず、安全でない設計による根本的なAPIの脆弱性が、攻撃への扉を開けています。さらに、これまでのところ、組織にはAPIファーストのセキュリティツールがありませんでした。おそらくWAFやレート制限のようなWebセキュリティツールを使うのでしょうが、それらはアプリケーションを保護するために作られたものであり、APIを保護するためのものではありません。これは誤検知のような問題につながる可能性があり、大部分が自動化されたトラフィック用に設計されたAPIセキュリティの必要性を明確にしています。

[2]出典：Gartner：「APIセキュリティ：APIを保護するために必要なこと」、2021年3月

リミックス！新しいOWASP APIトップ10



この中で明るい兆しは、アプリケーションセキュリティの向上に長年注力してきた組織であるOWASP財団が関与していることです。OWASPはそのトップ10Webアプリケーションセキュリティリスクで知られていますが、今度はAPIのセキュリティリスクと脆弱性のトップリストである「APIセキュリティトップ10」を発表しました。

実は、アプリケーションのセキュリティについて長い間懸念されてきたことは、APIの構築と保護についても当てはまります。

まず最初に、APIファーストを採用する組織は、APIを設計する際にセキュリティを前もって考慮しなければなりません。今述べたいくつかの攻撃と、彼らが悪用したOWASPのセキュリティリスクについて検証してみましょう。

OWASP APIトップ10

1. オブジェクトレベルの認可の不備
2. ユーザー認証の不備
3. 過剰なデータ露出
4. リソース不足およびレート制限
5. 機能レベルの認可の不備
6. マスアサインメント
7. 不適切なセキュリティ設定
8. インジェクション
9. 不適切な資産管理
10. 不十分なロギングとモニタリング

APIセキュリティの主な課題

1. 認証と認可の不備

上記の攻撃が悪用したいくつかの主要なOWASP APIリスクについて、認証と認可から詳しく見てみましょう。

JustDialは、エンドポイントでの認証の不備に遭い、誰でも呼び出すことができるようになってしまいました。認証を実装すると、正しいTLS証明書、APIキー、Webトークンなどを持つAPI呼び出しだけがリクエストを許可され、APIのセキュリティリスクを劇的に減らすことができます。

OWASPリストの第1位に話を移すと、多くのAPI攻撃はUSPSやT-Mobileで見られたような脆弱で、不備のある、あるいは存在しない認可を悪用します。オブジェクトレベルの認可の不備はよくあることで、APIエンドポイントが、アクセスを許可されたオブジェクトのID番号を、アクセスを許可されていない何かのIDに置き換えることによって悪用されます。多くの場合、リクエストのオブジェクトIDを変更するだけで、データへの不正なアクセスが可能になります。

APIパスおよびクエリーパラメータには、アクセスするリソースIDが含まれます。

認可された呼び出し：

`GET api.greatsampleapis.com/v1/users/235` 235はユーザーIDです。

操作されたAPI呼び出しは、235を236に変更し、つまりオブジェクト識別子、この場合はuserIDを調整し、ユーザー236のデータにアクセスすることで不正なアクセスを得ることができます。

`GET api.greatsampleapis.com/v1/users/236`

認可制御が行われていない場合、これは成功する可能性があります。開発者は、エンドポイントを脅威モデル化し、攻撃がオブジェクトのID値を調整または変更して他のデータにアクセスできないことを確認する必要があります。予測不可能なオブジェクトのID値を使用することでID値が連続せず、容易に推測できないようにすることも有効です。

2. マスアサインメント、データ流出およびインジェクション攻撃

レスポンスで過剰なデータを公開したり、入力で内部オブジェクトの変更を許可する攻撃もあります。

過剰なデータ露出は、APIがオブジェクトのプロパティを広く公開しようとして、レスポンスに過剰なデータを返し、リクエストを行うクライアントがデータをフィルタリングすることに依存する場合に起こります。

攻撃者は、レスポンスから得られる追加情報を使って、より強力な攻撃やフィッシングメールを作成することができます。例えば、レスポンスが以下のデータをすべて返した場合、非常に説得力のあるフィッシングメールに使用することができます。

```
{
“Id”: 213,
“FirstName”:“Sanjay”,
“LastName”:“Smythe”,
“EmailAddress”:“ssmythe@hacketyhack.com”,
“Occupation”: “Assistant to the Deputy Associate Vice Sub-undersecretary”,
“DOB”:“1986-05-21”,
“Bank”:“Easygo Financial”,
“AccountNumber”: 1362886306,
“PetName”: “Aloysius”,
}
```

マスアサインメント攻撃は、APIが内部のオブジェクトや変数を公開する際に、API呼び出しによって内部の値を変更したり悪用したりするものです。

OWASPは、このように言っています。

> 「ソフトウェアフレームワークは、開発者がフレームワークを使いやすくするために、HTTPリクエストのパラメータをプログラムコードの変数やオブジェクトに自動的に紐づけできる場合があります…。攻撃者はこの方法を利用して、開発者が意図しない新しいパラメータを作成し、その結果、意図しない新しい変数やオブジェクトをプログラムコード内に作成したり上書きしたりすることがあります。」

開発者はどうすればいいのでしょうか。開発者は、開発中における大量割り当てを呼び出す際の潜在的なリスクを理解し、悪用される可能性のある内部オブジェクトや変数を公開しないようにする必要があります。また、クライアントが更新できるようなプロパティをリストアップすることも検討すべきです。

Webアプリケーションは長い間、インジェクション攻撃の影響を受けやすく、それはAPIも同様です。インジェクション攻撃はよく知られているので、ここでは触れませんが、入力は渡される前に検証され、サニタイズされなければならないと言えば十分でしょう。また、APIのレスポンスにはデータ漏えい防止策を施し、大量の情報開示インシデントを防ぐために返されるレコードの数を制限するよう努力しなければなりません。

3. リソースの不正利用とシャドー/不正API

その他の攻撃がAPIを不正利用することで、過度の計算リソースを消費し、対処しきれなくなってDoS攻撃のような攻撃の被害を被る可能性もあります。クライアント/リソースごとのリクエスト数、1つのレスポンスで返されるレコード数、リクエストペイロードのサイズなどに制限が設けられていない場合、こうした攻撃への入り口は開かれたままになってしまいます。

JustDial攻撃で見られたように、本番APIは保護されていない可能性が高く、悪用される可能性があるため、忘れられ、シャドーまたは不正となる可能性があります。セキュリティの他の部分と同様に、適切なセキュリティ管理を適用するためには、ITの状態や攻撃対象領域を可視化する必要があります。APIエンドポイントの状態全体を可視化することも同様です。

APIを開発する場合、チームはAPIのバージョンを追跡するための洗練されたプロセスを持ち、どのAPIが実稼働しているか、何が非推奨であるかを理解する必要があります。

APIを保護するための考慮事項

これまで、APIとは何か、なぜ重要なのか、そしてAPIを標的とした一般的な攻撃について説明してきました。次に、最も一般的な攻撃からAPIを保護するために、CloudflareがどのようにAPIセキュリティを構築したかを確認してみましょう。効果的なAPIセキュリティは、可視性から、ポジティブなセキュリティモデル、不正利用の阻止、データ保護まで、すべてを考慮しなければなりません。

Cloudflare API Shield

可視性の基礎

可視性

セキュリティの他の側面と同様に、当社は保護するために、何かを見なければなりません。特に企業が数百、数千のAPIエンドポイントを持っている場合、APIも同様です。

APIの検出と可視性は、APIを管理するための重要な基礎となる側面であるため、組織は常にAPIエンドポイントの状態の明確なスナップショットを持ち、シャドーAPIや不正なAPIが問題になることを防ぐことができます。

JustDialで見たように、組織がAPIを見失うと、データ漏えいにつながる可能性があります。

APIの多層防御

API L7の保護

当社は長い間、レイヤー7のDDoS攻撃からアプリケーションを保護するためにWebアプリケーションファイアウォールを展開してきました。APIの保護は、サービス拒否攻撃やブルートフォースログイン攻撃、特定のIPアドレスによる一般的な不正利用を防ぐために、レート制限やDDoS保護のような多くの信頼できるコントロールから始める必要があります。これはAPIの使用制限を強制し、OWASP API 4—リソース不足とレート制限に対抗する可用性を確保するものです。

WAFのルールは、APIを標的とした一般的な攻撃を特定し、ブロックするためにも使用されるべきです。

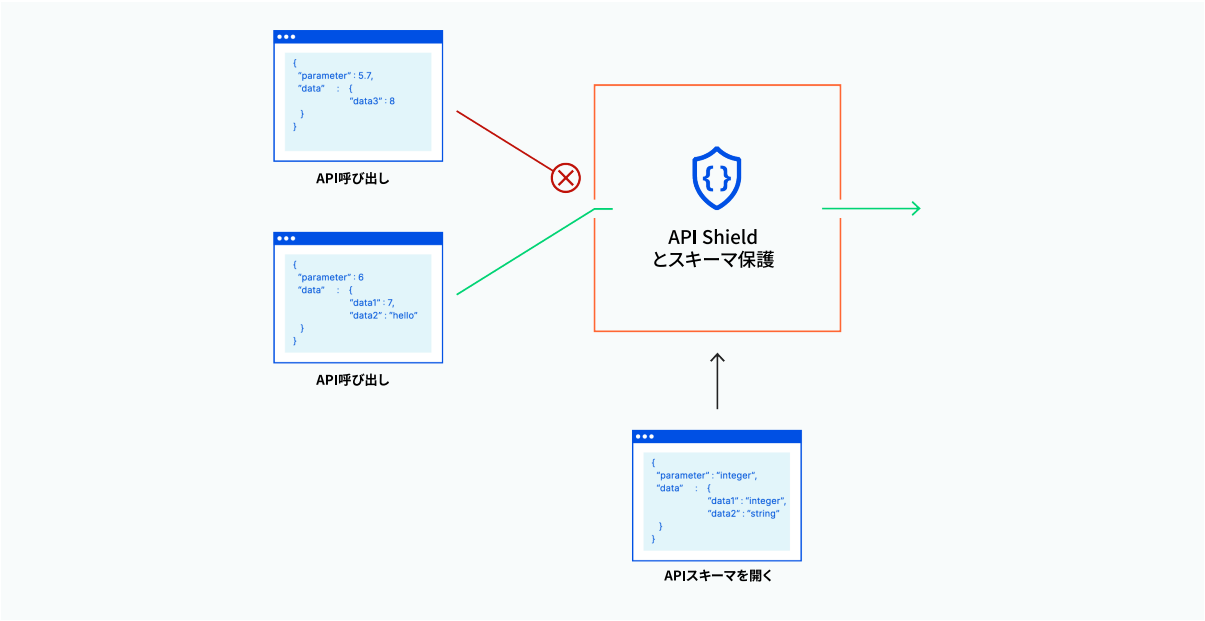
認証と承認

mTLS認証

当社が概説したAPI攻撃では、認証の欠如が壊滅的な被害をもたらすことがわかりました。認証は最初から組み込まれていなければならず、モバイルやIoTのようなユースケースでは証明書ベースのIDを強制するために相互TLSで強化されなければなりません。このアプローチは、有効な証明書を持つ正当なクライアントからのリクエストのみ接続を許可する、より積極的な許可リストモデルです。

資格情報の流出チェック

APIは、盗まれた資格情報を使ってログインを繰り返し試みるクレデンシャルスタッフィング攻撃から免れることはできません。これらのアカウント認証情報は、組織の管理外のサードパーティの侵害によって安全性が損なわれる可能性があります。認証チェックの一環として、APIセキュリティはログイン時に認証情報をスキャンし、流出した資格情報のデータベースと照合することができるはずです。資格情報の安全性が損なわれたように見える場合、APIセキュリティはパスワードリセットや多要素認証のような追加のセキュリティ手段を発動させ、もちろん試行をブロックする必要があります。



スキーマ検証では、APIスキーマのログまたはリクエストに準拠しないブロックリクエストと照合して、各リクエストを評価します。

ポジティブなAPI保護

APIスキーマの検証

開発者はAPIスキーマを作成するために多大な労力を費やします。これは、他の人がAPIとどのように相互作用することを期待するかの文書、または基本ルールです。これは、各エンドポイントにおけるリクエストメソッドやオペレーション(GET /users, POST /users)、または各オペレーションの入出力パラメータといったものを確立することができます。OpenAPI v3は、一般にSwagger標準としても知られており、APIを定義するための最も有名なスキーマです。

信頼できるAPIセキュリティは、スキーマに強制するポジティブなZero Trustモデルを使用すべきです。

スキーマがあれば、リクエストは自動的にスキーマに照らして検証されるはずです。スキーマに準拠していることが検証されたものを除き、すべてのAPI操作はブロックされます。